

RL78/G22

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Preliminary

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 be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

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

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

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

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

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/

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.

How to Use This Manual

Readers This manual is intended for user engineers who wish to understand the functions of the RL78/G22 and design and develop application systems and programs for these devices. The target products are as follows.

- 16-pin: R7F102G4x (x = C, E)**Note**
 - 20-pin: R7F102G6x (x = C, E)
 - 24-pin: R7F102G7x (x = C, E)
 - 25-pin: R7F102G8x (x = C, E)**Note**
 - 30-pin: R7F102GAx (x = C, E)
- 32-pin: R7F102GBx (x = C, E)
 - 36-pin: R7F102GCx (x = C, E)**Note**
 - 40-pin: R7F102GEx (x = C, E)
 - 44-pin: R7F102GFx (x = C, E)
 - 48-pin: R7F102GGx (x = C, E)

Note The 16-pin plastic HWQFN, 25-pin plastic WFLGA, and 36-pin plastic WFLGA products are in planning. Contact a Renesas Electronics sales office for details.

Purpose This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization The RL78/G22 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G22 User's Manual Hardware (This Manual) • Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	RL78 Family User's Manual Software • CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
 - Read this manual in the order of the **CONTENTS**.
- How to interpret the register format:
 - For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the RL78/G22 Microcontroller instructions:
 - Refer to the separate document **RL78 Family User's Manual: Software (R01US0015E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representations:	$\overline{\text{xxx}}$ (overscore over pin and signal name)
Note:	Footnote for item marked with Note in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representations:	Binary.....xxxx or xxxxB
	Decimal.....xxxx
	Hexadecimal xxxxH

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Devices

Document Name	Document No.
RL78/G22 User's Manual Hardware	This manual
RL78 Family User's Manual Software	R01US0015E

Documents Related to Flash Memory Programming and On-chip Debugging

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	Note 1
E2 Emulator User's Manual	R20UT3538E
E2 Emulator Lite RTE0T0002LKCE00000R User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	Note 2

Note 1. For a list of the documents relevant to the PG-FP6, visit the Web page below.
<https://www.renesas.com/us/en/software-tool/pg-fp6>

Note 2. For a list of the documents relevant to the Renesas Flash Programmer, visit the Web page below.
<https://www.renesas.com/us/en/software-tool/renesas-flash-programmer-programming-gui>

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
Renesas Microcontrollers RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	R50ZZ0003E
Semiconductor Reliability Handbook	R51ZZ0001E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

All trademarks and registered trademarks are the property of their respective owners.

EEPROM is a trademark of Renesas Electronics Corporation.

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

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

CONTENTS

1.	Outline	1
1.1	Features	1
1.2	List of Part Numbers	5
1.3	Pin Configuration (Top View)	7
1.3.1	16-pin products	7
1.3.2	20-pin products	8
1.3.3	24-pin products	9
1.3.4	25-pin products	10
1.3.5	30-pin products	11
1.3.6	32-pin products	12
1.3.7	36-pin products	13
1.3.8	40-pin products	14
1.3.9	44-pin products	15
1.3.10	48-pin products	16
1.4	Pin Identification	17
1.5	Block Diagram	18
1.6	Outline of Functions	19
2.	Pin Functions	23
2.1	Functions of Port Pins	23
2.1.1	16-pin products	24
2.1.2	20-pin products	25
2.1.3	24-pin products	26
2.1.4	25-pin products	28
2.1.5	30-pin products	30
2.1.6	32-pin products	32
2.1.7	36-pin products	34
2.1.8	40-pin products	36
2.1.9	44-pin products	38
2.1.10	48-pin products	40
2.2	Pin Functions Other than Port Pin Functions	42
2.2.1	Functions for each product	42
2.2.2	Description of pin functions	46
2.3	Connection of Unused Pins	48
2.4	Block Diagrams of Pins	49
3.	CPU Architecture	77
3.1	Memory Space	78
3.1.1	Internal program memory space	81
3.1.2	Mirror area	85
3.1.3	Internal data memory space	87
3.1.4	Special function register (SFR) area	88
3.1.5	Extended special function register (2nd SFR: 2nd special function register) area	88
3.1.6	Data memory addressing	89

3.2	Processor Registers	90
3.2.1	Control registers	90
3.2.2	General-purpose registers	92
3.2.3	ES and CS registers	93
3.2.4	Special function registers (SFRs)	94
3.2.5	Extended special function registers (2nd SFRs)	98
3.3	Instruction Address Addressing	110
3.3.1	Relative addressing	110
3.3.2	Immediate addressing	110
3.3.3	Table indirect addressing	111
3.3.4	Register indirect addressing	112
3.4	Addressing for Processing Data Addresses	113
3.4.1	Implied addressing	113
3.4.2	Register addressing	114
3.4.3	Direct addressing	115
3.4.4	Short direct addressing	116
3.4.5	SFR addressing	117
3.4.6	Register indirect addressing	118
3.4.7	Based addressing	119
3.4.8	Based indexed addressing	122
3.4.9	Stack addressing	123
4.	Port Functions	126
4.1	Port Functions	126
4.2	Port Configuration	126
4.2.1	Port 0	128
4.2.2	Port 1	128
4.2.3	Port 2	128
4.2.4	Port 3	128
4.2.5	Port 4	129
4.2.6	Port 5	129
4.2.7	Port 6	129
4.2.8	Port 7	129
4.2.9	Port 12	130
4.2.10	Port 13	130
4.2.11	Port 14	130
4.3	Registers for Controlling the Port Function	131
4.3.1	Port mode registers (PMxx)	135
4.3.2	Port registers (Pxx)	136
4.3.3	Pull-up resistor option registers (PUxx)	137
4.3.4	Port input mode registers (PIMxx)	138
4.3.5	Port output mode registers (POMxx)	139
4.3.6	Port digital input disable registers (PDIDISxx)	140
4.3.7	Port mode control A registers (PMCAxx)	141
4.3.8	Port mode control T registers (PMCTxx)	142
4.3.9	Peripheral I/O redirection register (PIOR)	143
4.3.10	Port function output enable register 1 (PFOE1)	144
4.3.11	Port mode select register (PMS)	145

4.4	Port Function Operations	146
4.4.1	Writing to I/O port	146
4.4.2	Reading from I/O port	146
4.4.3	Operations on I/O port	146
4.4.4	Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers	147
4.5	Register Settings When Using Alternate Function	149
4.5.1	Basic concept when using alternate function	149
4.5.2	Register settings for alternate function whose output function is not used	151
4.5.3	Register settings and port pin state	152
4.5.4	Examples of register settings for port and alternate functions	152
4.6	Cautions When Using Port Function	164
4.6.1	Cautions on 1-bit manipulation instruction for port register n (Pn)	164
4.6.2	Notes on specifying the pin settings	165
5.	Operation State Control	166
5.1	Configuration of Operation State Control	166
5.2	Registers for Controlling the Operation State Control	168
5.2.1	Flash operating mode select register (FLMODE)	168
5.2.2	Flash operating mode protect register (FLMWRP)	170
5.3	Initial Setting of Flash Operation Modes	171
5.4	Transitions between Flash Operation Modes	172
5.5	Details of Flash Operation Modes	173
5.5.1	Details of HS (high-speed main) mode	173
5.5.2	Details of LS (low-speed main) mode	174
5.5.3	Details of LP (low-power main) mode	175
5.5.4	Details on SUB mode	176
6.	Clock Generator	177
6.1	Functions of Clock Generator	177
6.2	Configuration of Clock Generator	179
6.3	Registers for Controlling the Clock Generator	182
6.3.1	Clock operation mode control register (CMC)	183
6.3.2	System clock control register (CKC)	186
6.3.3	Clock operation status control register (CSC)	188
6.3.4	Oscillation stabilization time counter status register (OSTC)	189
6.3.5	Oscillation stabilization time select register (OSTS)	191
6.3.6	Peripheral enable registers 0, 1 (PER0, PER1)	193
6.3.7	Subsystem clock supply mode control register (OSMC)	198
6.3.8	Subsystem clock select register (CKSEL)	200
6.3.9	High-speed on-chip oscillator frequency select register (HOCODIV)	201
6.3.10	Middle-speed on-chip oscillator frequency select register (MOCODIV)	202
6.3.11	High-speed system clock division register (MOSCDIV)	203
6.3.12	High-speed on-chip oscillator trimming register (HIOTRM)	204
6.3.13	Middle-speed on-chip oscillator trimming register (MIOTRM)	205
6.3.14	Low-speed on-chip oscillator trimming register (LIOTRM)	206
6.3.15	Standby mode release setting register (WKUPMD)	207
6.4	System Clock Oscillator	208
6.4.1	X1 oscillator	208

6.4.2	XT1 oscillator	208
6.4.3	High-speed on-chip oscillator	212
6.4.4	Middle-speed on-chip oscillator	212
6.4.5	Low-speed on-chip oscillator	212
6.5	Operations of the Clock Generator	213
6.6	Controlling Clocks	215
6.6.1	Example of setting the high-speed on-chip oscillator	215
6.6.2	Example of setting the X1 oscillator clock	217
6.6.3	Example of setting the XT1 oscillator clock	218
6.6.4	State transitions of the CPU clock	219
6.6.5	Conditions before changing the CPU clock and processing after changing the CPU clock	224
6.6.6	Time required for switchover of the CPU clock and main system clock	228
6.6.7	Conditions before clock oscillation is stopped	230
6.7	Resonator and Oscillator Constants	231
7.	Timer Array Unit (TAU)	232
7.1	Functions of Timer Array Unit	234
7.1.1	Independent channel operation function	234
7.1.2	Simultaneous channel operation function	236
7.1.3	8-bit timer operation function (channels 1 and 3 only)	237
7.1.4	LIN-bus supporting function (channel 7 of unit 0 only)	237
7.2	Configuration of Timer Array Unit	238
7.2.1	Timer counter register mn (TCRmn)	244
7.2.2	Timer data register mn (TDRmn)	246
7.3	Registers for Controlling the Timer Array Unit	247
7.3.1	Peripheral enable register 0 (PER0)	248
7.3.2	Peripheral reset control register 0 (PRR0)	249
7.3.3	Timer clock select register m (TPSm)	250
7.3.4	Timer mode register mn (TMRmn)	254
7.3.5	Timer status register mn (TSRmn)	258
7.3.6	Timer channel enable status register m (TEm)	259
7.3.7	Timer channel start register m (TSm)	260
7.3.8	Timer channel stop register m (TTm)	261
7.3.9	Timer input select register 0 (TIS0)	262
7.3.10	Timer input select register 1 (TIS1)	263
7.3.11	Timer output enable register m (TOEm)	264
7.3.12	Timer output register m (TOM)	265
7.3.13	Timer output level register m (TOLm)	266
7.3.14	Timer output mode register m (TOMm)	267
7.3.15	Input switch control register (ISC)	268
7.3.16	Noise filter enable register 1 (NFEN1)	269
7.3.17	Registers controlling port functions of pins to be used for timer I/O	271
7.4	Basic Rules of Timer Array Unit	272
7.4.1	Basic rules of simultaneous channel operation function	272
7.4.2	Basic rules of 8-bit timer operation function (channels 1 and 3 only)	274
7.5	Operations of Counters	275
7.5.1	Count clock (fCLK)	275
7.5.2	Timing of the start of counting	277

7.5.3	Operations of Counters	278
7.6	Channel Output (TOMn Pin) Control	283
7.6.1	TOMn pin output circuit configuration	283
7.6.2	TOMn pin output setting	284
7.6.3	Cautions on channel output operation	285
7.6.4	Collective manipulation of TOMn bit	290
7.6.5	Timer interrupts and TOMn outputs when counting is started	291
7.7	Timer Input (TImn) Control	292
7.7.1	TImn input circuit configuration	292
7.7.2	Noise filter	292
7.7.3	Cautions on channel input operation	293
7.8	Independent Channel Operation Function of Timer Array Unit	294
7.8.1	Operation as an interval timer or for square wave output	294
7.8.2	Operation as an external event counter	298
7.8.3	Operation as a frequency divider (channel 0 of unit 0 only)	302
7.8.4	Operation for input pulse interval measurement	307
7.8.5	Operation for input signal high-/low-level width measurement	312
7.8.6	Operation as a delay counter	316
7.9	Simultaneous Channel Operation Function of Timer Array Unit	320
7.9.1	Operation for the one-shot pulse output function	320
7.9.2	Operation for the PWM function	327
7.9.3	Operation for the multiple PWM output function	334
7.10	Cautions When Using Timer Array Unit	342
7.10.1	Cautions when using timer output	342
7.10.2	Point for caution when a timer output is to be used as an event input for the ELC	342
8.	Realtime Clock (RTC)	343
8.1	Functions of Realtime Clock	343
8.2	Configuration of the Realtime Clock	344
8.3	Registers for Controlling the Realtime Clock	346
8.3.1	Peripheral enable register 0 (PER0)	347
8.3.2	Subsystem clock supply mode control register (OSMC)	348
8.3.3	Realtime clock control register 0 (RTCC0)	349
8.3.4	Realtime clock control register 1 (RTCC1)	351
8.3.5	Second count register (SEC)	353
8.3.6	Minute count register (MIN)	353
8.3.7	Hour count register (HOUR)	354
8.3.8	Day count register (DAY)	356
8.3.9	Day-of-week count register (WEEK)	357
8.3.10	Month count register (MONTH)	358
8.3.11	Year count register (YEAR)	358
8.3.12	Time error correction register (SUBCUD)	359
8.3.13	Alarm minute register (ALARMWM)	360
8.3.14	Alarm hour register (ALARMWH)	360
8.3.15	Alarm day-of-week register (ALARMWW)	361
8.3.16	Port mode register 3 (PM3)	361
8.3.17	Port register 3 (P3)	362
8.4	Operations of the Realtime Clock	363
8.4.1	Starting the realtime clock operation	363

8.4.2	Shifting to HALT or STOP mode after starting operation	365
8.4.3	Reading from and writing to the counters of the realtime clock	366
8.4.4	Setting alarm by the realtime clock	368
8.4.5	1 Hz output by the realtime clock	369
8.4.6	Example of time error correction by the realtime clock	370
9.	32-bit Interval Timer (TML32)	375
9.1	Overview	375
9.2	Registers for Controlling the 32-bit Interval Timer	378
9.2.1	Peripheral enable register 1 (PER1)	379
9.2.2	Peripheral reset control register 1 (PRR1)	380
9.2.3	Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)	381
9.2.4	Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1)	381
9.2.5	Interval timer capture register 00 (ITLCAP00)	382
9.2.6	Interval timer control register (ITLCTL0)	383
9.2.7	Interval timer clock select register 0 (ITLCSEL0)	385
9.2.8	Interval timer frequency division register 0 (ITLFDIV00)	386
9.2.9	Interval timer frequency division register 1 (ITLFDIV01)	388
9.2.10	Interval timer capture control register 0 (ITLCC0)	389
9.2.11	Interval timer status register (ITLS0)	391
9.2.12	Interval timer match detection mask register (ITLMKF0)	393
9.3	Operation	394
9.3.1	Counter mode settings	394
9.3.2	Capture mode settings	397
9.3.3	Timer operation	398
9.3.4	Capture operation	399
9.3.5	Interrupt	401
9.3.6	Interval timer setting procedures	403
10.	Clock Output/Buzzer Output Controller (PCLBUZ)	407
10.1	Functions of Clock Output/Buzzer Output Controller	407
10.2	Configuration of Clock Output/Buzzer Output Controller	408
10.3	Registers for Controlling the Clock Output/Buzzer Output Controller	408
10.3.1	Clock output select registers n (CKSn)	408
10.3.2	Registers controlling port functions of pins to be used for clock or buzzer output	410
10.4	Operations of the Clock Output/Buzzer Output Controller	410
10.4.1	Operation of output pins	410
10.5	Point for Caution when the Clock Output/Buzzer Output Controller is to be Used	411
11.	Watchdog Timer (WDT)	412
11.1	Functions of Watchdog Timer	412
11.2	Configuration of Watchdog Timer	413
11.3	Register for Controlling the Watchdog Timer	414
11.3.1	Watchdog timer enable register (WDTE)	414
11.4	Operation of Watchdog Timer	415
11.4.1	Controlling operation of watchdog timer	415
11.4.2	Setting overflow time of watchdog timer	416
11.4.3	Setting window open period of watchdog timer	417
11.4.4	Setting watchdog timer interval interrupt	419

12.	A/D Converter (ADC)	420
12.1	Function of A/D Converter	420
12.2	Configuration of A/D Converter	423
12.3	Registers for Controlling the A/D Converter	425
12.3.1	Peripheral enable register 0 (PER0)	426
12.3.2	Peripheral reset control register 0 (PRR0)	427
12.3.3	A/D converter mode register 0 (ADM0)	428
12.3.4	A/D converter mode register 1 (ADM1)	436
12.3.5	A/D converter mode register 2 (ADM2)	437
12.3.6	10-bit A/D conversion result register (ADCR)	440
12.3.7	8-bit A/D conversion result register (ADCRH)	441
12.3.8	Analog input channel specification register (ADS)	442
12.3.9	Conversion result comparison upper limit setting register (ADUL)	444
12.3.10	Conversion result comparison lower limit setting register (ADLL)	444
12.3.11	A/D test register (ADTES)	445
12.3.12	Registers controlling port function of analog input pins	445
12.4	A/D Converter Conversion Operations	446
12.5	Input Voltage and Conversion Results	448
12.6	A/D Converter Operation Modes	449
12.6.1	Software trigger mode (select mode, sequential conversion mode)	449
12.6.2	Software trigger mode (select mode, one-shot conversion mode)	450
12.6.3	Software trigger mode (scan mode, sequential conversion mode)	451
12.6.4	Software trigger mode (scan mode, one-shot conversion mode)	452
12.6.5	Hardware trigger no-wait mode (select mode, sequential conversion mode)	453
12.6.6	Hardware trigger no-wait mode (select mode, one-shot conversion mode)	454
12.6.7	Hardware trigger no-wait mode (scan mode, sequential conversion mode)	455
12.6.8	Hardware trigger no-wait mode (scan mode, one-shot conversion mode)	456
12.6.9	Hardware trigger wait mode (select mode, sequential conversion mode)	457
12.6.10	Hardware trigger wait mode (select mode, one-shot conversion mode)	458
12.6.11	Hardware trigger wait mode (scan mode, sequential conversion mode)	459
12.6.12	Hardware trigger wait mode (scan mode, one-shot conversion mode)	460
12.7	A/D Converter Setup Flowchart	461
12.7.1	Setting up software trigger mode	461
12.7.2	Setting up hardware trigger no-wait mode	462
12.7.3	Setting up hardware trigger wait mode	463
12.7.4	Setup when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected (example for software trigger mode and one-shot conversion mode)	464
12.7.5	Setting up test mode	465
12.8	SNOOZE Mode Function	466
12.9	How to Read A/D Converter Characteristics Table	470
12.10	Points for Caution when the A/D Converter is to be Used	474
13.	Serial Array Unit (SAU)	478
13.1	Functions of Serial Array Unit	480
13.1.1	Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)	480
13.1.2	UART (UART0 to UART2)	481
13.1.3	Simplified I ² C (IIC00, IIC01, IIC11, IIC20, IIC21)	482

13.2	Configuration of Serial Array Unit	483
13.2.1	Shift Register	486
13.2.2	Lower 8 or 9 bits of the serial data register mn (SDRmn)	486
13.3	Registers for Controlling the Serial Array Unit	488
13.3.1	Peripheral enable register 0 (PER0)	489
13.3.2	Peripheral reset control register 0 (PRR0)	491
13.3.3	Serial clock select register m (SPSm)	492
13.3.4	Serial mode register mn (SMRmn)	493
13.3.5	Serial communication operation setting register mn (SCRmn)	495
13.3.6	Serial data register mn (SDRmn)	497
13.3.7	Serial flag clear trigger register mn (SIRmn)	499
13.3.8	Serial status register mn (SSRmn)	500
13.3.9	Serial channel start register m (SSm)	502
13.3.10	Serial channel stop register m (STm)	503
13.3.11	Serial channel enable status register m (SEm)	504
13.3.12	Serial output enable register m (SOEm)	505
13.3.13	Serial output register m (SOM)	506
13.3.14	Serial output level register m (SOLm)	507
13.3.15	Serial standby control register 0 (SSC0)	509
13.3.16	Input switch control register (ISC)	511
13.3.17	Noise filter enable register 0 (NFEN0)	512
13.3.18	Registers controlling port functions of serial input/output pins	513
13.3.19	UART loopback select register (ULBS)	514
13.4	Operation Stop Mode	515
13.4.1	Stopping the Operation by Units	515
13.4.2	Stopping the Operation by Channels	516
13.5	Operation of Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) Communication	517
13.5.1	Master Transmission	520
13.5.2	Master Reception	529
13.5.3	Master Transmission/Reception	539
13.5.4	Slave Transmission	548
13.5.5	Slave Reception	558
13.5.6	Slave Transmission/Reception	565
13.5.7	SNOOZE Mode	575
13.5.8	Calculating Transfer Clock Frequency	580
13.5.9	Procedure for Processing Errors that Occurred during Simplified SPI (CSI00, CSI01, CSI11, CSI20, and CSI21) Communication	582
13.6	Operation of UART (UART0 to UART2) Communication	583
13.6.1	UART Transmission	586
13.6.2	UART Reception	595
13.6.3	SNOOZE Mode	602
13.6.4	Calculating Baud Rate	609
13.6.5	Procedure for Processing Errors that Occurred during UART (UART0 to UART2) Communication	614
13.7	LIN Communication Operation	615
13.7.1	LIN Transmission	615
13.7.2	LIN Reception	618
13.8	Operation of Simplified I ² C (IIC00, IIC01, IIC11, IIC20, and IIC21) Communication	624
13.8.1	Address Field Transmission	627

13.8.2	Data Transmission	633
13.8.3	Data Reception	636
13.8.4	Stop Condition Generation	640
13.8.5	Calculating Transfer Rate	642
13.8.6	Procedure for Processing Errors that Occurred during Simplified I ² C (IIC00, IIC01, IIC11, IIC20, and IIC21) Communication	645
14.	Serial Interface IICA (IICA)	646
14.1	Functions of Serial Interface IICA	646
14.2	Configuration of Serial Interface IICA	649
14.3	Registers for Controlling Serial Interface IICA	652
14.3.1	Peripheral enable register 0 (PER0)	653
14.3.2	Peripheral reset control register 0 (PRR0)	654
14.3.3	IICA control register n0 (IICCTLn0)	655
14.3.4	IICA status register n (IICSn)	660
14.3.5	IICA flag register n (IICFn)	664
14.3.6	IICA control register n1 (IICCTLn1)	666
14.3.7	IICA low-level width setting register n (IICWLn)	669
14.3.8	IICA high-level width setting register n (IICWHn)	669
14.3.9	Registers for controlling the port function multiplexed with the IICA I/O pins	670
14.4	I ² C Bus Mode Functions	671
14.4.1	Pin configuration	671
14.4.2	Setting transfer clock by using IICWLn and IICWHn registers	672
14.5	I ² C Bus Definitions and Control Methods	674
14.5.1	Start conditions	674
14.5.2	Address	675
14.5.3	Transfer direction specification	675
14.5.4	Acknowledge (ACK)	676
14.5.5	Stop condition	677
14.5.6	Clock stretching	678
14.5.7	Release from clock stretching	680
14.5.8	Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching	681
14.5.9	Address match detection method	682
14.5.10	Error detection	682
14.5.11	Extension code	683
14.5.12	Arbitration	684
14.5.13	Wakeup function	686
14.5.14	Communication reservation	689
14.5.15	Cautions	693
14.5.16	Communication operations	694
14.5.17	Timing of I ² C interrupt request signal (INTIICAn) occurrence	702
14.6	Timing Charts	723
15.	Serial Interface UARTA (UARTA)	738
15.1	Overview	738
15.2	Registers for Controlling the Serial Interface UARTA	740
15.2.1	Peripheral enable register 1 (PER1)	741
15.2.2	Transmit buffer register (TXBAn)	742

15.2.3	Receive buffer register (RXBAn)	743
15.2.4	Operation mode setting register 0 (ASIMAn0)	744
15.2.5	Operation mode setting register 1 (ASIMAn1)	746
15.2.6	Baud rate generator control register (BRGCAn)	747
15.2.7	Status register (ASISAn)	748
15.2.8	Status clear trigger register (ASCTAn)	750
15.2.9	UARTA clock select register 0 (UTA0CK)	751
15.2.10	UART loopback select register (ULBS)	752
15.3	Operation	753
15.3.1	Operation stop mode	753
15.3.2	UART mode	753
15.3.3	Receive data noise filter	767
15.3.4	Baud rate generator	768
15.4	Points for Caution when the Serial Interface UARTA is to be Used	775
15.4.1	Port setting for RxDAn pin	775
15.4.2	Point for caution when selecting the UARTAn operation clock (fUTAn)	775
16.	Data Transfer Controller (DTC)	776
16.1	Functions of DTC	776
16.2	Configuration of DTC	778
16.3	Registers for Controlling the DTC	779
16.3.1	Allocation of DTC control data area and DTC vector table area	780
16.3.2	Control data allocation	781
16.3.3	Vector table	783
16.3.4	Peripheral enable register 1 (PER1)	785
16.3.5	DTC control register j (DTCCRj) (j = 0 to 23)	786
16.3.6	DTC block size register j (DTBLSj) (j = 0 to 23)	787
16.3.7	DTC transfer count register j (DTCCTj) (j = 0 to 23)	788
16.3.8	DTC transfer count reload register j (DTRLdj) (j = 0 to 23)	788
16.3.9	DTC source address register j (DTSARj) (j = 0 to 23)	789
16.3.10	DTC destination address register j (DTDARj) (j = 0 to 23)	789
16.3.11	DTC activation enable register i (DTCENi) (i = 0 to 4)	790
16.3.12	DTC base address register (DTCBAR)	793
16.4	DTC Operation	794
16.4.1	Activation sources	795
16.4.2	Normal mode	796
16.4.3	Repeat mode	799
16.4.4	Chain transfers	802
16.5	Points for Caution when the DTC is to be Used	804
16.5.1	Setting DTC control data and vector table	804
16.5.2	Allocation of DTC control data area and DTC vector table area	804
16.5.3	DTC pending instruction	805
16.5.4	Operation when accessing data flash memory space	805
16.5.5	Number of DTC execution clock cycles	806
16.5.6	DTC response time	807
16.5.7	DTC activation sources	807
16.5.8	Operation in standby mode	808

17.	Event Link Controller (ELC)	809
17.1	Functions of ELC	809
17.2	Configuration of ELC	809
17.3	Registers for Controlling the ELC	810
17.3.1	Event output destination select register n (ELSELRn) (n = 00 to 08, 10 to 19)	811
17.4	ELC Operation	814
18.	Interrupt Functions	816
18.1	Interrupt Function Types	816
18.2	Interrupt Sources and Configuration	816
18.3	Registers for Controlling the Interrupt Functions	822
18.3.1	Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)	826
18.3.2	Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)	828
18.3.3	Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)	830
18.3.4	External interrupt rising edge enable registers (EGP0, EGP1), External interrupt falling edge enable registers (EGN0, EGN1)	834
18.3.5	Program status word (PSW)	836
18.4	Interrupt Servicing Operations	837
18.4.1	Acknowledgment of maskable interrupt requests	837
18.4.2	Software interrupt request acknowledgment	840
18.4.3	Multiple interrupt servicing	840
18.4.4	Interrupt request held pending	844
19.	Key Interrupt Function	845
19.1	Functions of the Key Interrupt	845
19.2	Configuration of the Key Interrupt	845
19.3	Registers for Controlling the Key Interrupt	847
19.3.1	Key return control register (KRCTL)	847
19.3.2	Key return mode register 0 (KRM0)	848
19.3.3	Key return flag register (KRF)	849
19.3.4	Port mode register 7 (PM7)	850
20.	Standby Function	851
20.1	Standby Function	851
20.2	Registers for Controlling the Standby Function	852
20.2.1	Standby mode release setting register (WKUPMD)	853
20.3	Standby Function Operation	854
20.3.1	HALT mode	854
20.3.2	STOP mode	860
20.3.3	SNOOZE mode	866
21.	Reset Function	870
21.1	Timing of Reset Operation	872
21.2	Registers for Controlling the Reset Function	876
21.2.1	Reset control flag register (RESF)	876
21.2.2	Power-on-reset status register (PORSR)	879
21.2.3	Peripheral reset control register 0 (PRR0)	880
21.2.4	Peripheral reset control register 1 (PRR1)	881

22.	Power-on-Reset Circuit (POR)	882
22.1	Functions of Power-on-reset Circuit	882
22.2	Configuration of Power-on-reset Circuit	883
22.3	Operation of Power-on-reset Circuit	883
23.	Voltage Detector (LVD)	886
23.1	Functions of Voltage Detector	886
23.2	Configuration of Voltage Detector	887
23.3	Registers for Controlling the Voltage Detector	888
23.3.1	Voltage detection register (LVIM)	888
23.3.2	LVD detection flag clearing register (LVDFCLR)	889
23.3.3	Voltage detection level register (LVIS)	890
23.4	Operation of Voltage Detector	892
23.4.1	When used as reset mode	892
23.4.2	When used as interrupt mode	895
23.5	Points for Caution when the Voltage Detector is to be Used	898
24.	Safety Functions	901
24.1	Overview of Safety Functions	901
24.2	Registers for Controlling the Safety Functions	902
24.3	Operation of Safety Functions	902
24.3.1	Flash memory CRC operation function (high-speed CRC)	902
24.3.1.1	Flash memory CRC control register (CRC0CTL)	903
24.3.1.2	Flash memory CRC operation result register (PGCRCL)	904
24.3.2	CRC operation (general-purpose CRC)	906
24.3.2.1	CRC input register (CRCIN)	906
24.3.2.2	CRC data register (CRCD)	907
24.3.3	Flash memory guard function	908
24.3.3.1	Code flash memory guard register (GFLASH0)	908
24.3.3.2	Data flash memory guard register (GFLASH1)	909
24.3.3.3	Flash security area guard register (GFLASH2)	910
24.3.4	RAM parity error detection	911
24.3.4.1	RAM parity error control register (RPECTL)	911
24.3.5	RAM guard function	913
24.3.5.1	Invalid memory access detection control register (IAWCTL)	913
24.3.6	SFR guard function	914
24.3.6.1	Invalid memory access detection control register (IAWCTL)	914
24.3.7	Illicit memory access detection	915
24.3.7.1	Invalid memory access detection control register (IAWCTL)	916
24.3.8	Guard function of invalid memory access detection control register	917
24.3.8.1	Guard register of IAWCTL register (GIAWCTL)	917
24.3.9	Frequency detection	918
24.3.9.1	Timer input select register 0 (TIS0)	919
24.3.10	Testing of the A/D converter	920
24.3.10.1	A/D test register (ADTES)	922
24.3.10.2	Analog input channel specification register (ADS)	923
24.3.11	Detection of the digital output signal level of the I/O pins	925
24.3.11.1	Port mode select register (PMS)	925

24.3.12	UART loopback	926
24.3.12.1	UART loopback select register (ULBS)	926
25.	Security Functions	928
25.1	True Random Number Generator	928
25.1.1	Function of the true random number generator	928
25.1.2	Registers for controlling the true random number generator	928
25.1.2.1	Random number seed command register 0 (TRNGSCR0)	929
25.1.2.2	Random number seed data register (TRNGSDR)	930
25.1.3	Operations of the true random number generator	931
25.2	Flash Read Protection	932
25.2.1	Function of flash read protection	932
25.2.2	Setting of flash read protection	932
25.2.3	Operation	934
25.3	Unique ID	935
25.3.1	Function of a unique ID	935
25.3.2	ASCII codes representing the product name	935
26.	SNOOZE Mode Sequencer (SMS)	937
26.1	Functions of the SNOOZE Mode Sequencer	937
26.2	Configuration of the SNOOZE Mode Sequencer	937
26.3	Registers for Controlling the SNOOZE Mode Sequencer	938
26.3.1	Peripheral enable register 1 (PER1)	938
26.3.2	Peripheral reset control register 1 (PRR1)	939
26.3.3	Sequencer instruction registers p (SMSIp) (p = 0 to 31)	940
26.3.4	Sequencer general-purpose registers n (MSGn) (n = 0 to 15)	943
26.3.5	Sequencer control register (SMSC)	945
26.3.6	Sequencer status register (SMSS)	947
26.4	Operations of the SNOOZE Mode Sequencer	948
26.4.1	Internal operations of the SNOOZE mode sequencer	948
26.4.2	Memory space allocated to the sequencer	949
26.4.3	Sequencer flags	951
26.4.4	Procedures for running the SNOOZE mode sequencer	952
26.4.5	States of the SNOOZE mode sequencer	954
26.5	Commands for Use in Processing by the Sequencer	955
26.5.1	8-bit data transfer 1	956
26.5.2	8-bit data transfer 2	957
26.5.3	16-bit data transfer 1	958
26.5.4	16-bit data transfer 2	959
26.5.5	1-bit data setting	960
26.5.6	1-bit data clearing	961
26.5.7	1-bit data transfer	962
26.5.8	Word addition	963
26.5.9	Word subtraction	964
26.5.10	Word comparison	965
26.5.11	Logical shift right	966
26.5.12	Branch 1 (SCY = 1)	967
26.5.13	Branch 2 (SCY = 0)	968
26.5.14	Branch 3 (SZ = 1)	969

26.5.15	Branch 4 (SZ = 0)	970
26.5.16	Wait	971
26.5.17	Conditional wait 1 (bit = 1)	973
26.5.18	Conditional wait 2 (bit = 0)	974
26.5.19	Termination	975
26.5.20	Interrupt plus termination	976
26.5.21	DTC activation	977
26.6	Operation in Standby Modes	978
27.	Capacitive Sensing Unit (CTS2La)	979
27.1	Overview	981
27.2	Registers for Controlling the Capacitive Sensing Unit	983
27.2.1	Peripheral enable register 1 (PER1)	984
27.2.2	Peripheral reset control register 1 (PRR1)	984
27.2.3	CTS2 control registers AL and AH (CTS2CRAL, CTS2CRAH)	985
27.2.4	CTS2 control registers BL and BH (CTS2CRBL, CTS2CRBH)	992
27.2.5	CTS2 measurement channel registers L and H (CTS2MCHL, CTS2MCHH)	995
27.2.6	CTS2 channel enable control registers AL, AH, BL, and BH (CTS2CHACAL, CTS2CHACAH, CTS2CHACBL, CTS2CHACBH)	997
27.2.7	CTS2 channel transmit/receive control registers AL, AH, BL, and BH (CTS2CHTRCAL, CTS2CHTRCAH, CTS2CHTRCBL, CTS2CHTRCBH)	999
27.2.8	CTS2 status register L (CTS2SURL)	1001
27.2.9	CTS2 sensor offset registers 0 and 1 (CTS2USO0, CTS2USO1)	1004
27.2.10	CTS2 sensor counter registers L and H (CTS2USC, CTS2UUC)	1006
27.2.11	CTS2 calibration registers L and H (CTS2UDBGR0, CTS2UDBGR1)	1007
27.2.12	CTS2 sensor unit clock control registers AL, AH, BL, and BH (CTS2SUCLK0, CTS2SUCLK1, CTS2SUCLK2, CTS2SUCLK3)	1010
27.2.13	CTS2 trimming registers AL and AH (CTS2UTRIM0, CTS2UTRIM1)	1012
27.2.14	CTS2 trimming registers BL and BH (CTS2UTRIM2, CTS2UTRIM3)	1014
27.3	Usage Notes of the Capacitive Sensing Unit	1015
28.	Regulator	1016
28.1	Overview	1016
29.	Option Bytes	1017
29.1	Functions of Option Bytes	1017
29.1.1	User option bytes (000C0H to 000C2H or 020C0H to 020C2H)	1017
29.1.2	On-chip debug option byte (000C3H or 020C3H)	1018
29.2	Format of User Option Bytes	1019
29.3	Format of On-chip Debug Option Byte	1024
29.4	Setting of Option Bytes	1025
30.	Flash Memory	1026
30.1	Serial Programming Using Flash Memory Programmer	1027
30.1.1	Programming environment	1029
30.1.2	Communications mode	1029
30.2	Serial Programming Using External Device (that Incorporates UART)	1030
30.2.1	Programming environment	1030
30.2.2	Communications mode	1031

30.3	Handling of Pins on the Board	1032
30.3.1	P40/TOOL0 pin	1032
30.3.2	RESET pin	1033
30.3.3	Port pins	1034
30.3.4	REGC pin	1034
30.3.5	Power supply	1034
30.4	Programming Method	1035
30.4.1	Serial programming procedure	1035
30.4.2	Flash memory programming mode	1036
30.4.3	Selecting communications mode	1037
30.4.4	Communications commands	1038
30.5	Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values)	1040
30.6	Self-Programming	1041
30.6.1	Self-programming procedure	1042
30.6.2	Registers for controlling the flash memory	1043
30.6.2.1	Flash address pointer registers H and L (FLAPH, FLAPL)	1044
30.6.2.2	Flash end address pointer registers H and L (FLSEDH, FLSEDL)	1045
30.6.2.3	Flash write buffer registers H and L (FLWH, FLWL)	1047
30.6.2.4	Flash protect command register (PFCMD)	1048
30.6.2.5	Flash status register (PFS)	1049
30.6.2.6	Flash programming mode control register (FLPMC)	1050
30.6.2.7	Flash area selection register (FLARS)	1051
30.6.2.8	Flash memory sequencer initial setting register (FSSET)	1052
30.6.2.9	Flash memory sequencer control register (FSSQ)	1054
30.6.2.10	Flash extra area sequencer control register (FSSE)	1056
30.6.2.11	Flash registers initialization register (FLRST)	1058
30.6.2.12	Flash memory sequencer status registers H and L (FSASTH, FSASTL)	1059
30.6.2.13	Flash security flag monitoring register (FLSEC)	1061
30.6.2.14	Flash FSW monitoring register E (FLFSWE)	1062
30.6.2.15	Flash FSW monitoring register S (FLFSWS)	1063
30.6.2.16	Data flash control register (DFLCTL)	1064
30.6.2.17	Interrupt vector jump enable register (VECTCTRL)	1065
30.6.2.18	Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)	1066
30.6.3	Setting the flash memory control mode	1067
30.6.3.1	Procedure for executing the specific sequence	1067
30.6.3.2	Procedure for entry to the code flash memory programming mode	1068
30.6.3.3	Procedure for entry to the data flash memory programming mode	1068
30.6.3.4	Procedure for entry to the non-programmable mode	1068
30.6.4	Initializing the registers for use with the flash memory sequencer	1069
30.6.5	Setting the operating frequency of the flash memory sequencer	1069
30.6.6	Rewriting the flash memory	1070
30.6.6.1	Overview	1070
30.6.6.2	Selecting the area to be rewritten	1070
30.6.6.3	Commands for use with the code/data flash memory area sequencer	1070
30.6.6.4	Operations for rewriting the code flash memory area	1071
30.6.6.5	Operations for rewriting the data flash memory area	1073
30.6.6.6	Commands for use with the extra area sequencer	1075
30.6.6.7	Operations for rewriting the extra area	1075

30.6.6.8	Data to be set for the commands for use with the extra area sequencer	1076
30.6.6.9	Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas	1080
30.6.6.10	Procedure for forcibly terminating a command for use with the code/data flash memory area sequencer	1080
30.6.7	Interrupts in code flash memory programming mode	1081
30.6.7.1	Overview	1081
30.6.7.2	Operation to change the branch destinations of the interrupts	1081
30.6.7.3	Operation to change the interrupt branch destination	1082
30.6.8	Example of executing the commands to rewrite the flash memory areas	1083
30.6.8.1	Example of executing the commands to rewrite the code flash memory area	1083
30.6.8.2	Example of executing the commands to rewrite the data flash memory area	1084
30.6.8.3	Example of executing the commands to rewrite the extra area	1085
30.6.9	Notes on self-programming	1086
30.7	Boot Swap Function	1087
30.8	Flash Shield Window Function	1089
30.9	Security Settings	1090
30.10	Data Flash Memory	1093
30.10.1	Overview of the data flash memory	1093
30.10.2	Procedure for accessing the data flash memory	1094
31.	On-chip Debugging	1095
31.1	Connection between the E2 or E2 Lite and RL78/G22	1095
31.2	Connection between the External Device that Incorporates UART and RL78/G22	1096
31.3	Security Settings for On-Chip Debugging	1096
31.4	Allocation of Memory Spaces to User Resources	1096
32.	BCD Correction Circuit	1098
32.1	BCD Correction Circuit Function	1098
32.2	Register for Controlling the BCD Correction Circuit	1098
32.2.1	BCD correction result register (BCDADJ)	1098
32.3	BCD Correction Circuit Operation	1099
33.	Instruction Set	1101
33.1	Conventions Used in Operation List	1102
33.1.1	Operand identifiers and specification methods	1102
33.1.2	Description of operation column	1103
33.1.3	Description of flag operation column	1104
33.1.4	PREFIX instruction	1104
33.2	Operation List	1105
34.	Electrical Characteristics for TA = -40 to +105°C	1123
34.1	Absolute Maximum Ratings	1123
34.2	Characteristics of the Oscillators	1125
34.2.1	Characteristics of the X1 oscillator	1125
34.2.2	Characteristics of the XT1 oscillator	1125
34.2.3	Characteristics of the On-chip Oscillators	1126
34.3	DC Characteristics	1127
34.3.1	Characteristics of Pins	1127

34.3.2	Characteristics of the supply current	1132
34.4	AC Characteristics	1138
34.5	Characteristics of the Peripheral Functions	1142
34.5.1	Serial array unit	1142
34.5.2	Serial interface UARTA	1171
34.5.3	Serial interface IICA	1172
34.6	Characteristics of the Analog Circuits	1175
34.6.1	Characteristics of the A/D converter	1175
34.6.2	Characteristics of the temperature sensor and internal reference voltage	1179
34.6.3	Characteristics of the POR circuit	1179
34.6.4	Characteristics of the LVD circuit	1180
34.6.5	Characteristics of the rising slope of the power supply voltage	1182
34.7	Characteristics of Retention of RAM Data	1183
34.8	Characteristics of Flash Memory Programming	1183
34.9	Dedicated Flash Memory Programmer Communication (UART)	1185
34.10	Timing of Entry to Flash Memory Programming Modes	1185
35.	Package Drawings	1186
35.1	16-pin Products	1186
35.2	20-pin Products	1187
35.3	24-pin Products	1188
35.4	25-pin Products	1189
35.5	30-pin Products	1190
35.6	32-pin Products	1191
35.7	36-pin Products	1193
35.8	40-pin Products	1194
35.9	44-pin Products	1195
35.10	48-pin Products	1196
APPENDIX A	Revision History	1198
A.1	Major Revisions in This Edition	1198

Section 1 Outline

1.1 Features

Ultra-low power consumption technology

- VDD = single power supply voltage of 1.6 to 5.5 V
- HALT mode
- STOP mode
 - High-speed wakeup from the STOP mode is possible.
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- The minimum instruction execution time can be changed from high to ultra-low speed.
 - High speed: 0.03125 μ s at 32 MHz operation with the high-speed on-chip oscillator clock
 - Ultra-low speed: 30.5 μ s at 32.768 kHz operation with the subsystem clock
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register $\times$ 8) $\times$ 4 banks
- On-chip RAM: 4 KB

Code flash memory

- Code flash memory: 32 or 64 KB
- Block size: 2 KB
- Security function: Prohibition of block erase and rewriting
- On-chip debugging
- Self-programming with boot swapping and flash shield window

Data flash memory

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (typ.)

High-speed on-chip oscillator

- Selectable from among 32 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: $\pm 1.0\%$ (VDD = 1.8 to 5.5 V, TA = -20 to +85°C)

Middle-speed on-chip oscillator

- Selectable from among 4 MHz, 2 MHz, and 1 MHz with adjustability

Low-speed on-chip oscillator

- 32.768 kHz (typ.) with adjustability

Operating ambient temperature

- T_A = -40 to +85°C (2D: Consumer applications)
- T_A = -40 to +105°C (3C: Industrial applications)

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detectors (LVD0 and LVD1)

Data transfer controller (DTC)

- Transfer modes: Normal transfer mode, repeat transfer mode, block transfer mode
- Activation sources: Activated by interrupt sources.
- Chain transfer function

SNOOZE mode sequencer (SMS)

- Calculations and comparison of values by the commands for use in processing by the sequencer can realize intermittent operations where the RL78/G22 does not have to return to normal operation.
- Sequentially handling a total of 32 processes with the use of desired commands from among 21 different ones
- The SNOOZE mode sequencer offers operation with low power consumption without using the CPU, flash memory, and RAM.

Event link controller (ELC)

- Event signals can be set up between specified peripheral functions.

Serial interface

- Simplified SPI (CSI^{Note}): 1 to 5 channels
- UART/UART (LIN-bus supported)/UARTA: 1 to 4 channels
- I²C/Simplified I²C: 2 to 6 channels

Timers

- 16-bit timer: 1 to 8 channels
- 32-bit interval timer: 1 channel in 32-bit counter mode
2 channels in 16-bit counter mode
4 channels in 8-bit counter mode
- Realtime clock: 1 channel (counting of one second to 99 years, alarm interrupt, and clock correction)
- Watchdog timer: 1 channel (operates with the dedicated low-speed on-chip oscillator clock)

A/D converter

- 8-/10-bit resolution A/D converter (V_{DD} = 1.6 to 5.5 V)
- Analog input: 3 to 10 channels
- Internal reference voltage (1.48 V) and temperature sensor

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

Capacitive sensing unit

- CTSU2La operating voltage condition: $V_{DD} = 1.8$ to 5.5 V
- Self-capacitance method: A single pin configures a single key, supporting up to 29 keys
- Mutual capacitance method: Matrix configuration with 8×8 pins, supporting up to 64 keys

Input/output port pins

- Number of port pins: 12 to 44
 - N-ch open drain I/O pins (withstand voltage of 6 V): 0 to 4
 - N-ch open drain I/O pins (withstand voltage of V_{DD}): 4 to 13
- Can be set to N-ch open drain or TTL input buffer, and use of an on-chip pull-up resistor can be specified.
- Connectable to a device with different voltage (1.8, 2.5, or 3 V)

Others

- Binary-coded decimal (BCD) correction circuit
- Key interrupt input
- Clock output/buzzer output controller

Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

O ROM, RAM capacities

Flash ROM	Data flash memory	RAM	RL78/G22				
			16 pins	20 pins	24 pins	25 pins	30 pins
64 KB	2 KB	4 KB	R7F102G4E	R7F102G6E	R7F102G7E	R7F102G8E	R7F102GAE
32 KB	2 KB	4 KB	R7F102G4C	R7F102G6C	R7F102G7C	R7F102G8C	R7F102GAC

Flash ROM	Data flash memory	RAM	RL78/G22				
			32 pins	36 pins	40 pins	44 pins	48 pins
64 KB	2 KB	4 KB	R7F102GBE	R7F102GCE	R7F102GEE	R7F102GFE	R7F102GGE
32 KB	2 KB	4 KB	R7F102GBC	R7F102GCC	R7F102GEC	R7F102GFC	R7F102GGC

1.2 List of Part Numbers

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G22

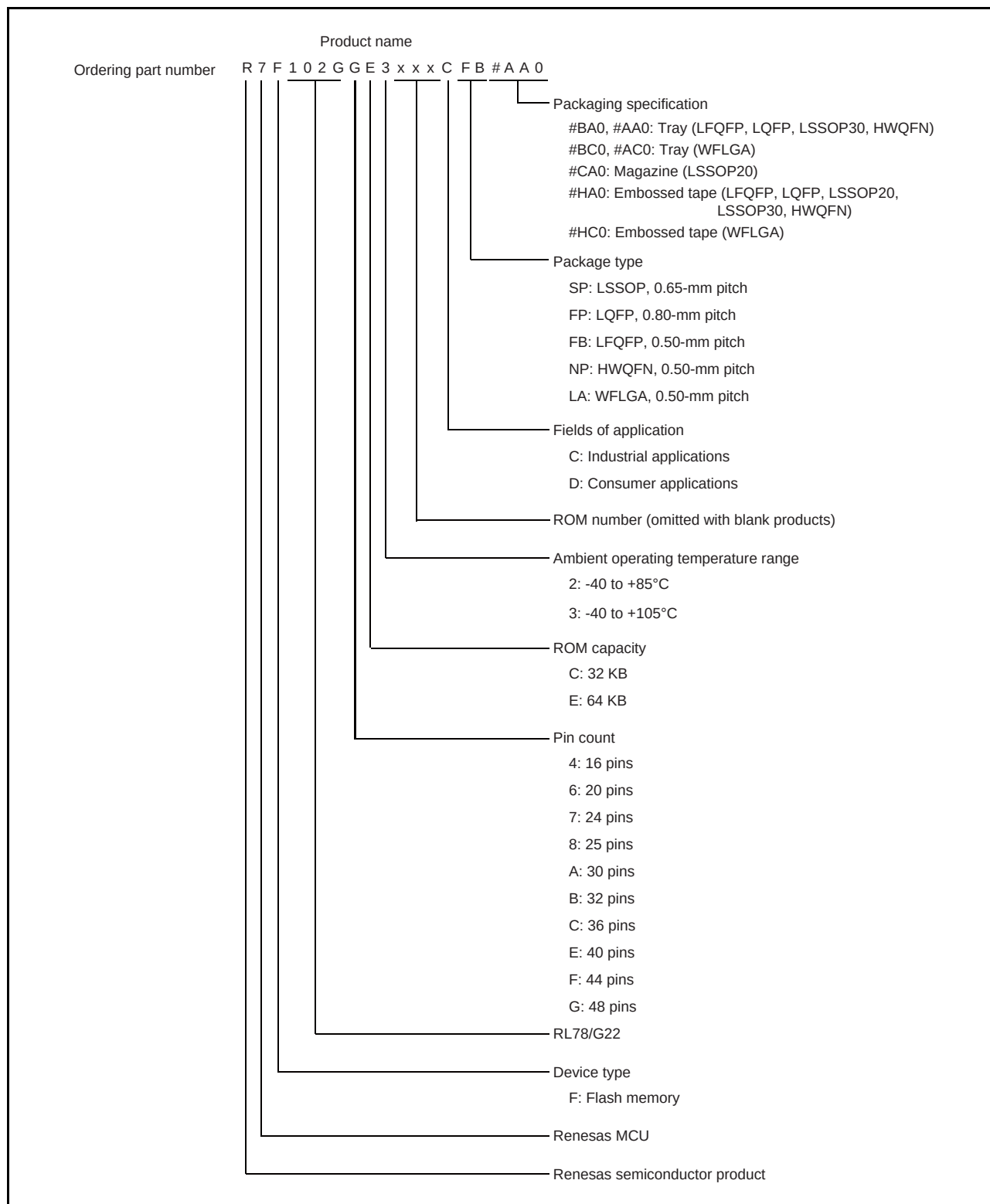


Table 1 - 1 List of Ordering Part Numbers

Pin Count	Package	Fields of Application Note 1	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
16 pins	16-pin plastic HWQFN (3 × 3 mm, 0.50-mm pitch)	C	R7F102G4C3CNP, R7F102G4E3CNP	#AA0, #BA0, #HA0	Note 2
		D	R7F102G4C2DNP, R7F102G4E2DNP		
20 pins	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	C	R7F102G6C3CSP, R7F102G6E3CSP	#CA0, #HA0	PLSP0020JB-A
		D	R7F102G6C2DSP, R7F102G6E2DSP		
24 pins	24-pin plastic HWQFN (4 × 4 mm, 0.5 mm pitch)	C	R7F102G7C3CNP, R7F102G7E3CNP	#AA0, #BA0, #HA0	PWQN0024KG-A
		D	R7F102G7C2DNP, R7F102G7E2DNP		
25 pins	25-pin plastic WFLGA (3 × 3 mm, 0.5 mm pitch)	C	R7F102G8C3CLA, R7F102G8E3CLA	#AA0, #BA0, #HA0	Note 3
		D	R7F102G8C2DLA, R7F102G8E2DLA		
30 pins	30-pin plastic LSSOP (7.62 mm (300), 0.65-mm pitch)	C	R7F102GAC3CSP, R7F102GAE3CSP	#AA0, #BA0, #HA0	PLSP0030JB-B
		D	R7F102GAC2DSP, R7F102GAE2DSP		
32 pins	32-pin plastic HWQFN (5 × 5 mm, 0.50-mm pitch)	C	R7F102GBC3CNP, R7F102GBE3CNP	#AA0, #BA0, #HA0	PWQN0032KE-A
		D	R7F102GBC2DNP, R7F102GBE2DNP		
	32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)	C	R7F102GBC3CFP, R7F102GBE3CFP	#AA0, #BA0, #HA0	PLQP0032GB-A
		D	R7F102GBC2DFP, R7F102GBE2DFP		
36 pins	36-pin plastic WFLGA (4 × 4 mm, 0.50-mm pitch)	C	R7F102GCC3CLA, R7F102GCE3CLA	#BC0, #AC0, #HC0	Note 3
		D	R7F102GCC2DLA, R7F102GCE2DLA		
40 pins	40-pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)	C	R7F102GEC3CNP, R7F102GEE3CNP	#AA0, #BA0, #HA0	PWQN0040KD-A
		D	R7F102GEC2DNP, R7F102GEE2DNP		
44 pins	44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)	C	R7F102GFC3CFP, R7F102GFE3CFP	#AA0, #BA0, #HA0	PLQP0044GC-A
		D	R7F102GFC2DFP, R7F102GFE2DFP		
48 pins	48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)	C	R7F102GGC3CFB, R7F102GGE3CFB	#AA0, #BA0, #HA0	PLQP00048KB-B
		D	R7F102GGC2DFB, R7F102GGE2DFB		
	48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)	C	R7F102GGC3CNP, R7F102GGE3CNP	#AA0, #BA0, #HA0	PWQN0048KC-A
		D	R7F102GGC2DNP, R7F102GGE2DNP		

Note 1. For the fields of application, see **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G22**.

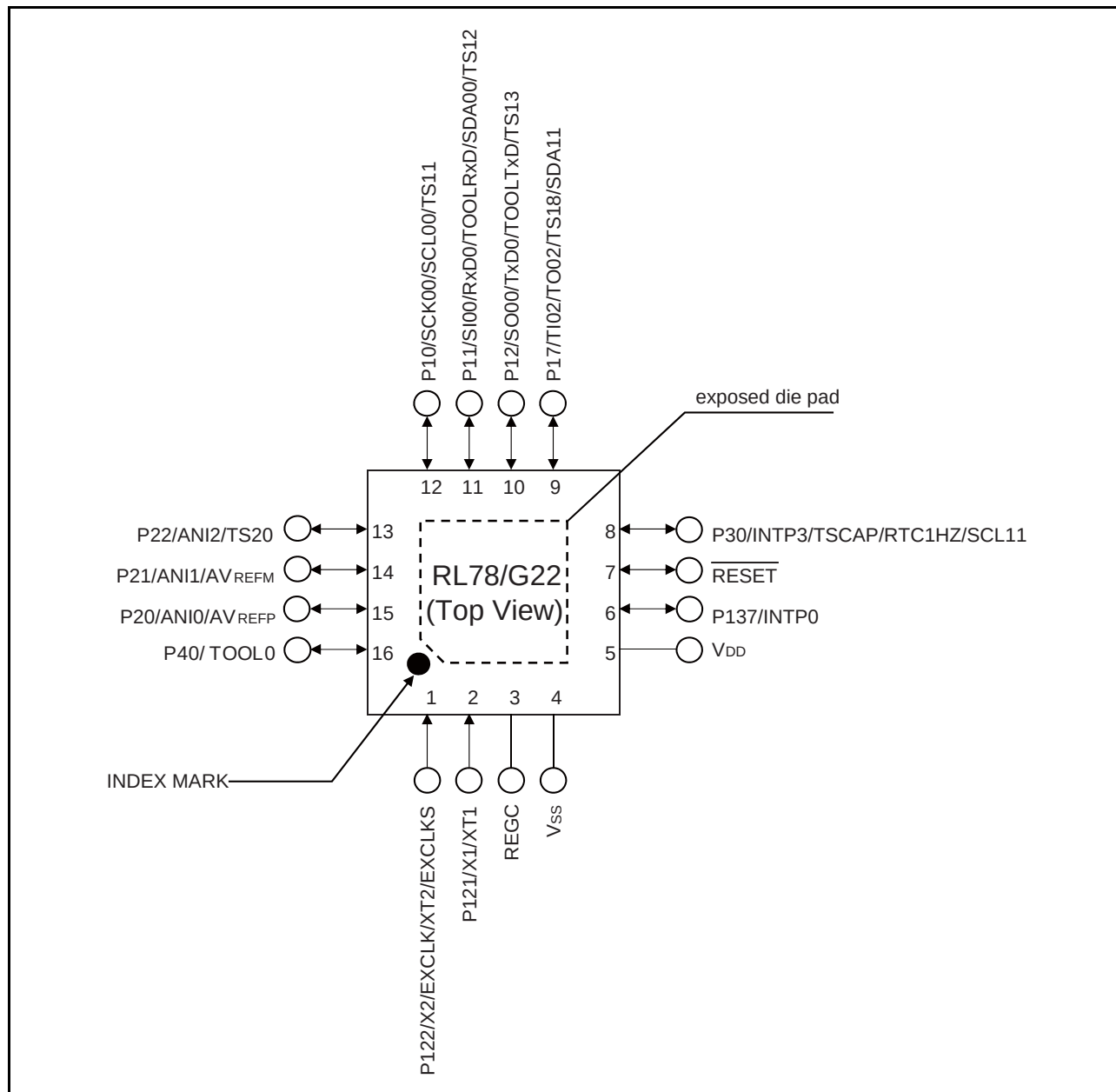
Note 2. The 16-pin plastic QFN products are in planning. Contact a Renesas Electronics sales office for details.

Note 3. The 25- and 36-pin plastic WFLGA products are in planning. Contact a Renesas Electronics sales office for details.

1.3 Pin Configuration (Top View)

1.3.1 16-pin products

- 16-pin plastic HWQFN (3 × 3 mm, 0.5-mm pitch)

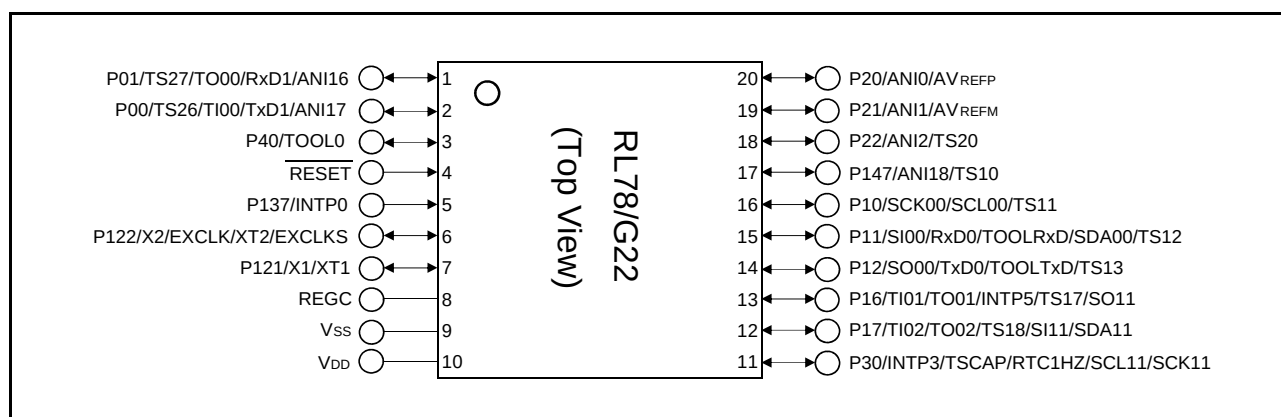


Caution Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF).

Remark For pin identification, see 1.4 Pin Identification.

1.3.2 20-pin products

- 20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)

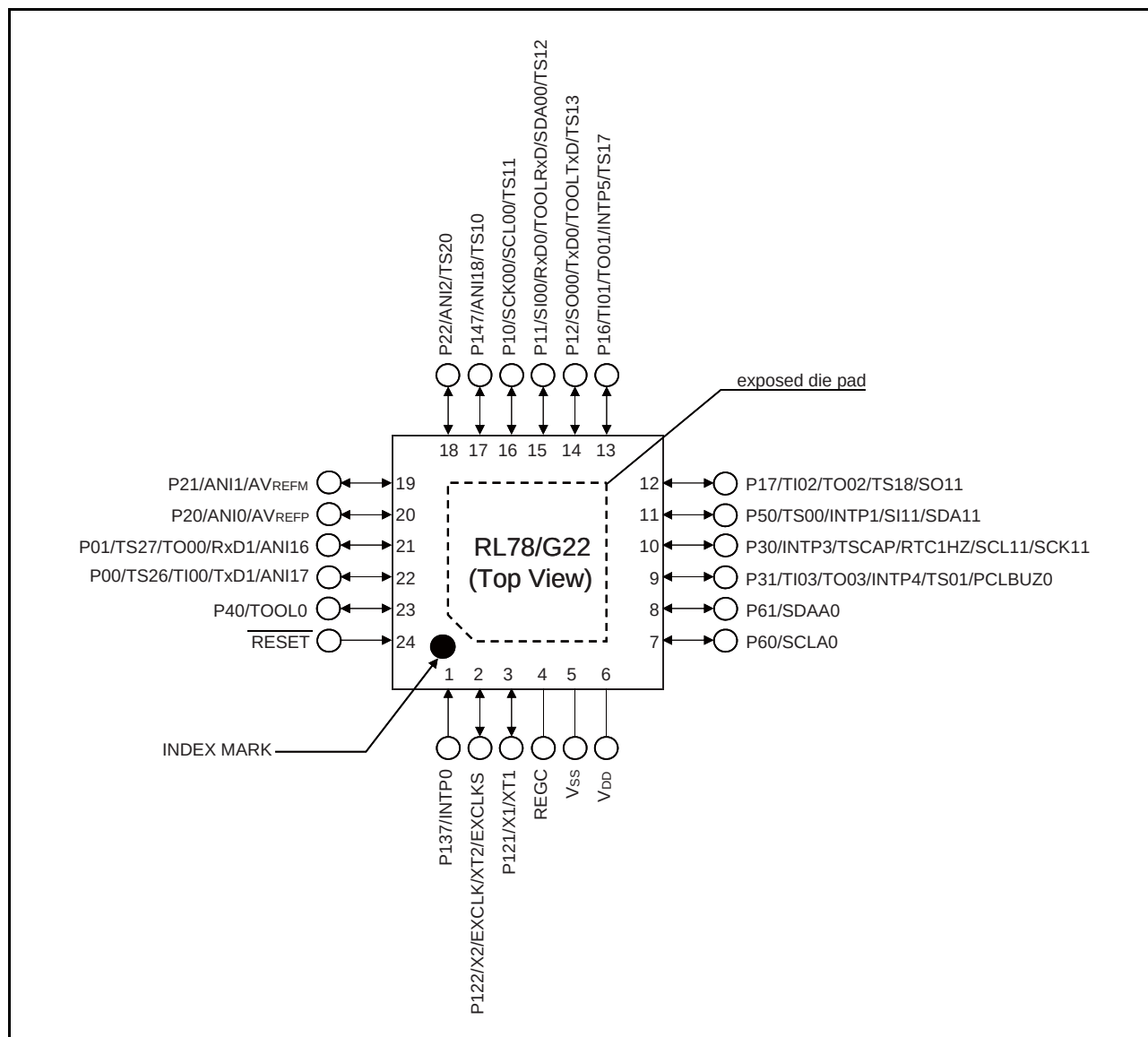


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remark For pin identification, see 1.4 Pin Identification.

1.3.3 24-pin products

- 24-pin plastic HWQFN (4 × 4 mm, 0.5-mm pitch)

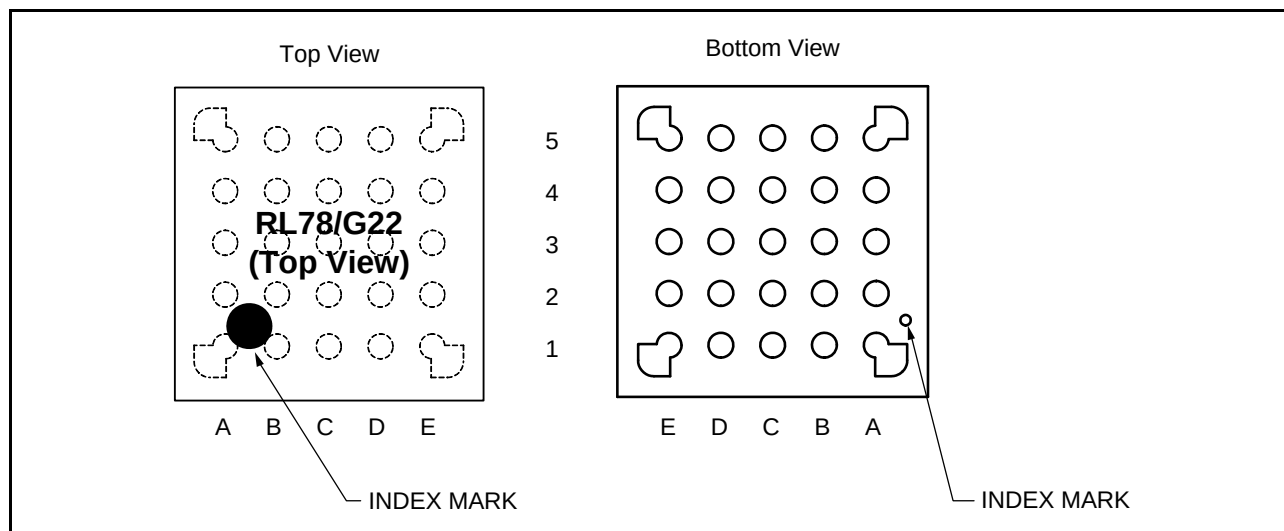


Caution Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF).

Remark For pin identification, see 1.4 Pin Identification.

1.3.4 25-pin products

- 25-pin plastic WFLGA (3 × 3 mm, 0.5-mm pitch)



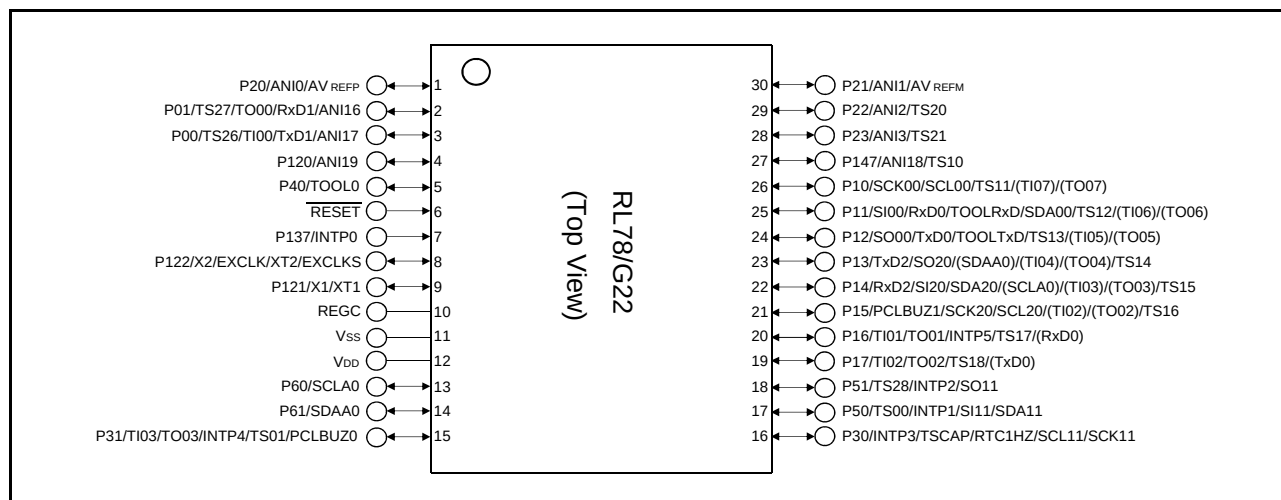
	A	B	C	D	E	
5	P40/TOOL0	RESET	P01/TS27/TO00/RxD1/ANI16	P22/ANI2/TS20	P147/ANI18/TS10	5
4	P122/X2/EXCLK/XT2/EXCLKS	P137/INTP0	P00/TS26/TI00/TxD1/ANI17	P21/ANI1/AVREFM	P10/SCK00/SCL00/TS11	4
3	P121/X1/XT1	VDD	P20/ANI0/AVREFP	P12/SO00/TxD0/TOOLTxD/TS13	P11/SI00/RxD0/TOOLRxD/SDA00/TS12	3
2	REGC	VSS	P30/INTP3/TSCAP/RTC1HZ/SCL11/SCK11	P17/TI02/TO02/TS18/SO11	P50/TS00/INTP1/SI11/SDA11	2
1	P60/SCLA0	P61/SDAA0	P31/TI03/TO03/INTP4/TS01/PCLBUZ0	P16/TI01/TO01/INTP5/TS17	P130/TS19	1
	A	B	C	D	E	

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remark For pin identification, see 1.4 Pin Identification.

1.3.5 30-pin products

- 30-pin plastic LSSOP (7.62 mm (300), 0.65-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μ F).

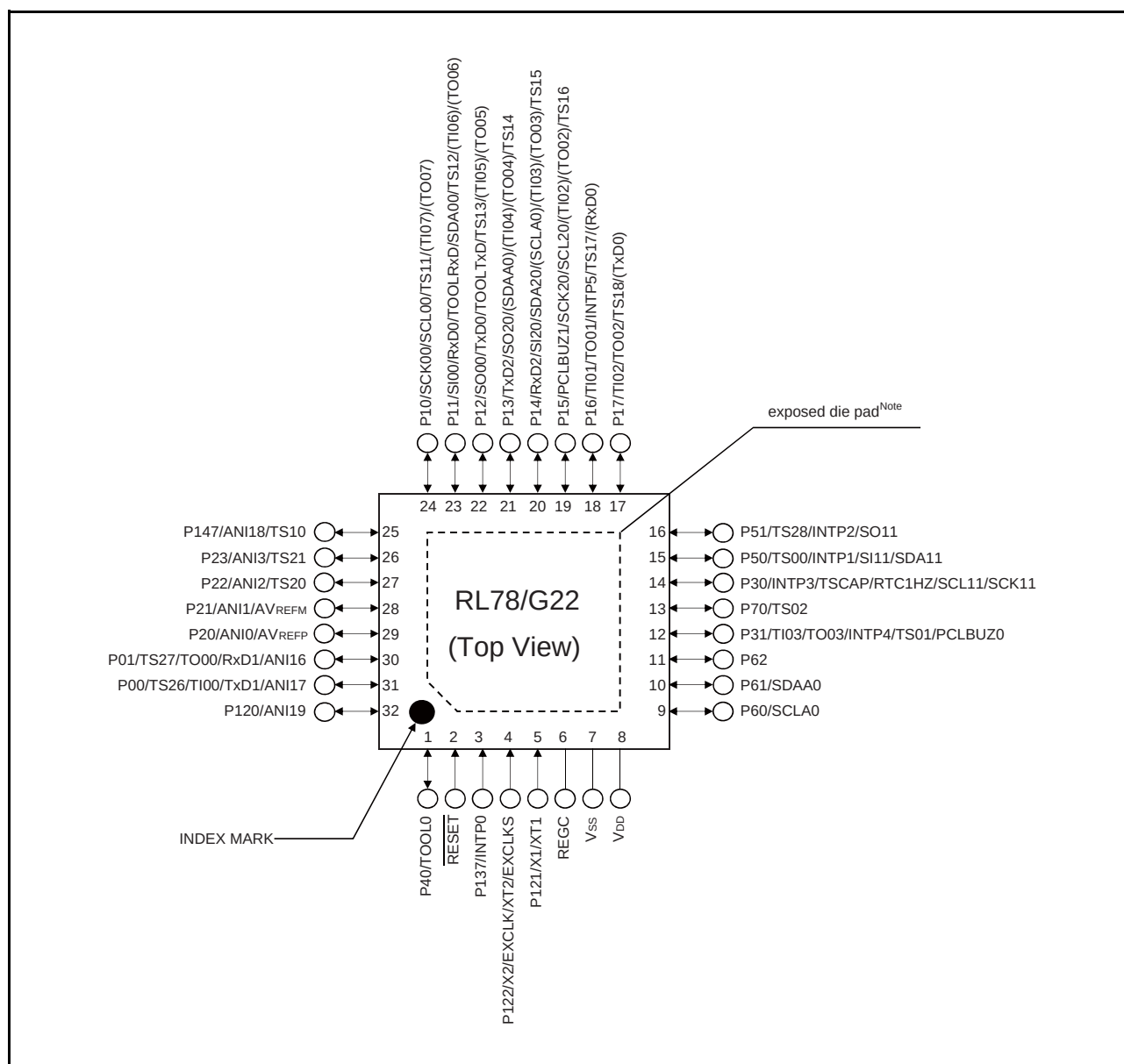
Remark 1. For pin identification, see 1.4 Pin Identification.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Refer to Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR).

1.3.6 32-pin products

- 32-pin plastic HWQFN (5 × 5 mm, 0.50-mm pitch)
- 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)



Note The 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch) products do not have an exposed die pad.

Caution Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF).

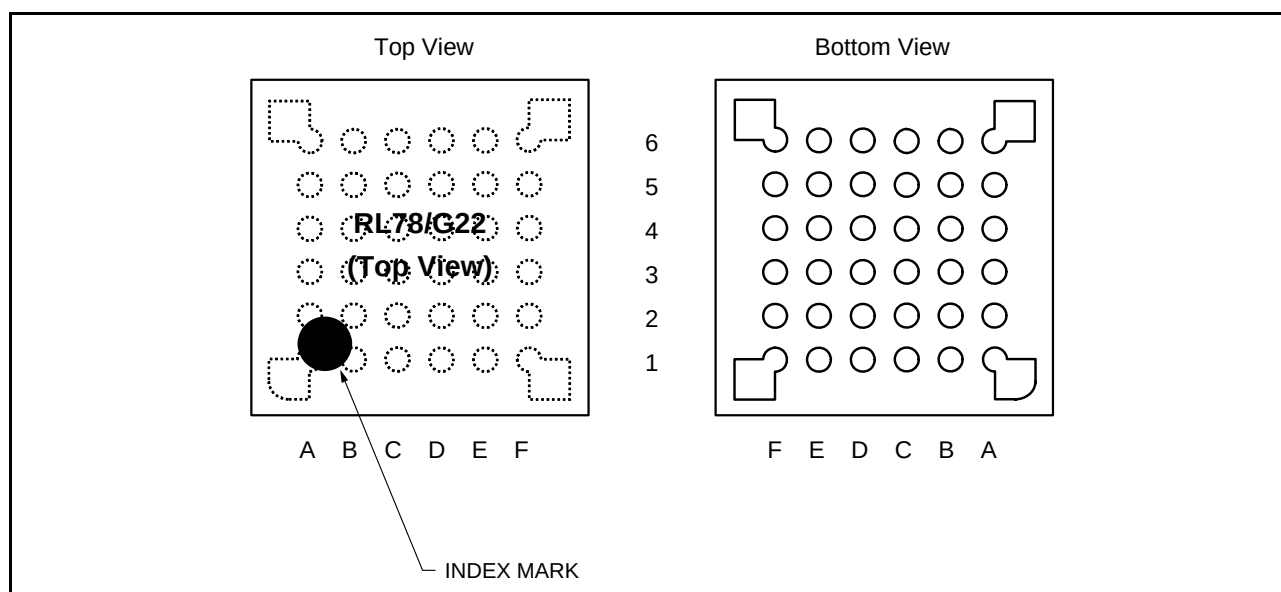
Remark 1. For pin identification, see 1.4 Pin Identification.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).
Refer to **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to V_{SS}.

1.3.7 36-pin products

- 36-pin plastic WFLGA (4 × 4 mm, 0.50-mm pitch)



	A	B	C	D	E	F
6	P60/SCLA0	VDD	P121/X1/XT1	P122/X2/EXCLK/ XT2/EXCLKS	P137/INTP0	P40/TOOL0
5	P62	P61/SDAA0	VSS	REGC	RESET	P120/ANI19
4	P72/TS04/SO21/ TxDA0	P71/TS03/SI21/ SDA21/RxDA0	P14/RxD2/SI20/ SDA20/(SCLA0)/ (TI03)/(TO03)/TS15	P31/TI03/TO03/ INTP4/TS01/ PCLBUZ0	P00/TS26/TI00/ TxD1	P01/TS27/TO00/ RxD1
3	P50/TS00/INTP1/ SI11/SDA11	P70/TS02/SCK21/ SCL21	P15/PCLBUZ1/ SCK20/SCL20/ (TI02)/(TO02)/TS16	P22/ANI2/TS20	P20/ANI0/AVREFP	P21/ANI1/AVREFM
2	P30/INTP3/TSCAP/ RTC1HZ/SCL11/ SCK11	P16/TI01/TO01/ INTP5/TS17/(RxD0)	P12/SO00/TxD0/ TOOLTxD/TS13/ (TI05)/(TO05)	P11/SI00/RxD0/ TOOLRxD/SDA00/ TS12/(TI06)/(TO06)	P24/ANI4/TS22	P23/ANI3/TS21
1	P51/TS28/INTP2/ SO11	P17/TI02/TO02/ TS18/(TxD0)	P13/TxD2/SO20/ (SDAA0)/(TI04)/ (TO04)/TS14	P10/SCK00/SCL00/ TS11/(TI07)/(TO07)	P147/ANI18/TS10	P25/ANI5/TS23

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

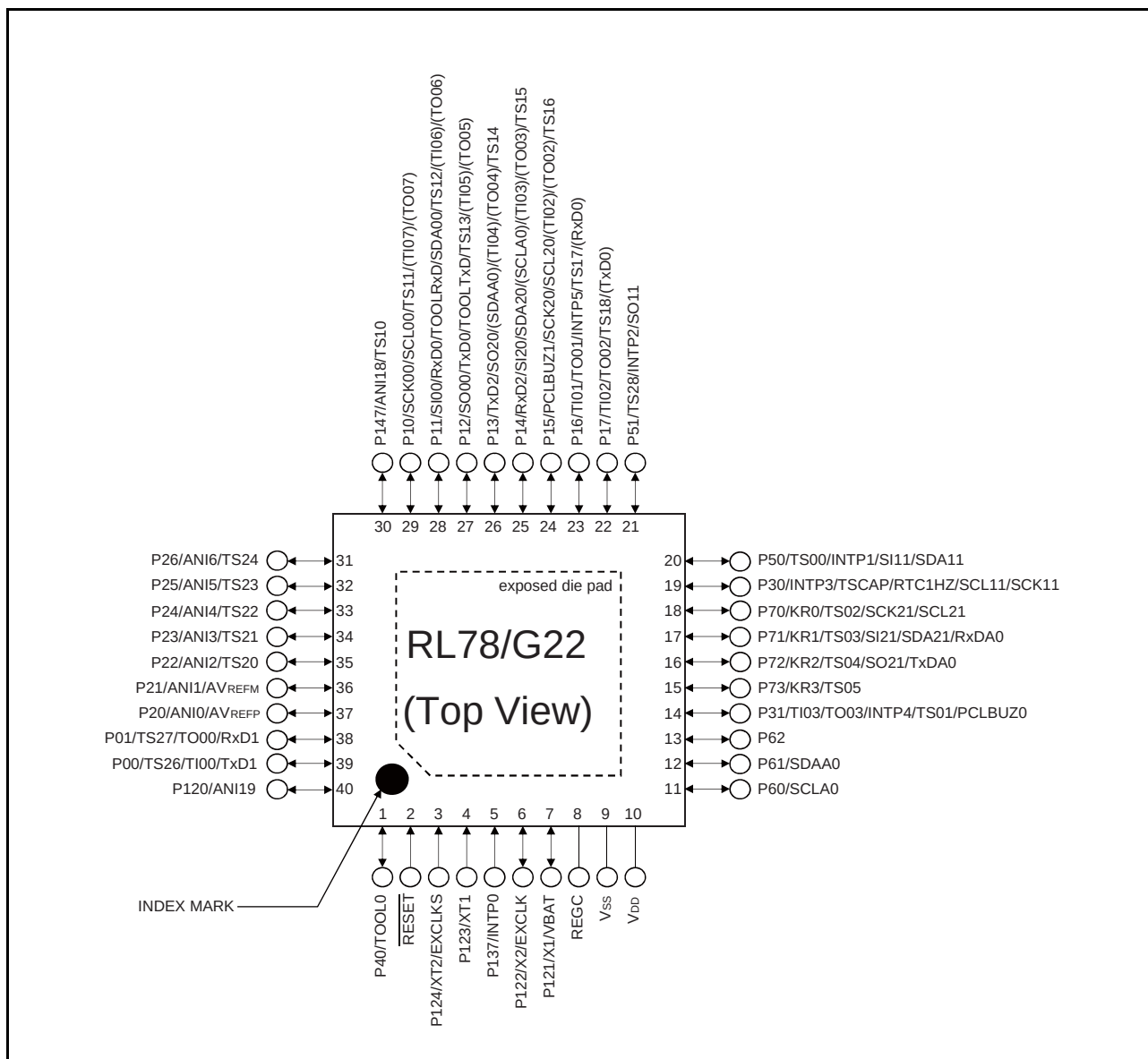
Remark 1. For pin identification, see 1.4 Pin Identification.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Refer to **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

1.3.8 40-pin products

- 40-pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see 1.4 Pin Identification.

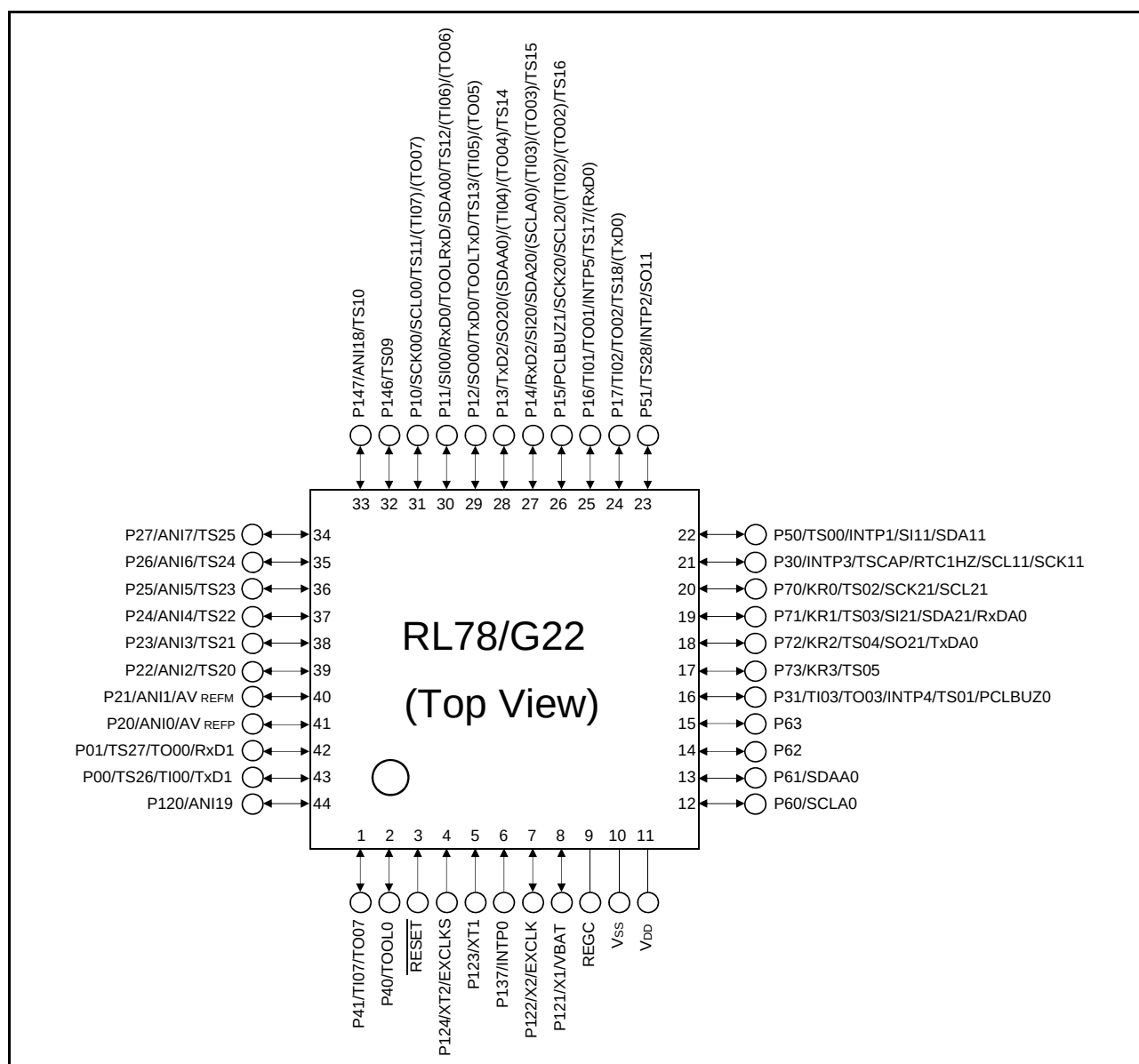
Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Refer to **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to Vss.

1.3.9 44-pin products

- 44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μ F).

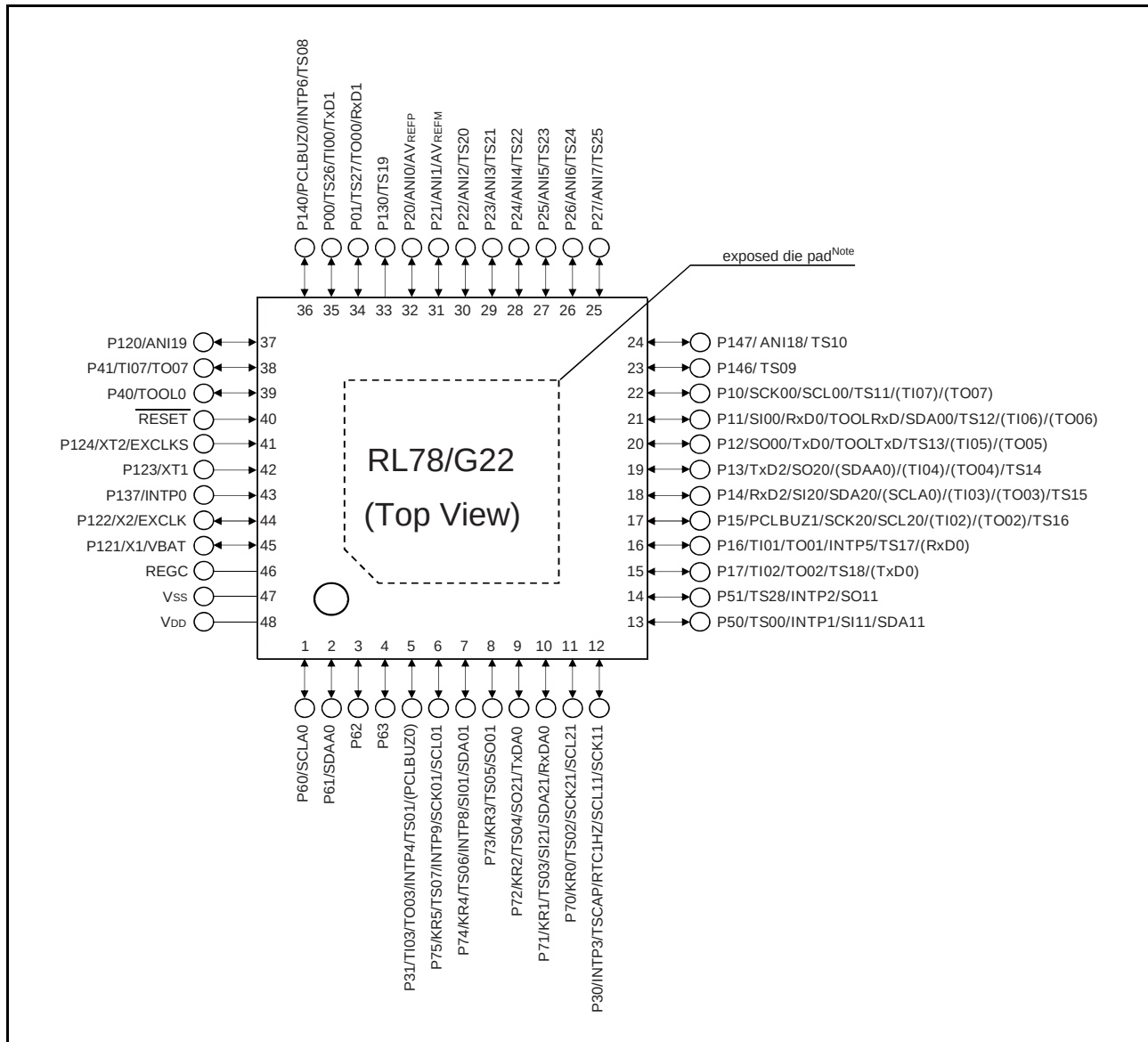
Remark 1. For pin identification, see 1.4 Pin Identification.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

Refer to **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

1.3.10 48-pin products

- 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)
- 48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)



Note The 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch) products do not have an exposed die pad.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see 1.4 Pin Identification.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

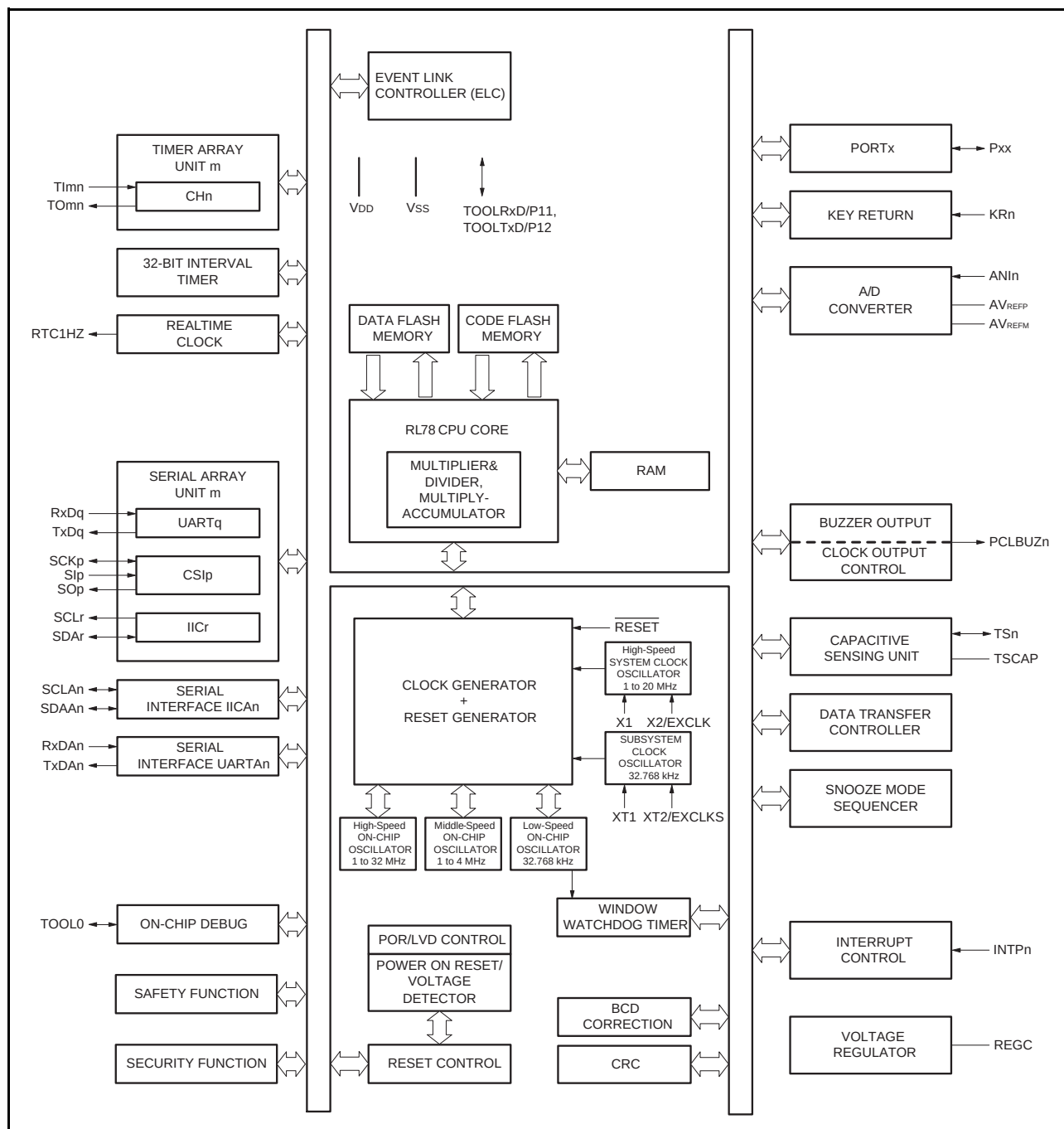
Refer to **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to Vss.

1.4 Pin Identification

ANI0 to ANI7,		RxD0 to RxD2,	
ANI16 to ANI19	: Analog input	RxDA0	: Receive data
AVREFM	: Analog reference voltage minus	SCLA0,	
AVREFP	: Analog reference voltage plus	SCK00, SCK01,	
EXCLK	: External clock input (main system clock)	SCK11, SCK20, SCK21	: Serial clock input/output
EXCLKS	: External clock input (subsystem clock)	SCL00, SCL01,	
		SCL11, SCL20,	
		SCL21	: Serial clock output
INTP0 to INTP6, INTP8,		SDAA0, SDA00,	
INTP9	: Interrupt request from peripheral	SDA01, SDA11,	
KR0 to KR5	: Key return	SDA20, SDA21	: Serial data input/output
P00, P01	: Port 0	SI00, SI01, SI11,	
P10 to P17	: Port 1	SI20, SI21	: Serial data input
P20 to P27	: Port 2	SO00, SO01	
P30, P31	: Port 3	SO11, SO20, SO21	: Serial data output
P40, P41	: Port 4	TSCAP	: Touch sensor capacitance
P50, P51	: Port 5	TI00 to TI07	: Timer input
P60 to P63	: Port 6	TO00 to TO07	: Timer output
P70 to P75	: Port 7	TOOL0	: Data input/output for tool
P120 to P124	: Port 12	TOOLRxD, TOOLTxD	: Data input/output for external device
P130, P137	: Port 13	TS00 to TS28	: Capacitive touch sensor
P140, P146, P147	: Port 14	TxD0 to TxD2	: Transmit data
PCLBUZ0, PCLBUZ1	: Programmable clock output/buzzer output	TxDA0	
REGC	: Regulator capacitance	VBAT	: Battery backup power supply
RESET	: Reset	VDD	: Power supply
RTC1HZ	: Realtime clock correction clock (1 Hz) output	VSS	: Ground
		X1, X2	: Crystal oscillator (main system clock)
		XT1, XT2	: Crystal oscillator (subsystem clock)

1.5 Block Diagram



Caution 1. The serial interface UARTA is only incorporated in the 36- to 48-pin products.

Caution 2. The key return function is only incorporated in the 40- to 48-pin products.

Remark m: Unit number, n: Channel number, p: Simplified SPI (CSI) number, q: UART number, r: Simplified I²C number, xx: Port number

1.6 Outline of Functions

Caution This outline describes the functions at the time when peripheral I/O redirection register (PIOR) is set to 00H.

(1/3)

Item		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
		R7F102G4x	R7F102G6x	R7F102G7x	R7F102G8x	R7F102GAx	R7F102GBx	R7F102GCx	R7F102GEx	R7F102GFx	R7F102GGx
Code flash memory		32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB	32 or 64 KB
Data flash memory		2 KB	2 KB	2 KB	2 KB	2 KB	2 KB	2 KB	2 KB	2 KB	2 KB
RAM		4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB	4 KB
Address space		1 MB									
CPU/peripheral hardware clock frequency (fCLK)		Main system clock HS mode: 1 to 32 MHz (VDD = 1.8 to 5.5 V) Main system clock HS mode: 1 to 4 MHz ^{Note 1} (VDD = 1.6 to 5.5 V) Main system clock LS mode: 1 to 24 MHz (VDD = 1.8 to 5.5 V) Main system clock LS mode: 1 to 4 MHz ^{Note 1} (VDD = 1.6 to 5.5 V) Main system clock LP mode: 1 to 2 MHz ^{Note 1} (VDD = 1.6 to 5.5 V) Subsystem clock: 32.768 kHz (VDD = 1.6 to 5.5 V)									
Main system clock	High-speed system clock (fMX)	1 to 20 MHz									
	High-speed on-chip oscillator clock (fIH)	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz									
	Middle-speed on-chip oscillator clock (fIM)	1 MHz, 2 MHz, 4 MHz									
Subsystem clock	Subsystem clock X (fsx)	32.768 kHz (VDD = 2.4 to 5.5 V)							32.768 kHz (VDD = 1.6 to 5.5 V)		
	Low-speed on-chip oscillator clock (fIL)	32.768 kHz (typ.)									
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)									
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (fIH))									
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.									

(2/3)

Item		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
		R7F102G4x	R7F102G6x	R7F102G7x	R7F102G8x	R7F102GAx	R7F102GBx	R7F102GCx	R7F102GEx	R7F102GFx	R7F102GGx
I/O port	Total number of pins	12	16	20	21	26	28	32	36	40	44
	CMOS I/O	11 (N-ch open drain I/O [withstand voltage of VDD]: 4)	15 (N-ch open drain I/O [withstand voltage of VDD]: 5)	17 (N-ch open drain I/O [withstand voltage of VDD]: 6)	17 (N-ch open drain I/O [withstand voltage of VDD]: 6)	23 (N-ch open drain I/O [withstand voltage of VDD]: 10)	24 (N-ch open drain I/O [withstand voltage of VDD]: 10)	28 (N-ch open drain I/O [withstand voltage of VDD]: 12)	30 (N-ch open drain I/O [withstand voltage of VDD]: 12)	33 (N-ch open drain I/O [withstand voltage of VDD]: 12)	36 (N-ch open drain I/O [withstand voltage of VDD]: 13)
	CMOS input	1	1	1	1	1	1	1	3	3	3
	CMOS output	—	—	—	1	—	—	1	1	1	1
	N-ch open drain I/O (withstand voltage of 6 V)	—	—	2	2	2	3	3	3	4	4
Timers	16-bit timer	1 channel	3 channels	4 channels	4 channels	8 channels	8 channels	8 channels	8 channels	8 channels	8 channels
	Watchdog timer	1 channel									
	Realtime clock (RTC)	1 channel									
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode									
	Timer output	1 channel	3 channels (PWM outputs: 2 ^{Note 2})	4 channels (PWM outputs: 3 ^{Note 2})		4 channels (PWM outputs: 3 ^{Note 2}), 8 channels (PWM outputs: 7 ^{Note 2}) ^{Note 3}				5 channels (PWM outputs: 4 ^{Note 2}), 8 channels (PWM outputs: 7 ^{Note 2}) ^{Note 3}	
	RTC output	1 channel									
Clock output/buzzer output		2									
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (at the 32-MHz operation with the main system clock (fMAIN)) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (at the 32.768-kHz operation with the low-speed peripheral clock (fSP))									
8-/10-bit resolution A/D converter		3 channels	6 channels	6 channels	6 channels	8 channels	8 channels	8 channels	9 channels	10 channels	10 channels

(3/3)

Item		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
		R7F102G4x	R7F102G6x	R7F102G7x	R7F102G8x	R7F102GAx	R7F102GBx	R7F102GCx	R7F102GEx	R7F102GFx	R7F102GGx
Serial interfaces	[16-pin products]										
	- Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified I²C: 1 channel										
	[20-, 24-, and 25-pin products]										
	- Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel										
	[30- and 32-pin products]										
	- Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART (UART supporting LIN-bus): 1 channel										
	[36-, 40-, and 44-pin products]										
	- Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART (UART supporting LIN-bus): 1 channel										
	[48-pin products]										
	- Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART: 1 channel - Simplified SPI (CSI): 1 channel, simplified I²C: 1 channel, UART: 1 channel - Simplified SPI (CSI): 2 channels, simplified I²C: 2 channels, UART (UART supporting LIN-bus): 1 channel										
	UART	—						1 channel	1 channel	1 channel	1 channel
	I ² C bus	—	—	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel
Data transfer controller (DTC)		14 sources	18 sources	21 sources	21 sources	28 sources	28 sources	30 sources	31 sources	31 sources	32 sources
Event link controller (ELC)		1									
SNOOZE mode sequencer (SMS)		1									
Capacitive sensing unit		5	9	11	12	16	17	21	23	25	29
Vectored interrupt sources	Internal	16	19	21	21	29	29	32	32	32	32
	External	2	3	5	5	6	6	6	7	7	10
Key interrupt		—	—	—	—	—	—	—	4	4	6
Reset		- Reset by RESET pin - Internal reset by watchdog timer - Internal reset by power-on-reset - Internal reset by voltage detectors (LVD0 and LVD1) - Internal reset by illegal instruction execution^{Note 4} - Internal reset by RAM parity error - Internal reset by illegal-memory access									
Power-on-reset circuit		Detection voltage 1.50 V (typ.)									
Voltage detector		Detection voltage - Rising edge: 1.67 to 4.00 V (6 stages) for LVD0 1.67 to 4.16 V (18 stages) for LVD1 - Falling edge: 1.63 to 3.92 V (6 stages) for LVD0 1.63 to 4.08 V (18 stages) for LVD1									
On-chip debugging		Available									
Power supply voltage		V _{DD} = 1.6 to 5.5 V (2D: Consumer applications), V _{DD} = 1.6 to 5.5 V (3C: Industrial applications)									
Operating ambient temperature		T _A = -40 to +85°C (2D: Consumer applications), T _A = -40 to +105°C (3C: Industrial applications)									

(Notes are listed on the next page.)

- Note 1.** Ensure that the operating voltage is at least 1.8 V during overwriting of the flash memory.
- Note 2.** The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves).
For details, see **7.9.3 Operation for the multiple PWM output function**.
- Note 3.** This applies when the setting of the PIOR0 bit is 1.
- Note 4.** In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

Section 2 Pin Functions

2.1 Functions of Port Pins

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 2 - 1 Pin I/O Buffer Power Supplies

16-, 20-, 24-, 25-, 30-, 32-, 36-, 40-, 44-, and 48-pin products

Power Supply	Corresponding Pins
VDD	All pins

2.1.1 16-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11	Port 1. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P12 and P17 as outputs.
P11				SI00/RxD0/TOOLRxD/SDA00/TS12	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13	
P17	8-38-2			TI02/TO02/TS18/SDA11	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 3-bit I/O port. Input or output can be specified in 1-bit units. P20 to P22 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCL11	Port 3. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	7-2-1	I/O	Input port	X1/XT1	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V _{DD} , either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

2.1.2 20-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-33-4	I/O	Analog function	TS26/ANI17/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-33-2			TS27/ANI16/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11	Port 1. 5-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P16, and P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P12 and P17 as outputs.
P11				SI00/RxD0/TOOLRx/D/SDA00/TS12	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13	
P16	8-38-1			TI01/TO01/INTP5/TS17/SO11	
P17	8-38-2			TI02/TO02/TS18/SI11/SDA11	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 3-bit I/O port. Input or output can be specified in 1-bit units. P20 to P22 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	7-2-1	I/O	Input port	X1/XT1	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V _{DD} , either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

2.1.3 24-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-33-4	I/O	Analog function	TS26/ANI17/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-33-2			TS27/ANI16/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11	Port 1. 5-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P16, and P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P12 and P17 as outputs.
P11				SI00/RxD0/TOOLRx/D/SDA00/TS12	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13	
P16	8-38-1			TI01/TO01/INTP5/TS17	
P17	8-38-2			TI02/TO02/TS18/SO11	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 3-bit I/O port. Input or output can be specified in 1-bit units. P20 to P22 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 1-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P121	7-2-1	I/O	Input port	X1/XT1	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

2.1.4 25-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-33-4	I/O	Analog function	TS26/ANI17/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-33-2			TS27/ANI16/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11	Port 1. 5-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P16, and P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P12 and P17 as outputs.
P11				SI00/RxD0/TOOLRx/D/SDA00/TS12	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13	
P16	8-38-1			TI01/TO01/INTP5/TS17	
P17	8-38-2			TI02/TO02/TS18/SO11	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 3-bit I/O port. Input or output can be specified in 1-bit units. P20 to P22 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 1-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P121	7-2-1	I/O	Input port	X1/XT1	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/XT2/EXCLKS	
P130	1-1-5	Output	Output port	TS19	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	INTP0	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

2.1.5 30-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-33-4	I/O	Analog function	TS26/ANI17/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-33-2			TS27/ANI16/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5			I/O	
P21		ANI1/AVREFM			
P22	4-35-1	ANI2/TS20			
P23	4-37-1	ANI3/TS21			
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			TS28/INTP2/SO11	
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (6-V tolerant).
P61				SDAA0	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (VDD tolerant) can be set for P120 as an output. P120 can be set for the analog pin function ^{Note} .
P121	7-2-1		Input port	X1/XT1	
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.6 32-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-33-4	I/O	Analog function	TS26/ANI17/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-33-2			TS27/ANI16/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P23	4-37-1			ANI3/TS21	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			TS28/INTP2/SO11	
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P62	12-38-1			—	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P70	7-31-2	I/O	Input port	TS02	Port 7. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (VDD tolerant) can be set for P120 as an output. P120 can be set for the analog pin function ^{Note} .
P121	7-2-1		Input port	X1/XT1	
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.7 36-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-31-3	I/O	Input port	TS26/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-31-1			TS27/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 6-bit I/O port. Input or output can be specified in 1-bit units. P20 to P25 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P23	4-37-1			ANI3/TS21	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			TS28/INTP2/SO11	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P62				—	
P70	7-31-2	I/O	Input port	TS02/SCK21/SCL21	Port 7. 3-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (VDD tolerant) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/SO21/TxDA0	
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (VDD tolerant) can be set for P120 as an output. P120 can be set for the analog pin function Note .
P121	7-2-1		Input port	X1/XT1	
P122				X2/EXCLK/XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function Note .
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.8 40-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-31-3	I/O	Input port	TS26/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-31-1			TS27/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5			I/O	
P21		ANI1/AVREFM			
P22	4-35-1	ANI2/TS20			
P23	4-37-1	ANI3/TS21			
P24	4-33-1	ANI4/TS22			
P25		ANI5/TS23			
P26		ANI6/TS24			
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			TS28/INTP2/SO11	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P62				—	
P70	7-31-2	I/O	Input port	KR0/TS02/SCK21/SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (VDD tolerant) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			KR1/TS03/SI21/SDA21/RxDA0	
P72	7-31-3			KR2/TS04/SO21/TxDA0	
P73	7-31-2			KR3/TS05	
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (VDD tolerant) can be set for P120 as an output. P120 can be set for the analog pin function ^{Note} .
P121	7-2-1		Input port	VBAT/X1	
P122				X2/EXCLK	
P123	2-2-1			XT1	
P124				XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/TS10	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.9 44-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-31-3	I/O	Input port	TS26/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-31-1			TS27/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5			I/O	
P21		ANI1/AVREFM			
P22	4-35-1	ANI2/TS20			
P23	4-37-1	ANI3/TS21			
P24	4-33-1	ANI4/TS22			
P25		ANI5/TS23			
P26		ANI6/TS24			
P27		ANI7/TS25			
P30	7-31-2	I/O	Input port		INTP3/TSCAP/RTC1HZ/SCK11/SCL11
P31				TI03/TO03/INTP4/TS01/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			TI07/TO07	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			TS28/INTP2/SO11	
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P62	12-38-1			—	
P63				—	
P70	7-31-2	I/O	Input port	KR0/TS02/SCK21/SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			KR1/TS03/SI21/SDA21/RxDA0	
P72	7-31-3			KR2/TS04/SO21/TxDA0	
P73	7-31-2			KR3/TS05	
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (V _{DD} tolerant) can be set for P120 as an output. P120 can be set for the analog pin function ^{Note} .
P121	7-2-1		Input port	VBAT/X1	
P122				X2/EXCLK	
P123	2-2-1		Input	XT1	
P124				XT2/EXCLKS	
P137	2-1-3	Input	Input port	INTP0	Port 13. 1-bit input-only port.
P146	7-1-3	I/O	Input port	TS09	Port 14. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
P147	7-9-5		Analog function	ANI18/TS10	
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V _{DD} , either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.10 48-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-31-3	I/O	Input port	TS26/TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P00 as an output.
P01	8-31-1			TS27/TO00/RxD1	
P10	8-1-11	I/O	Input port	SCK00/SCL00/TS11/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (V _{DD} tolerant) can be set for P10 to P15 and P17 as outputs. P13 can be set for the analog pin function ^{Note} .
P11				SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)	
P12	7-1-12			SO00/TxD0/TOOLTxD/TS13/(TI05)/(TO05)	
P13	8-6-9			TxD2/SO20/(SDAA0)/(TI04)/(TO04)/TS14	
P14	8-1-11			RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)/TS15	
P15				PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)/TS16	
P16	8-38-1			TI01/TO01/INTP5/TS17/(RxD0)	
P17	8-38-2			TI02/TO02/TS18/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions ^{Note} .
P21				ANI1/AVREFM	
P22	4-35-1			ANI2/TS20	
P23	4-37-1			ANI3/TS21	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	
P30	7-31-2	I/O	Input port	INTP3/TSCAP/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TI03/TO03/INTP4/TS01/(PCLBUZ0)	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			TI07/TO07	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-31-3	I/O	Input port	TS00/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (V _{DD} tolerant) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	7-38-1			P51/TS28/INTP2/SO11	
P60	12-38-3	I/O	Input port	SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (6-V tolerant).
P61				SDAA0	
P62	12-38-1			—	
P63				—	
P70	7-31-2	I/O	Input port	KR0/TS02/SCK21/SCL21	Port 7. 6-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (V _{DD} tolerant) can be set for P71, P72, and P74 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			KR1/TS03/SI21/SDA21/RxDA0	
P72	7-31-3			KR2/TS04/SO21/TxDA0	
P73	7-31-2			KR3/TS05/SO01	
P74	7-31-3			KR4/TS06/INTP8/SI01/SDA01	
P75	7-31-2			KR5/TS07/INTP9/SCK01/SCL01	
P120	7-9-6	I/O	Analog function	ANI19	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (V _{DD} tolerant) can be set for P120 as an output. P120 can be set for the analog pin function ^{Note} .
P121	7-2-1		Input port	VBAT/X1	
P122				X2/EXCLK/EXCLKS	
P123	2-2-1		Input	XT1	
P124				XT2/EXCLKS	
P130	1-1-5	I/O	Output port	TS19	Port 13. 1-bit output port and 1-bit input-only port. Note that when the P130 pin is to be used for the TS19 multiplexed function, the pin functions as an I/O pin.
P137	2-1-3	Input	Input port	INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6/TS08	Port 14. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function ^{Note} .
P146				TS09	
P147	7-9-5		Analog function	ANI18/TS10	
RESET	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V _{DD} , either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register x (PMCAx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)**.

2.2 Pin Functions Other than Port Pin Functions

2.2.1 Functions for each product

(1/4)

Function Name	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
ANI0	√	√	√	√	√	√	√	√	√	√
ANI1	√	√	√	√	√	√	√	√	√	√
ANI2	√	√	√	√	√	√	√	√	√	√
ANI3	√	√	√	√	√	√	—	—	—	—
ANI4	√	√	√	√	—	—	—	—	—	—
ANI5	√	√	√	√	—	—	—	—	—	—
ANI6	√	√	√	—	—	—	—	—	—	—
ANI7	√	√	—	—	—	—	—	—	—	—
ANI16	—	—	—	—	√	√	√	√	√	—
ANI17	—	—	—	—	√	√	√	√	√	—
ANI18	√	√	√	√	√	√	√	√	√	—
ANI19	√	√	√	√	√	√	—	—	—	—
TS00	√	√	√	√	√	√	√	√	—	—
TS01	√	√	√	√	√	√	√	√	—	—
TS02	√	√	√	√	√	—	—	—	—	—
TS03	√	√	√	√	—	—	—	—	—	—
TS04	√	√	√	√	—	—	—	—	—	—
TS05	√	√	√	—	—	—	—	—	—	—
TS06	√	—	—	—	—	—	—	—	—	—
TS07	√	—	—	—	—	—	—	—	—	—
TS08	√	—	—	—	—	—	—	—	—	—
TS09	√	√	—	—	—	—	—	—	—	—
TS10	√	√	√	√	√	√	√	√	√	—
TS11	√	√	√	√	√	√	√	√	√	√
TS12	√	√	√	√	√	√	√	√	√	√
TS13	√	√	√	√	√	√	√	√	√	√
TS14	√	√	√	√	√	√	—	—	—	—
TS15	√	√	√	√	√	√	—	—	—	—
TS16	√	√	√	√	√	√	—	—	—	—
TS17	√	√	√	√	√	√	√	√	√	—
TS18	√	√	√	√	√	√	√	√	√	√
TS19	√	—	—	—	—	—	√	—	—	—
TS20	√	√	√	√	√	√	√	√	√	√
TS21	√	√	√	√	√	√	—	—	—	—
TS22	√	√	√	√	—	—	—	—	—	—
TS23	√	√	√	√	—	—	—	—	—	—

(2/4)

Function Name	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
TS24	√	√	√	—	—	—	—	—	—	—
TS25	√	√	—	—	—	—	—	—	—	—
TS26	√	√	√	√	√	√	√	√	√	—
TS27	√	√	√	√	√	√	√	√	√	—
TS28	√	√	√	√	√	√	—	—	—	—
TSCAP	√	√	√	√	√	√	√	√	√	√
INTP0	√	√	√	√	√	√	√	√	√	√
INTP1	√	√	√	√	√	√	√	√	—	—
INTP2	√	√	√	√	√	√	—	—	—	—
INTP3	√	√	√	√	√	√	√	√	√	√
INTP4	√	√	√	√	√	√	√	√	—	—
INTP5	√	√	√	√	√	√	√	√	√	—
INTP6	√	—	—	—	—	—	—	—	—	—
INTP8	√	—	—	—	—	—	—	—	—	—
INTP9	√	—	—	—	—	—	—	—	—	—
KR0	√	√	√	—	—	—	—	—	—	—
KR1	√	√	√	—	—	—	—	—	—	—
KR2	√	√	√	—	—	—	—	—	—	—
KR3	√	√	√	—	—	—	—	—	—	—
KR4	√	—	—	—	—	—	—	—	—	—
KR5	√	—	—	—	—	—	—	—	—	—
PCLBUZ0	√	√	√	√	√	√	√	√	—	—
PCLBUZ1	√	√	√	√	√	√	—	—	—	—
REGC	√	√	√	√	√	√	√	√	√	√
RTC1HZ	√	√	√	√	√	√	√	√	√	√
RESET	√	√	√	√	√	√	√	√	√	√
RxD0	√	√	√	√	√	√	√	√	√	√
RxD1	√	√	√	√	√	√	√	√	√	—
RxD2	√	√	√	√	√	√	—	—	—	—
TxD0	√	√	√	√	√	√	√	√	√	√
TxD1	√	√	√	√	√	√	√	√	√	—
TxD2	√	√	√	√	√	√	—	—	—	—
RxDA0	√	√	√	√	—	—	—	—	—	—
TxDA0	√	√	√	√	—	—	—	—	—	—
SCK00	√	√	√	√	√	√	√	√	√	√
SCK01	√	—	—	—	—	—	—	—	—	—
SCK11	√	√	√	√	√	√	√	√	√	√
SCK20	√	√	√	√	√	√	—	—	—	—
SCK21	√	√	√	√	—	—	—	—	—	—
SCL00	√	√	√	√	√	√	√	√	√	√

(3/4)

Function Name	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
SCL01	√	—	—	—	—	—	—	—	—	—
SCL11	√	√	√	√	√	√	√	√	√	√
SCL20	√	√	√	√	√	√	—	—	—	—
SCL21	√	√	√	√	—	—	—	—	—	—
SDA00	√	√	√	√	√	√	√	√	√	√
SDA01	√	—	—	—	—	—	—	—	—	—
SDA11	√	√	√	√	√	√	√	√	√	√
SDA20	√	√	√	√	√	√	—	—	—	—
SDA21	√	√	√	√	—	—	—	—	—	—
SI00	√	√	√	√	√	√	√	√	√	√
SI01	√	—	—	—	—	—	—	—	—	—
SI11	√	√	√	√	√	√	√	√	√	√
SI20	√	√	√	√	√	√	—	—	—	—
SI21	√	√	√	√	—	—	—	—	—	—
SO00	√	√	√	√	√	√	√	√	√	√
SO01	√	—	—	—	—	—	—	—	—	—
SO11	√	√	√	√	√	√	√	√	√	—
SO20	√	√	√	√	√	√	—	—	—	—
SO21	√	√	√	√	—	—	—	—	—	—
SCLA0	√	√	√	√	√	√	√	√	—	—
SDAA0	√	√	√	√	√	√	√	√	—	—
TI00	√	√	√	√	√	√	√	√	√	—
TI01	√	√	√	√	√	√	√	√	√	—
TI02	√	√	√	√	√	√	√	√	√	√
TI03	√	√	√	√	√	√	√	√	—	—
TI04	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TI05	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TI06	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TI07	√	√	(√)	(√)	(√)	(√)	—	—	—	—
TO00	√	√	√	√	√	√	√	√	√	—
TO01	√	√	√	√	√	√	√	√	√	—
TO02	√	√	√	√	√	√	√	√	√	√
TO03	√	√	√	√	√	√	√	√	—	—
TO04	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TO05	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TO06	(√)	(√)	(√)	(√)	(√)	(√)	—	—	—	—
TO07	√	√	(√)	(√)	(√)	(√)	—	—	—	—
X1	√	√	√	√	√	√	√	√	√	√
X2	√	√	√	√	√	√	√	√	√	√
EXCLK	√	√	√	√	√	√	√	√	√	√

(4/4)

Function Name	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
XT1	√	√	√	√	√	√	√	√	√	√
XT2	√	√	√	√	√	√	√	√	√	√
EXCLKS	√	√	√	√	√	√	√	√	√	√
VDD	√	√	√	√	√	√	√	√	√	√
VBAT	√	√	√	—	—	—	—	—	—	—
AVREFP	√	√	√	√	√	√	√	√	√	√
AVREFM	√	√	√	√	√	√	√	√	√	√
VSS	√	√	√	√	√	√	√	√	√	√
TOOLRxD	√	√	√	√	√	√	√	√	√	√
TOOLTxD	√	√	√	√	√	√	√	√	√	√
TOOL0	√	√	√	√	√	√	√	√	√	√

Remark Pin functions with a check mark enclosed in parentheses in the above table are only available when the bit corresponding to the given function in the peripheral I/O redirection register (PIOR) is set to 1.

2.2.2 Description of pin functions

(1/2)

Function Name	I/O	Function
ANI0 to ANI7, ANI16 to ANI19	Input	Analog voltage inputs for the A/D converter (see Figure 12 - 45 Analog Input Pin Connection)
TS00 to TS28	I/O	Electrostatic capacitance measurement pins (touch sensor)
TSCAP	I/O	Pin for connecting a power supply stabilization capacitor for the touch sensor interface. Connect this pin to Vss via a capacitor (10 nF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
INTP0 to INTP6, INTP8, INTP9	Input	External interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
KR0 to KR5	Input	Key interrupt inputs
PCLBUZ0, PCLBUZ1	Output	Clock outputs/buzzer outputs
REGC	—	Pin for connecting a regulator output stabilization capacitor for internal operation. Connect this pin to Vss via a capacitor (0.47 to 1 μ F). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
RTC1HZ	Output	Realtime clock correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When an external reset signal is not used, connect this pin directly or via a resistor to VDD.
RxD0 to RxD2	Input	Serial data input pins for serial interfaces UART0, UART1, and UART2
TxD0 to TxD2	Output	Serial data output pins for serial interfaces UART0, UART1, and UART2
RxDA0	Input	Serial data input pin for the UARTA0 serial interface
TxDA0	Output	Serial data output pin for the UARTA0 serial interface
SCK00, SCK01, SCK11, SCK20, SCK21	I/O	Serial clock I/O pins for serial interfaces CSI00, CSI01, CSI11, CSI20, and CSI21
SCL00, SCL01, SCL11, SCL20, SCL21	Output	Serial clock output pins for serial interfaces IIC00, IIC01, IIC11, IIC20, and IIC21
SDA00, SDA01, SDA11, SDA20, SDA21	I/O	Serial data I/O pins for serial interfaces IIC00, IIC01, IIC11, IIC20, and IIC21
SI00, SI01, SI11, SI20, SI21	Input	Serial data input pins for serial interfaces CSI00, CSI01, CSI11, CSI20, and CSI21
SO00, SO01, SO11, SO20, SO21	Output	Serial data output pins for serial interfaces CSI00, CSI01, CSI11, CSI20, and CSI21
SCLA0	I/O	Clock I/O pin for the IICA0 serial interface
SDAA0	I/O	Serial data I/O pin for the IICA0 serial interface
TI00 to TI07	Input	Pins for inputting an external counting clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	Output	Timer output pins for 16-bit timers 00 to 07
X1, X2	—	Resonator connection for the main system clock
EXCLK	Input	External clock input for the main system clock
XT1, XT2	—	Resonator connection for the subsystem clock
EXCLKS	Input	External clock input for the subsystem clock
VDD	—	Positive power supply
VBAT	—	Power supply pin for battery backup
AVREFP	Input	Positive reference voltage input of the A/D converter
AVREFM	Input	Negative reference voltage input of the A/D converter
VSS	—	Ground voltage
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming

(2/2)

Function Name	I/O	Function
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming
TOOL0	I/O	Data I/O for flash memory programmer or debugger

Caution The relationship between the voltage on P40/TOOL0 and the operating mode after release from the reset state is as follows.

Table 2 - 2 Relationship between the Voltage on P40/TOOL0 and Operating Mode after Release from the Reset State

P40/TOOL0	Operating Mode
V _{DD}	Normal operating mode
0 V	Flash memory programming mode

For details, see **30.4 Programming Method**.

Remark As a measure against noise and latch up, connect a bypass capacitor (about 0.1 μ F) with relatively thick wire at the shortest distance from the pins in a line from V_{DD} to V_{SS}.

2.3 Connection of Unused Pins

Table 2 - 3 shows the connections of unused pins.

Remark The pins mounted depend on the product. Refer to **1.3 Pin Configuration (Top View)** and **2.1 Functions of Port Pins**.

Table 2 - 3 Connections of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00, P01	I/O	Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P10 to P17		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P20 to P27		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P30, P31		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P40/TOOL0		Input: Independently connect the pin to VDD via a resistor, or leave the pin open-circuit. Output: Leave the pin open-circuit.
P41		Input: Independently connect the pin to VDD or VSS via a resistor. Output: Leave the pin open-circuit.
P50, P51		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P60 to P63		Input: Independently connect the pins to VDD or VSS via resistors. Output: Set the port's output latch to 0 and leave the pins open-circuit, or set the port's output latch to 1 and independently connect the pins to VDD or VSS via resistors.
P70 to P75		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P120		Input: Independently connect the pin to VDD or VSS via a resistor. Output: Leave the pin open-circuit.
P121, P122		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P123, P124	Input	Set the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC), set the XTSTOP bit in the clock operation status control register (CSC) to 1, and leave the pins open-circuit. Note Alternatively, provide the pins with independent connections to VDD or VSS via resistors.
P130	Output	Leave the pin open-circuit.
P137	Input	Set the PDIDIS137 bit in port digital input disable register (PDIDIS) to 1, and leave the pin open-circuit. Alternatively, provide the pin with an independent connection to VDD or VSS via a resistor.
P140, P146, P147	I/O	Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
RESET	Input	Connect the pin directly or via a resistor to VDD.
REGC	—	Connect the pin to VSS via a capacitor (0.47 to 1 µF).

Note When the low-speed on-chip oscillator clock (f_{IL}) is selected for the CPU/peripheral hardware clock frequency (f_{CLK}), the current may increase approximately by 1 µA.

2.4 Block Diagrams of Pins

Figures 2 - 1 to 2 - 27 show the block diagrams of the pins described in 2.1.1 16-pin products to 2.1.10 48-pin products.

Figure 2 - 1 Pin Block Diagram for Pin Type 1-1-5

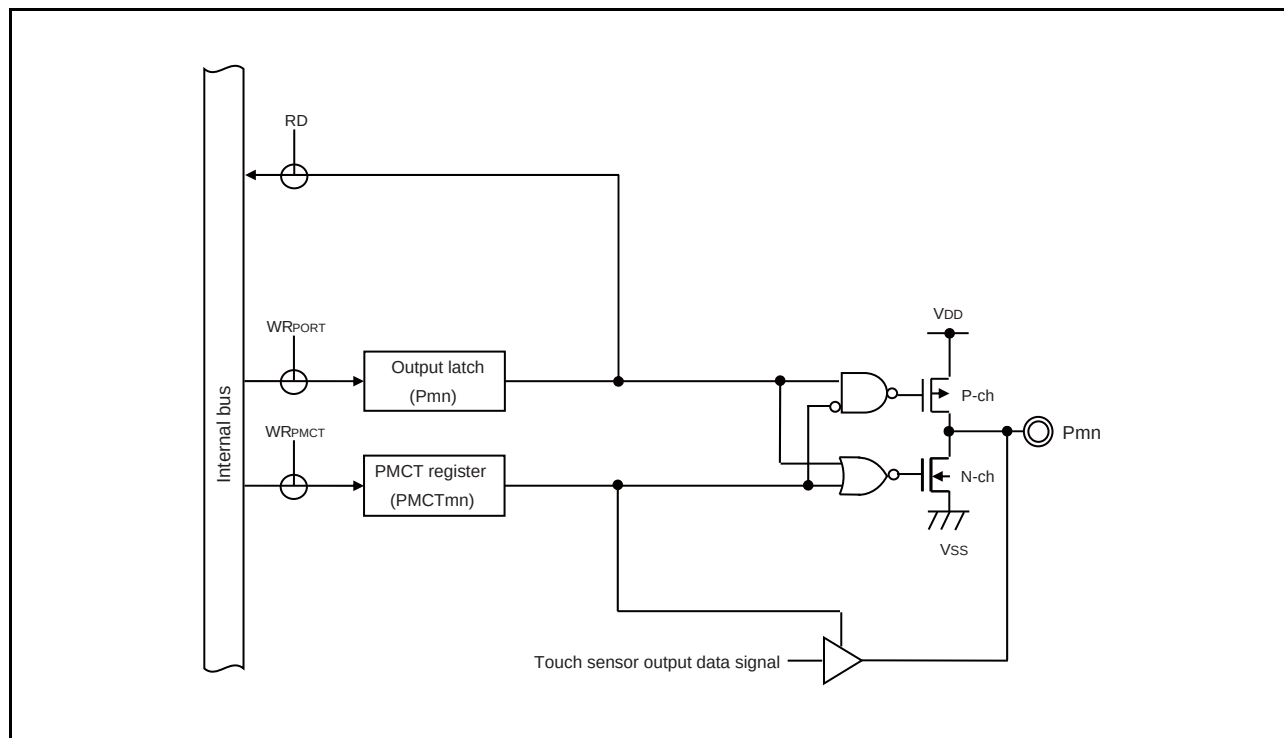


Figure 2 - 2 Pin Block Diagram for Pin Type 2-1-1

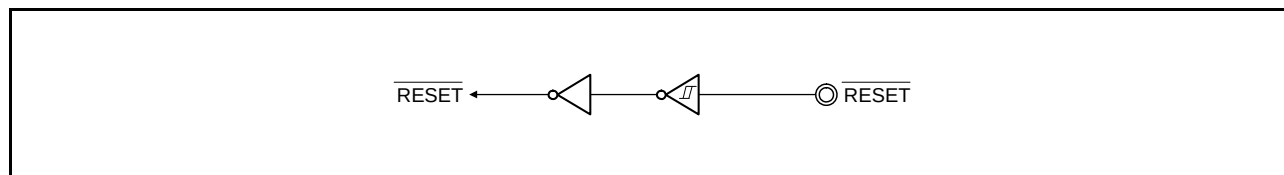
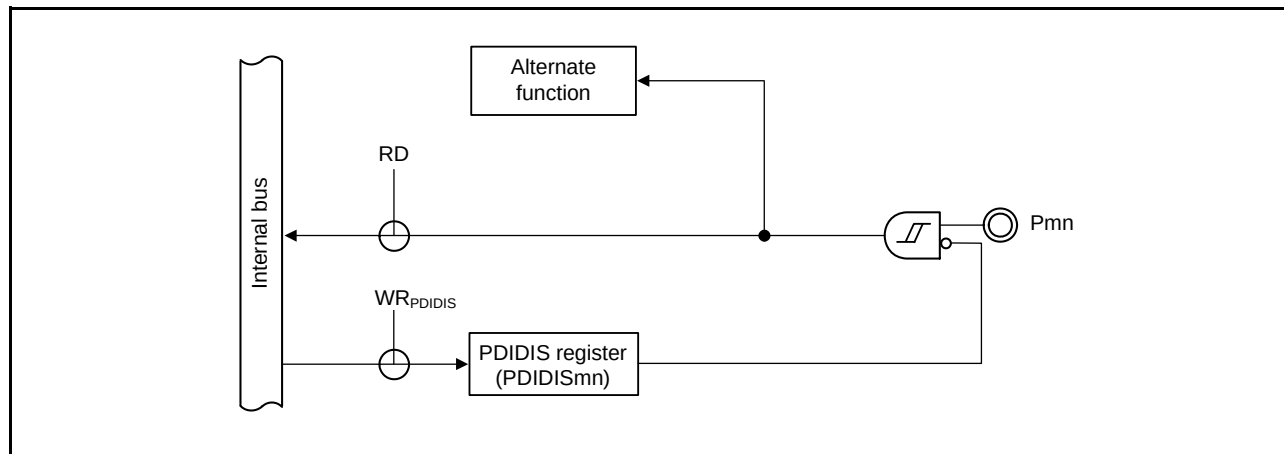
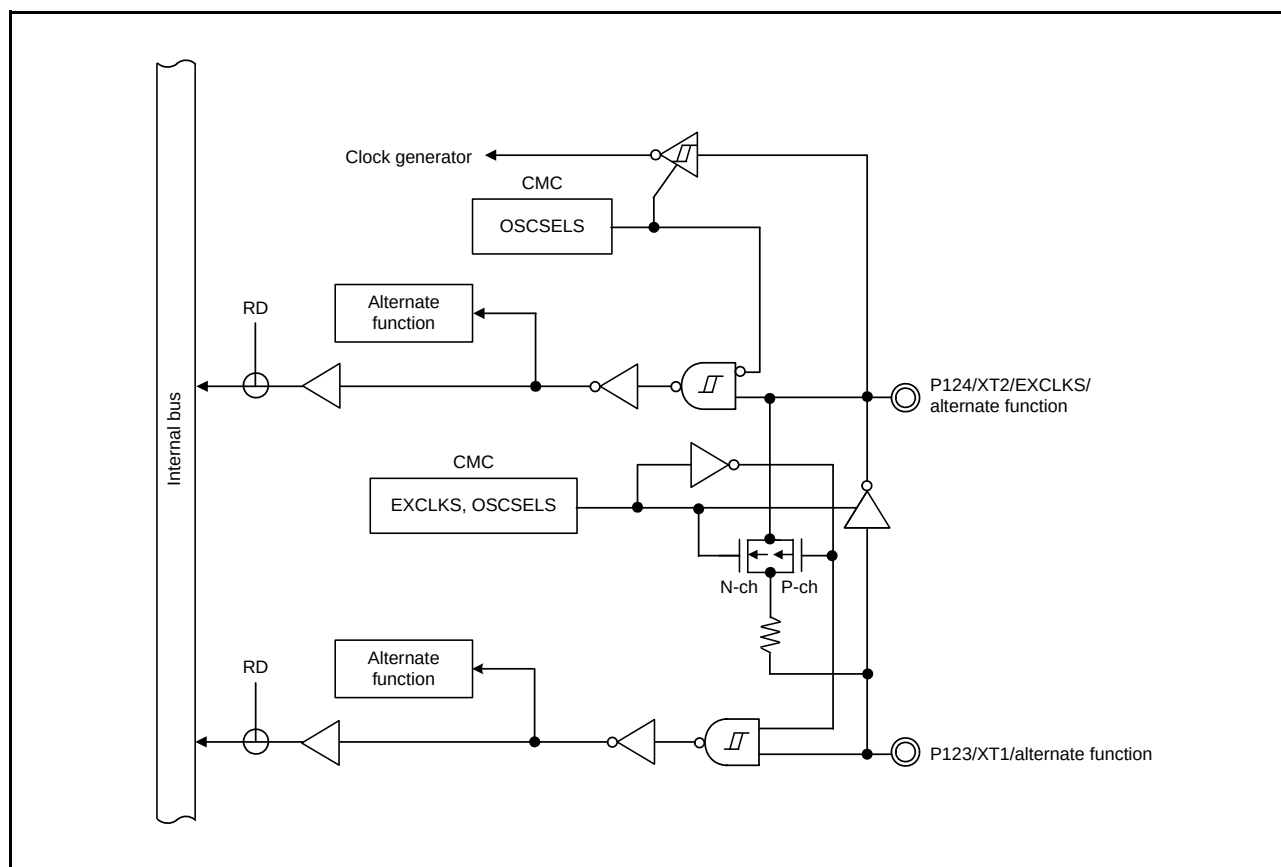


Figure 2 - 3 Pin Block Diagram for Pin Type 2-1-3



Remark For alternate functions, see 2.1 Functions of Port Pins.

Figure 2 - 4 Pin Block Diagram for Pin Type 2-2-1



Remark For alternate functions, see 2.1 Functions of Port Pins.

Figure 2 - 5 Pin Block Diagram for Pin Type 4-3-5

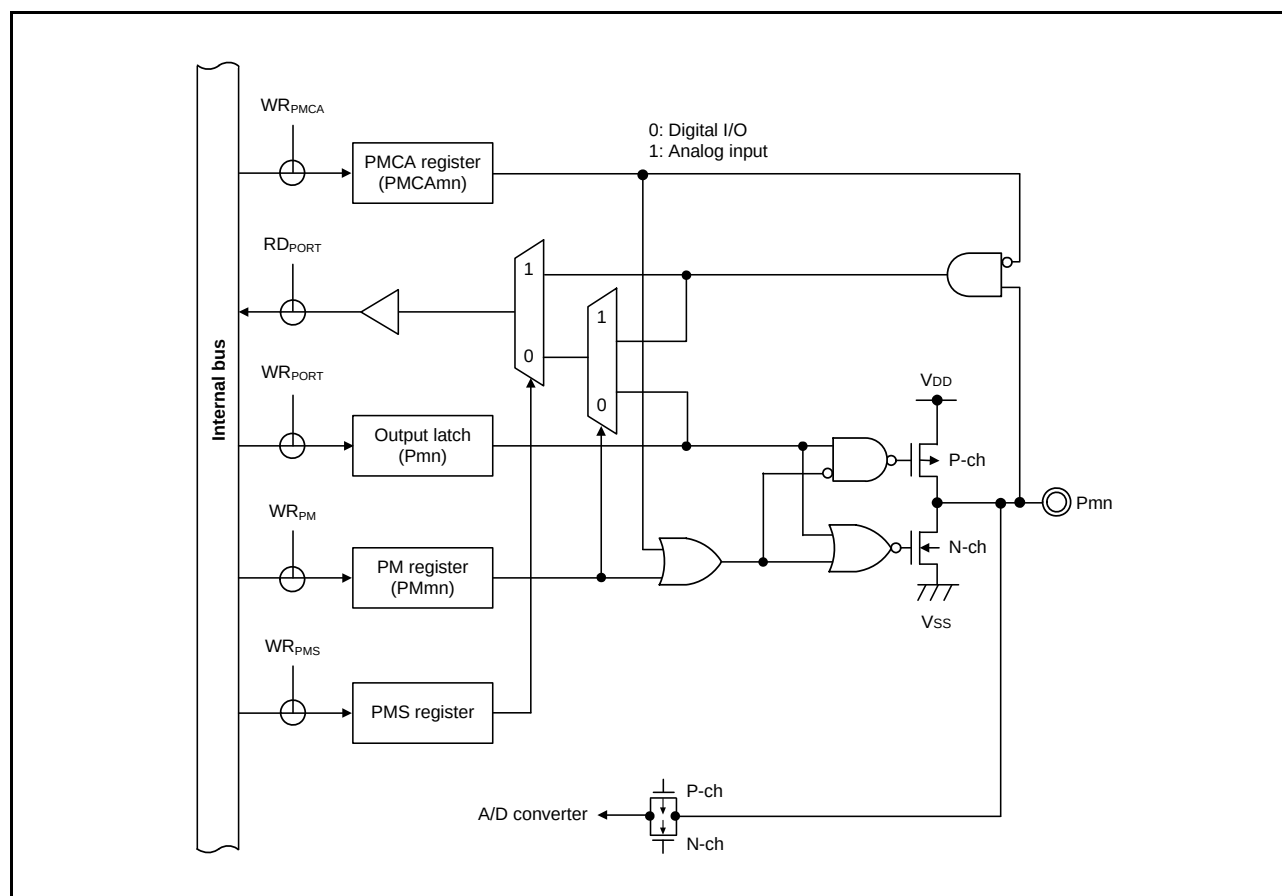


Figure 2 - 6 Pin Block Diagram for Pin Type 4-33-1

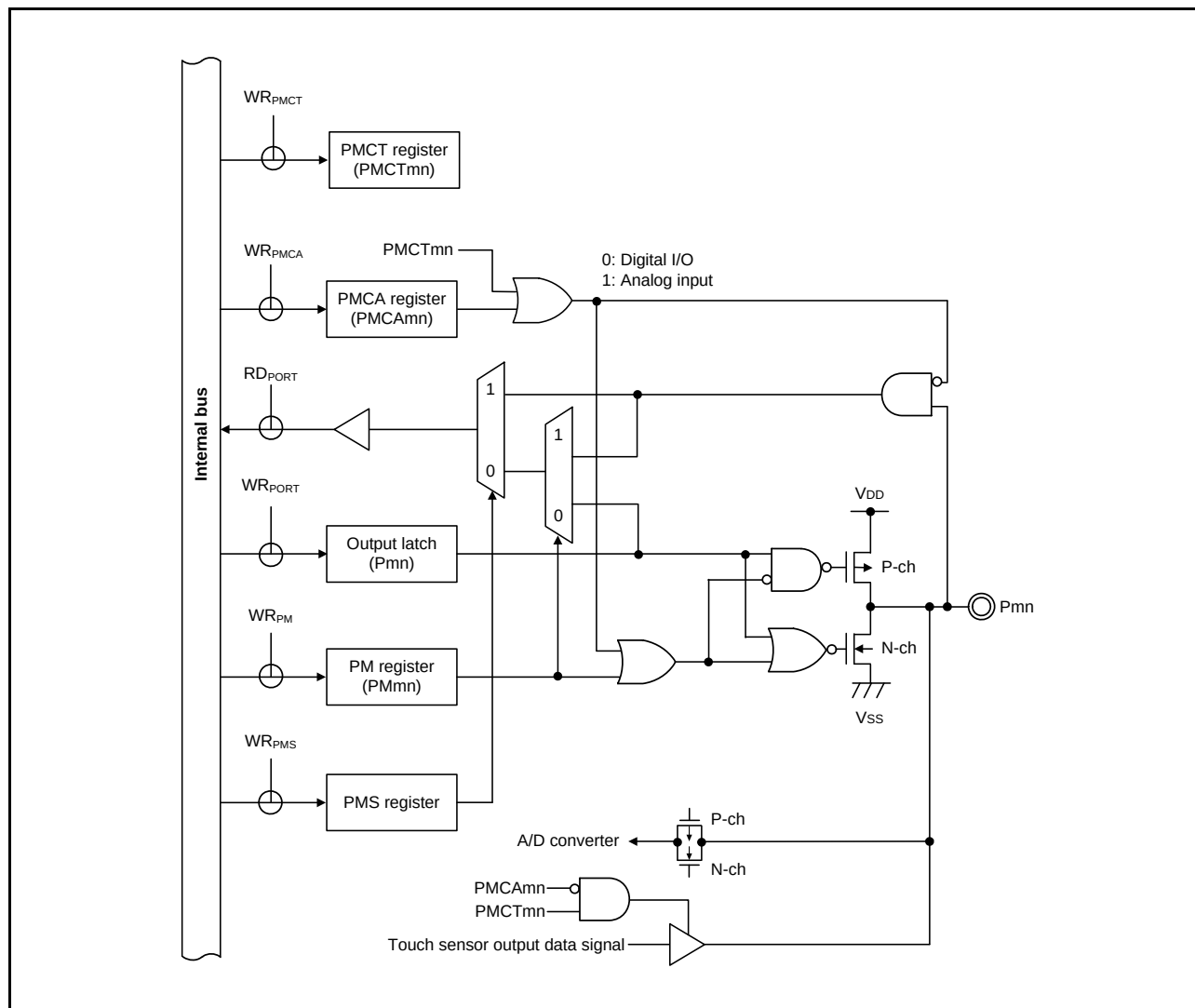


Figure 2 - 7 Pin Block Diagram for Pin Type 4-35-1

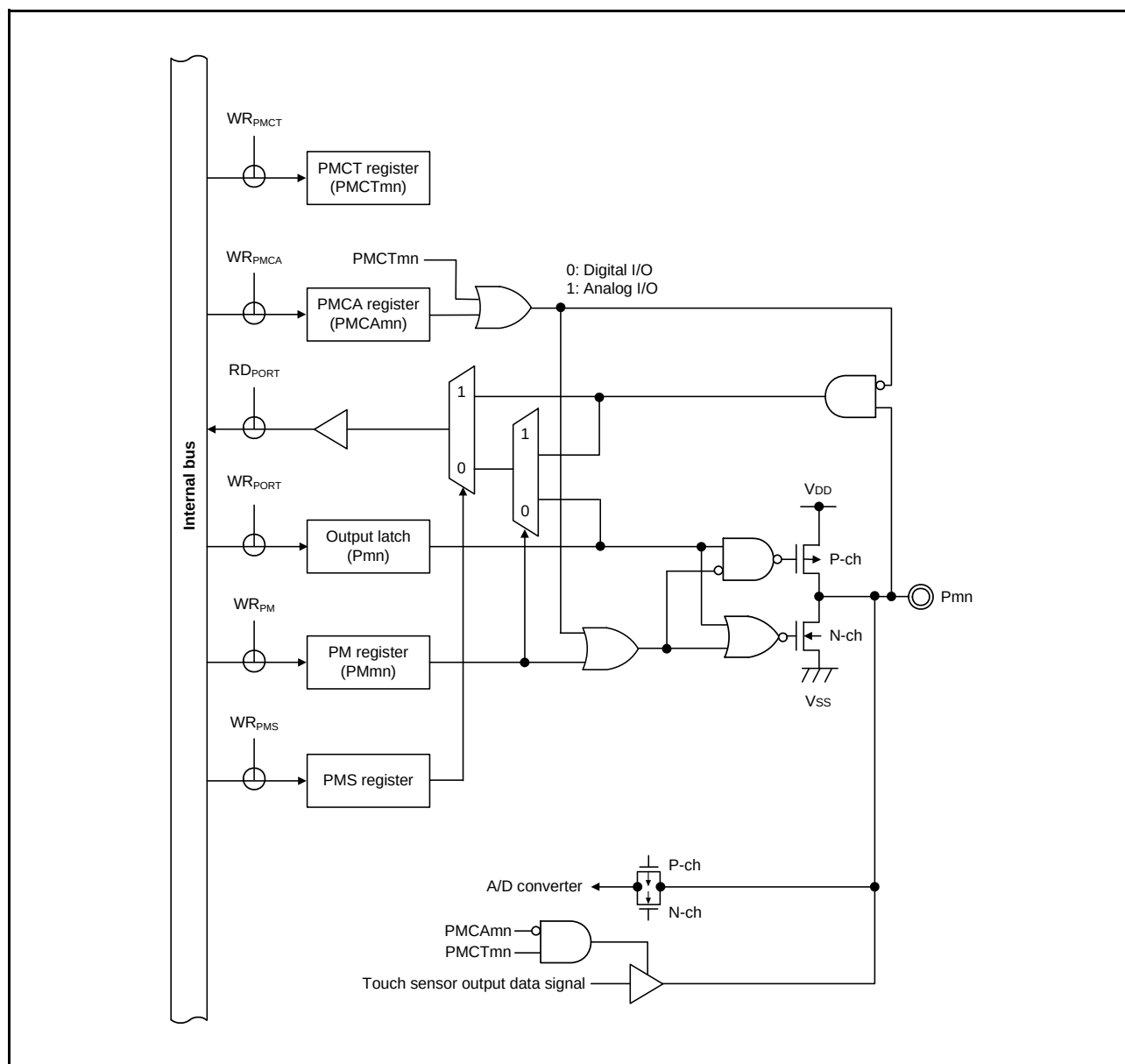
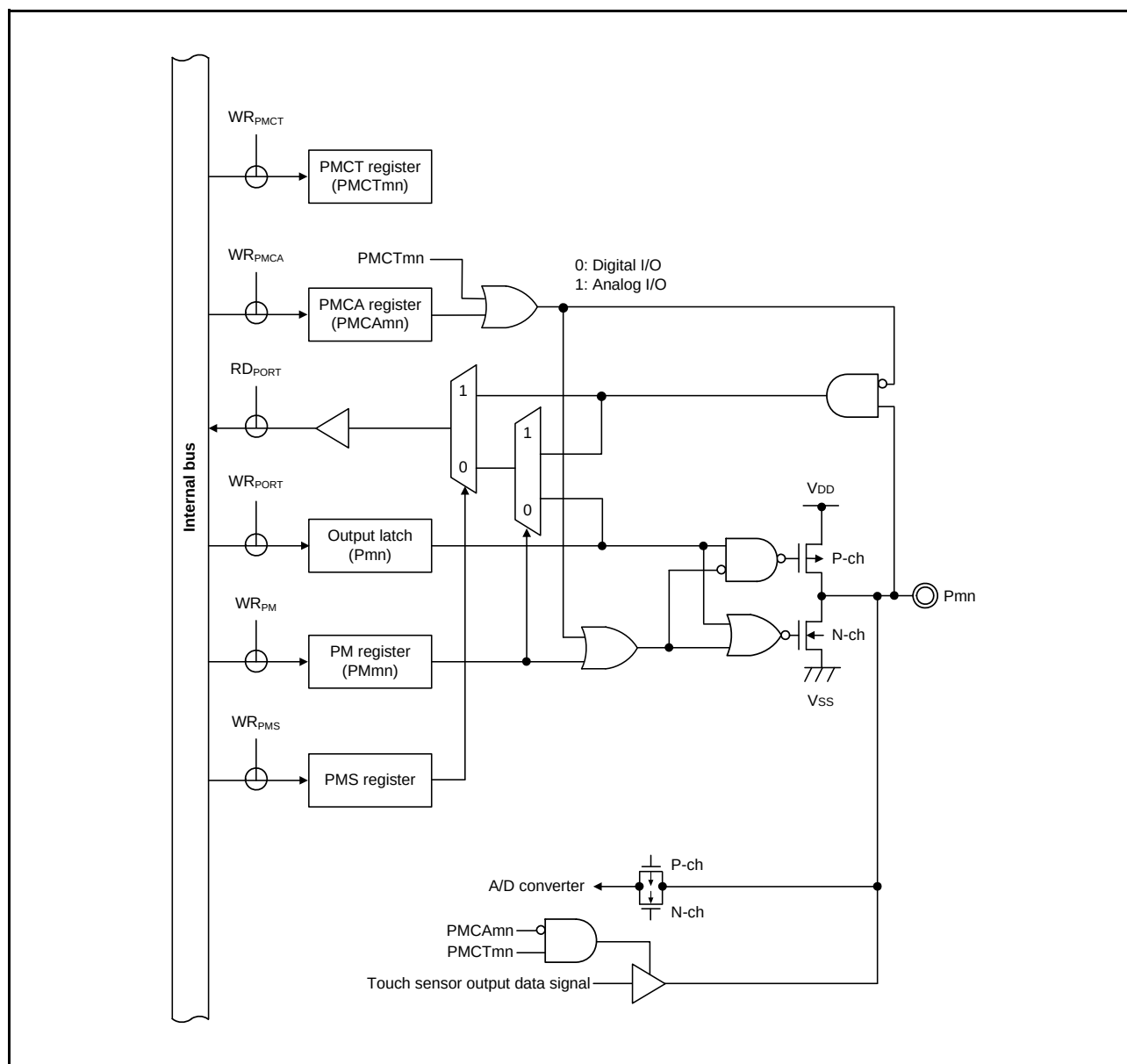


Figure 2 - 8 Pin Block Diagram for Pin Type 4-37-1



The schematic diagram illustrates the internal architecture of the PM module. An **Internal bus** on the left provides control signals: **WRPU**, **RDPORT**, **WRPORT**, **WRPMS**, and **WRPM**. These signals are connected to various registers and control blocks:

- WRPU** connects to the **PU register (PUMn)**.
- RDPORT** connects to a buffer that drives the **RDPORT** output pin.
- WRPORT** connects to the **Output latch (Pmn)**.
- WRPMS** connects to the **PMS register**.
- WRPM** connects to the **PM register (PMmn)**.

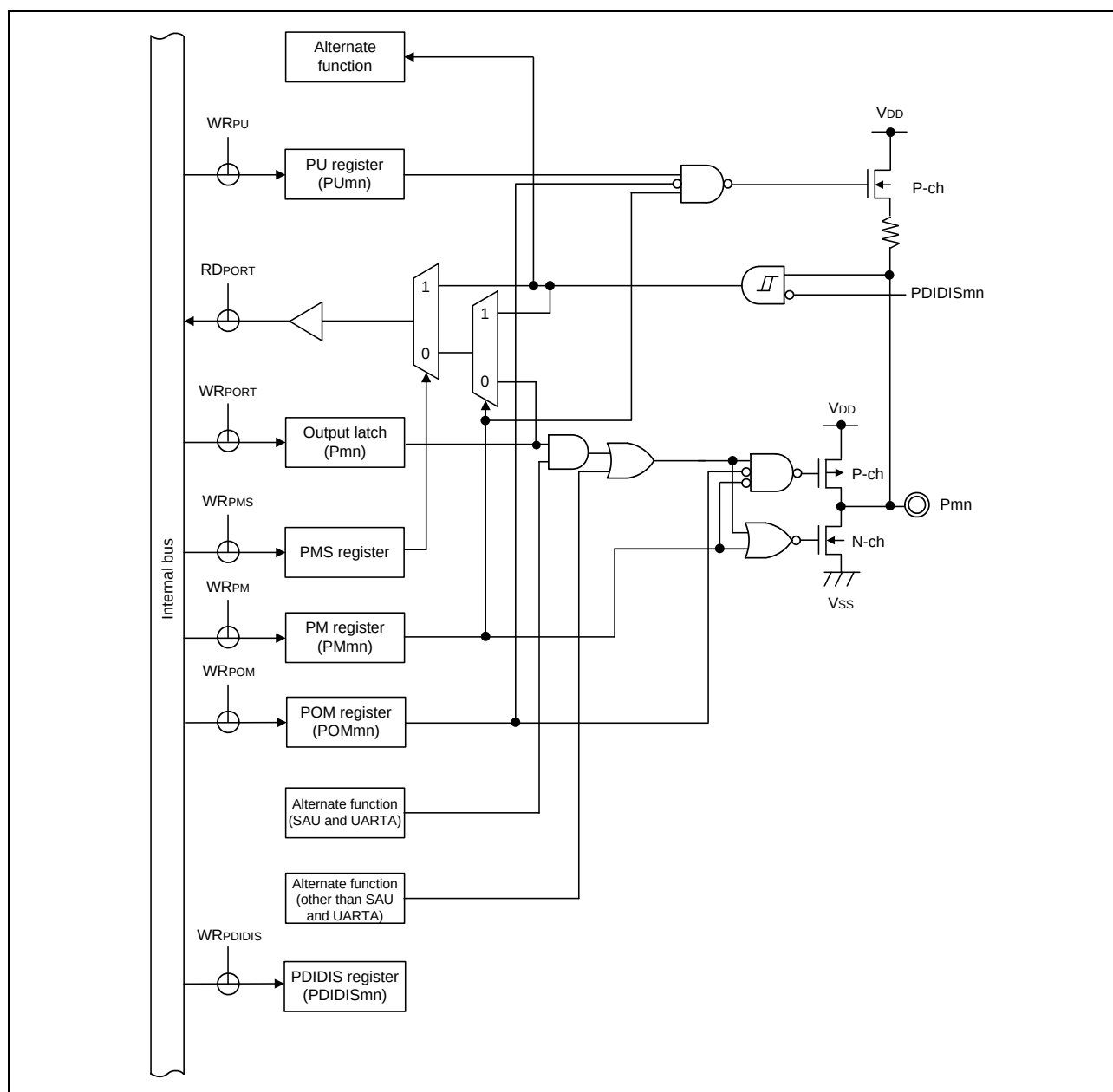
Control logic is implemented using several blocks and gates:

- Alternate function** and **Alternate function (SAU and UARTA)** blocks provide inputs to the **PU register (PUMn)** and the **Output latch (Pmn)**.
- Alternate function (other than SAU and UARTA)** provides an input to the **PM register (PMmn)**.
- The **PU register (PUMn)** output is ANDed with the **Alternate function (SAU and UARTA)** input to drive the **P-ch** MOSFET gate.
- The **Output latch (Pmn)** output is ANDed with the **Alternate function (other than SAU and UARTA)** input to drive the **P-ch** MOSFET gate.
- The **PMS register** output is ANDed with the **Alternate function (SAU and UARTA)** input to drive the **N-ch** MOSFET gate.
- The **PM register (PMmn)** output is ANDed with the **Alternate function (other than SAU and UARTA)** input to drive the **N-ch** MOSFET gate.

The output stage consists of a **P-ch** MOSFET (pull-up) and an **N-ch** MOSFET (pull-down). The **P-ch** MOSFET is connected to **VDD** and the **N-ch** MOSFET is connected to **VSS**. The output node between the two MOSFETs is labeled **Pmn**.

Remark 2. SAU: Serial array unit

Figure 2 - 10 Pin Block Diagram for Pin Type 7-1-12

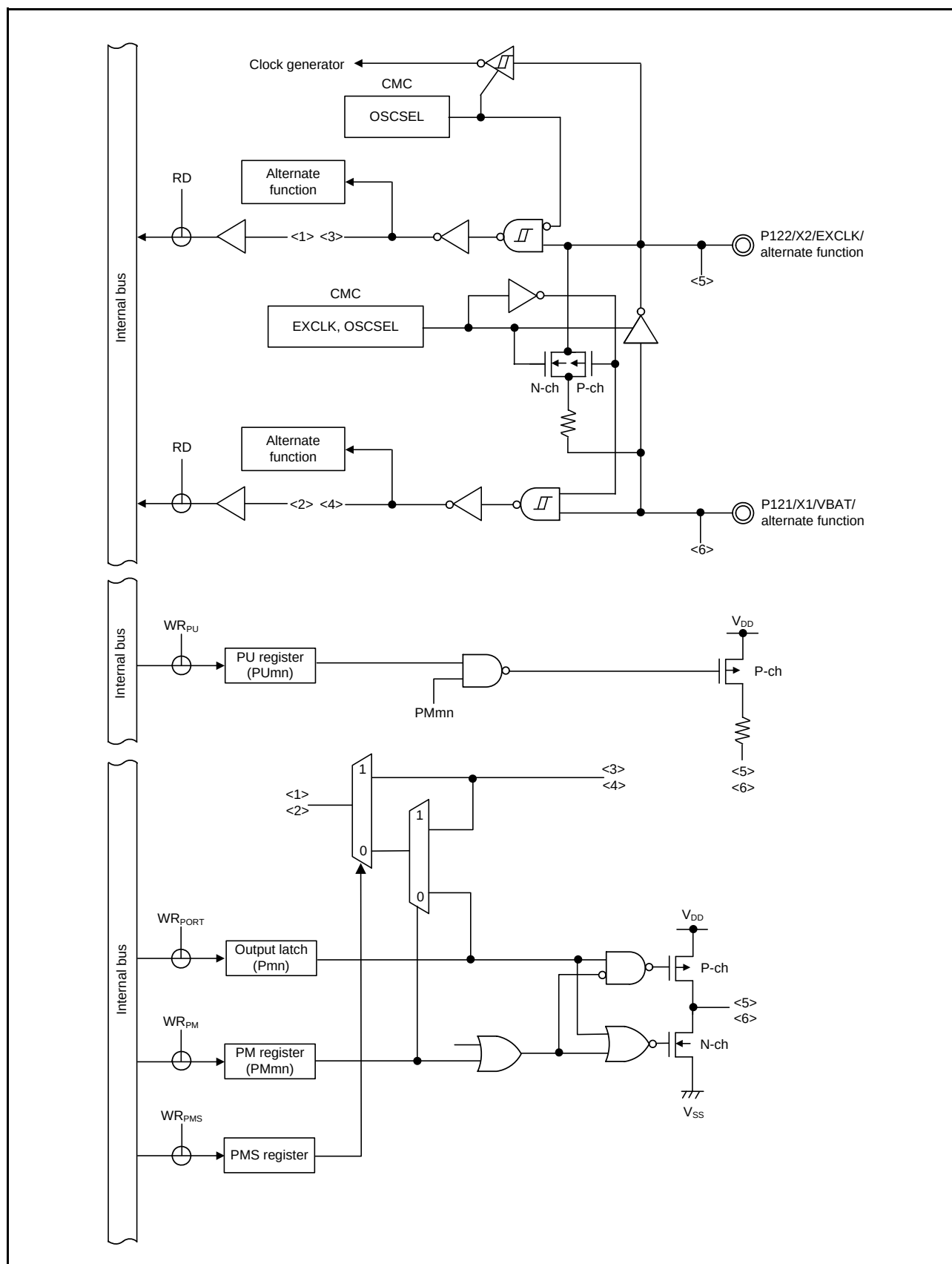


Caution The input buffer is enabled even if the type 7-1-12 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 7-1-12 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

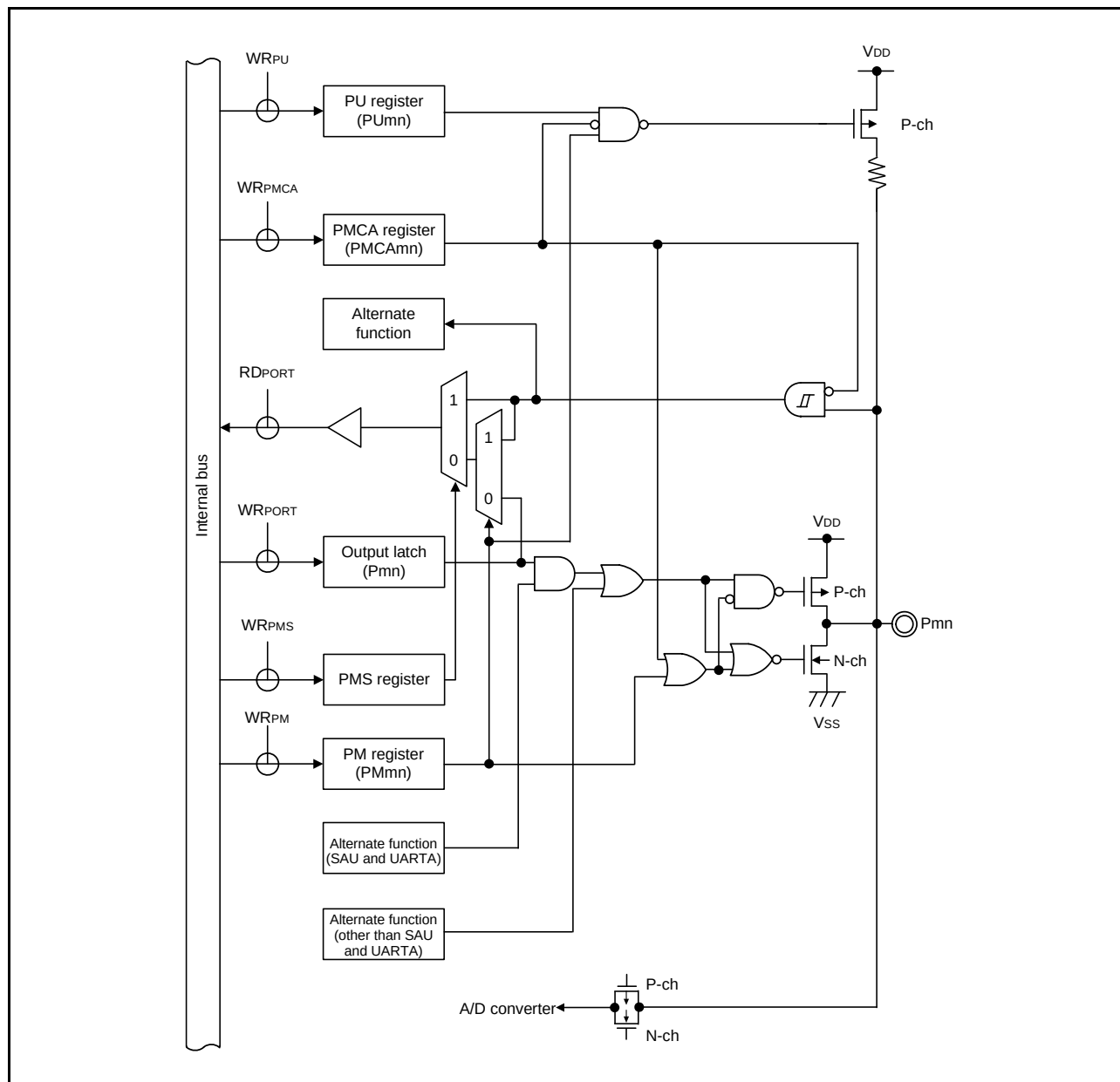
Remark 2. SAU: Serial array unit

Figure 2 - 11 Pin Block Diagram for Pin Type 7-2-1



Remark For alternate functions, see 2.1 Functions of Port Pins.

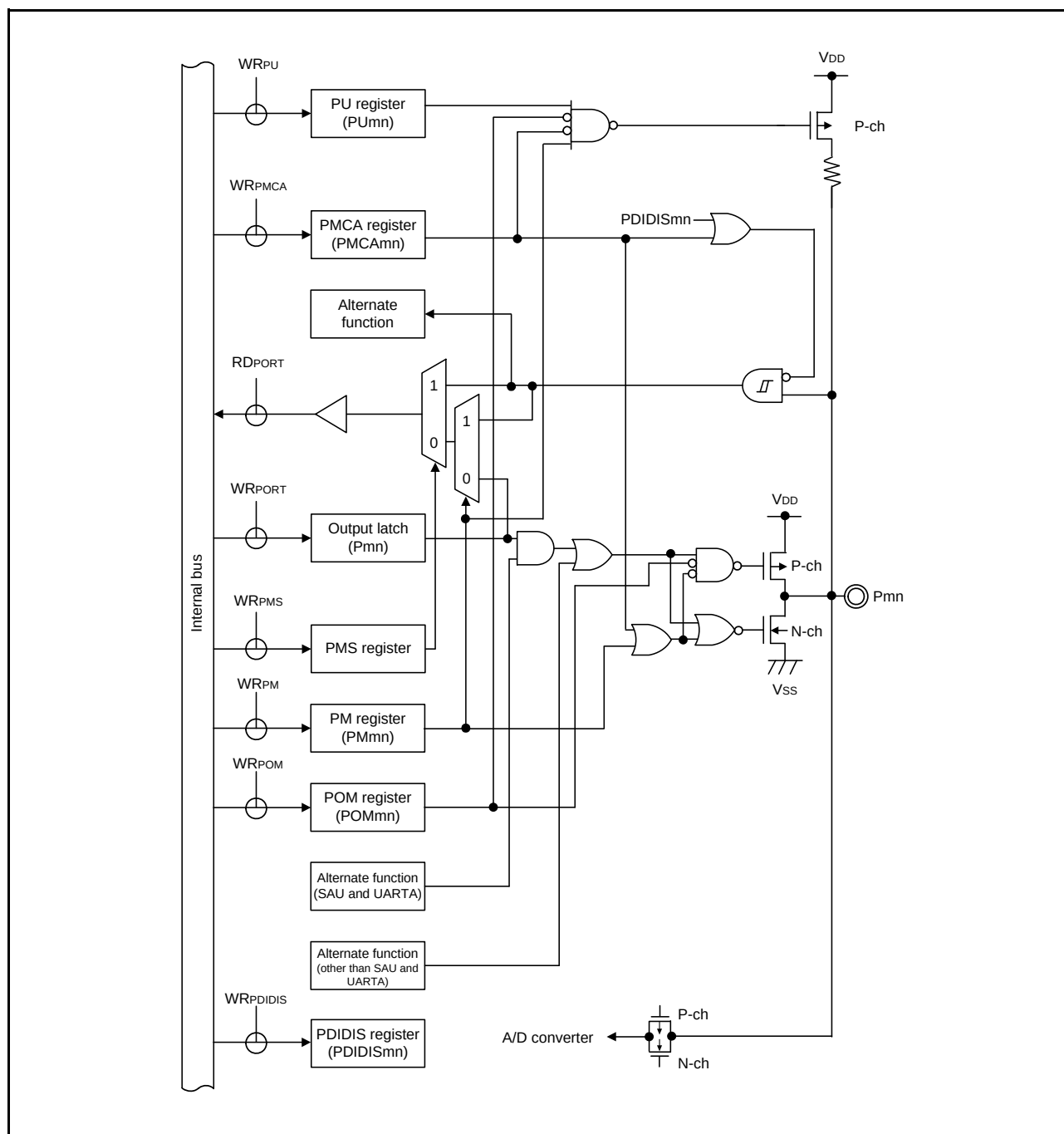
Figure 2 - 12 Pin Block Diagram for Pin Type 7-9-5



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 13 Pin Block Diagram for Pin Type 7-9-6

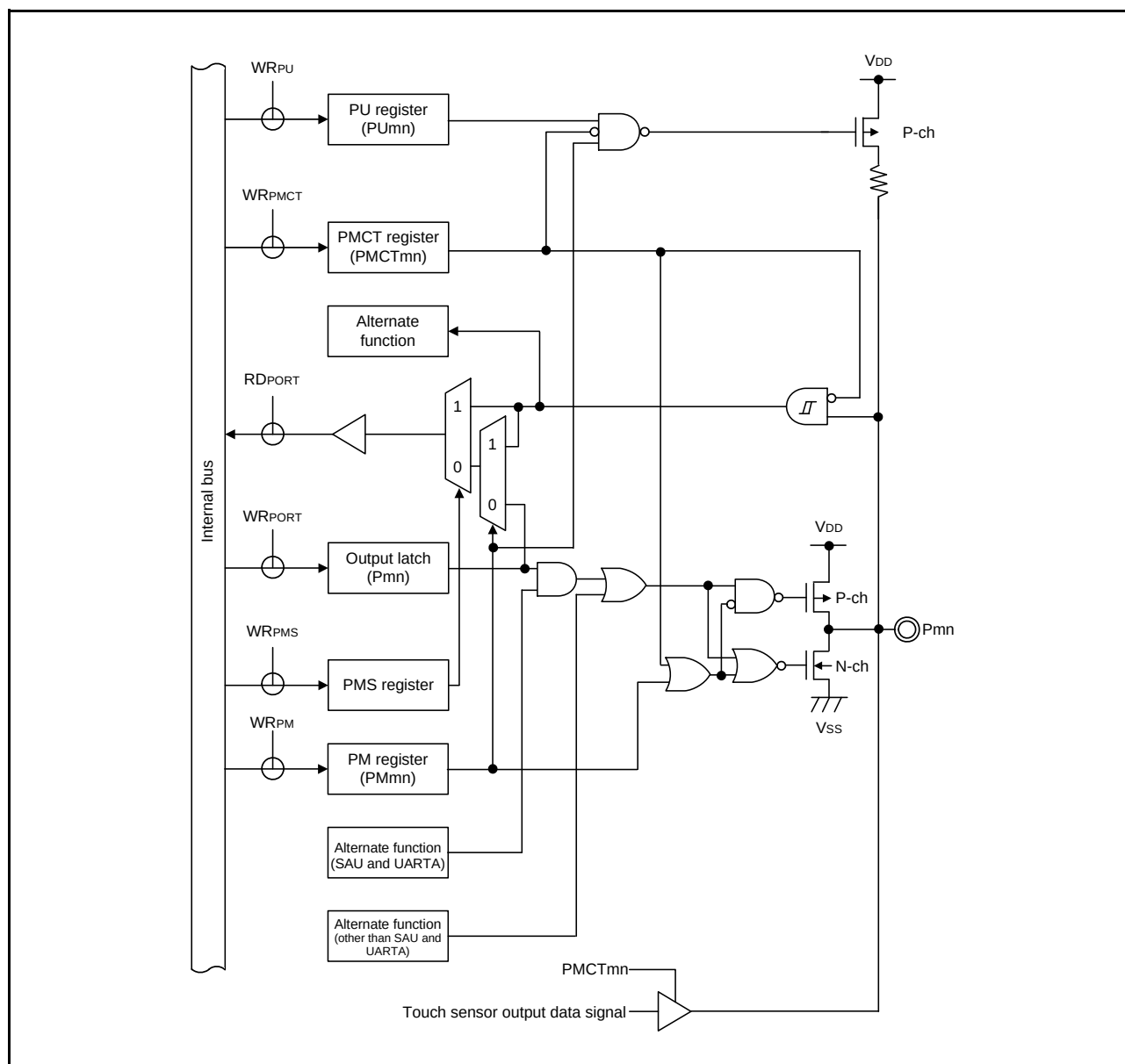


Caution The input buffer is enabled even if the type 7-9-6 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 7-9-6 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 14 Pin Block Diagram for Pin Type 7-31-2



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

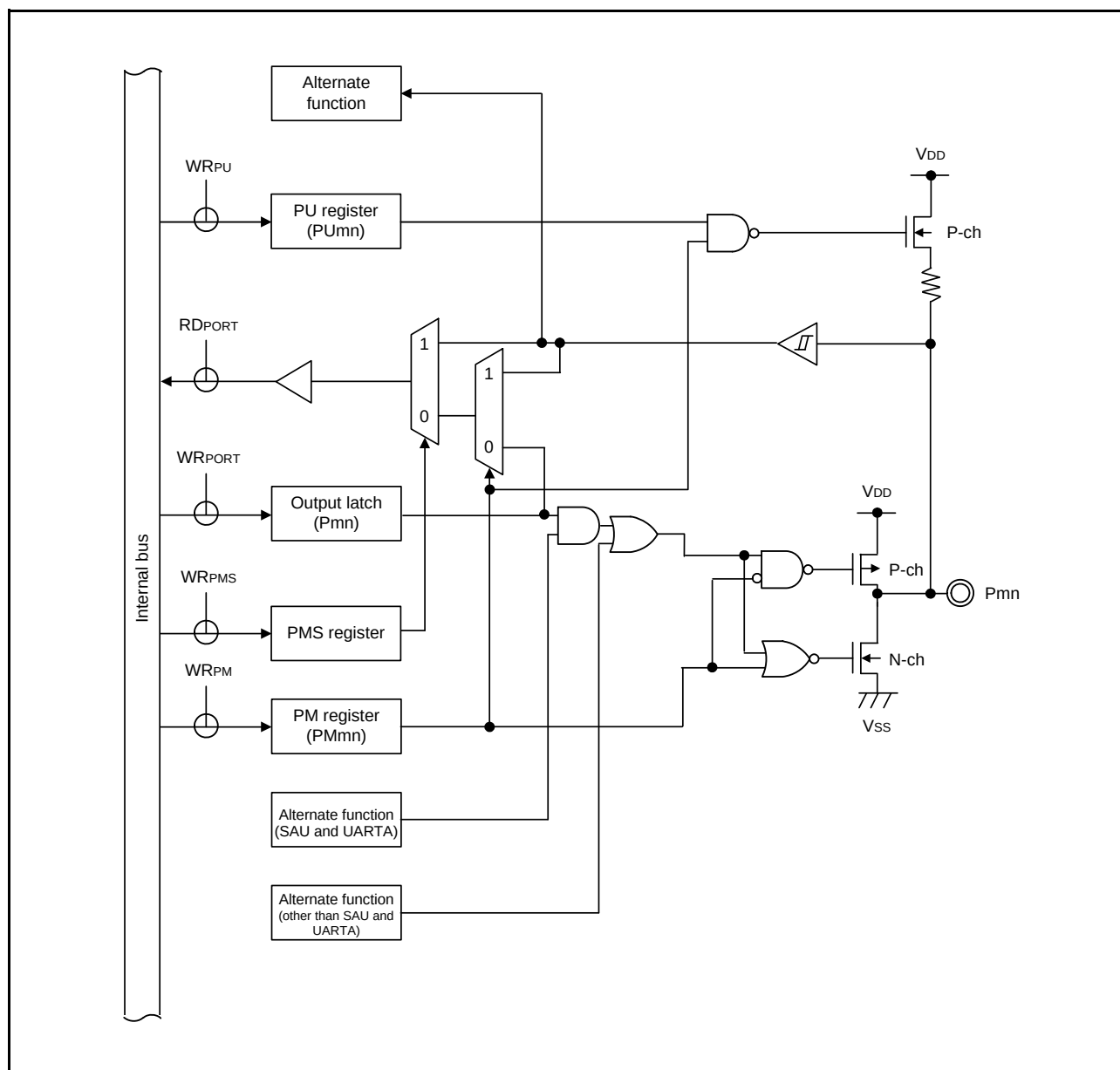
[illegible]

Remark 2. SAU: Serial array unit

[illegible]

Remark 2. SAU: Serial array unit

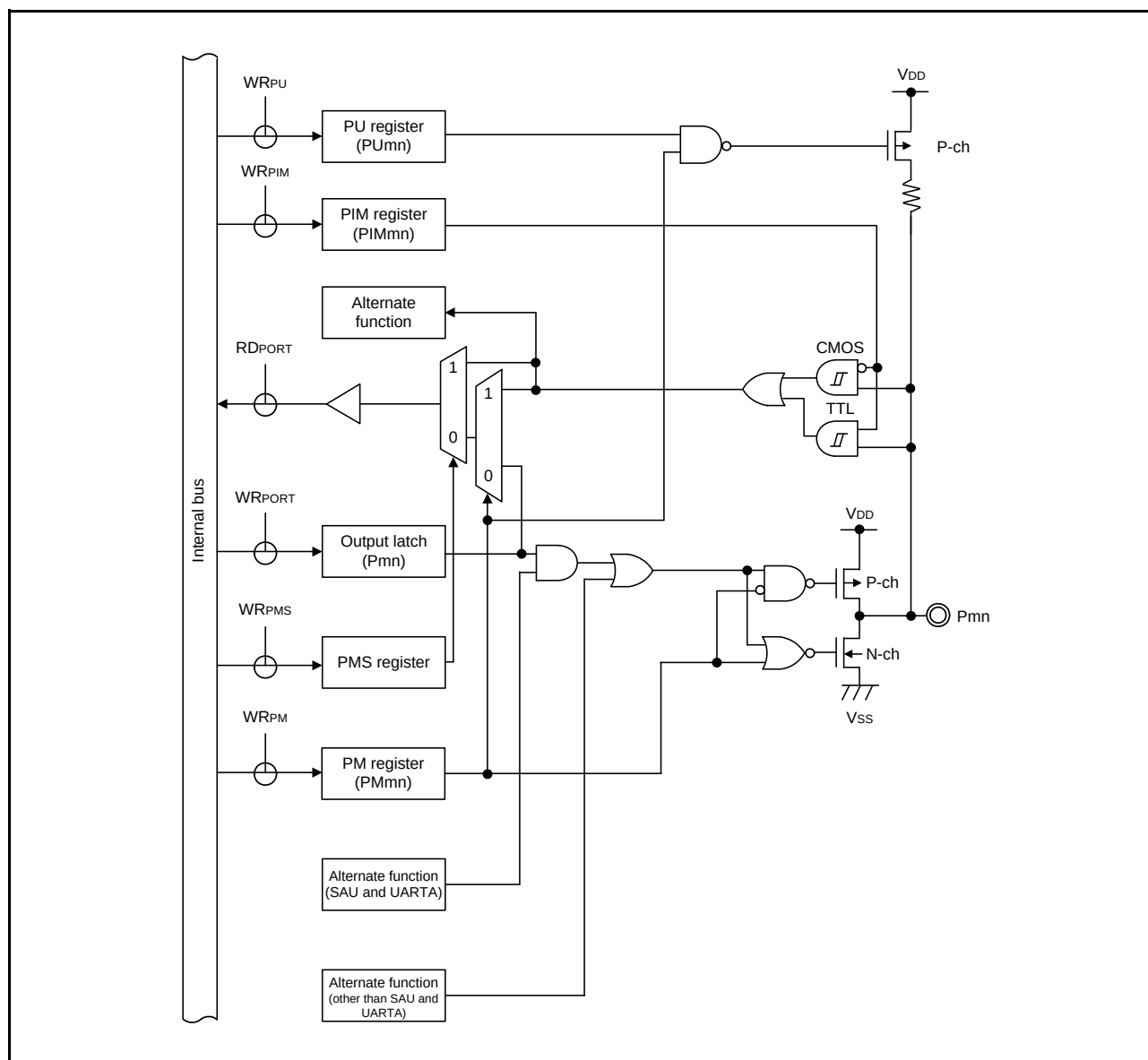
Figure 2 - 17 Pin Block Diagram for Pin Type 7-38-1



Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 18 Pin Block Diagram for Pin Type 8-1-3

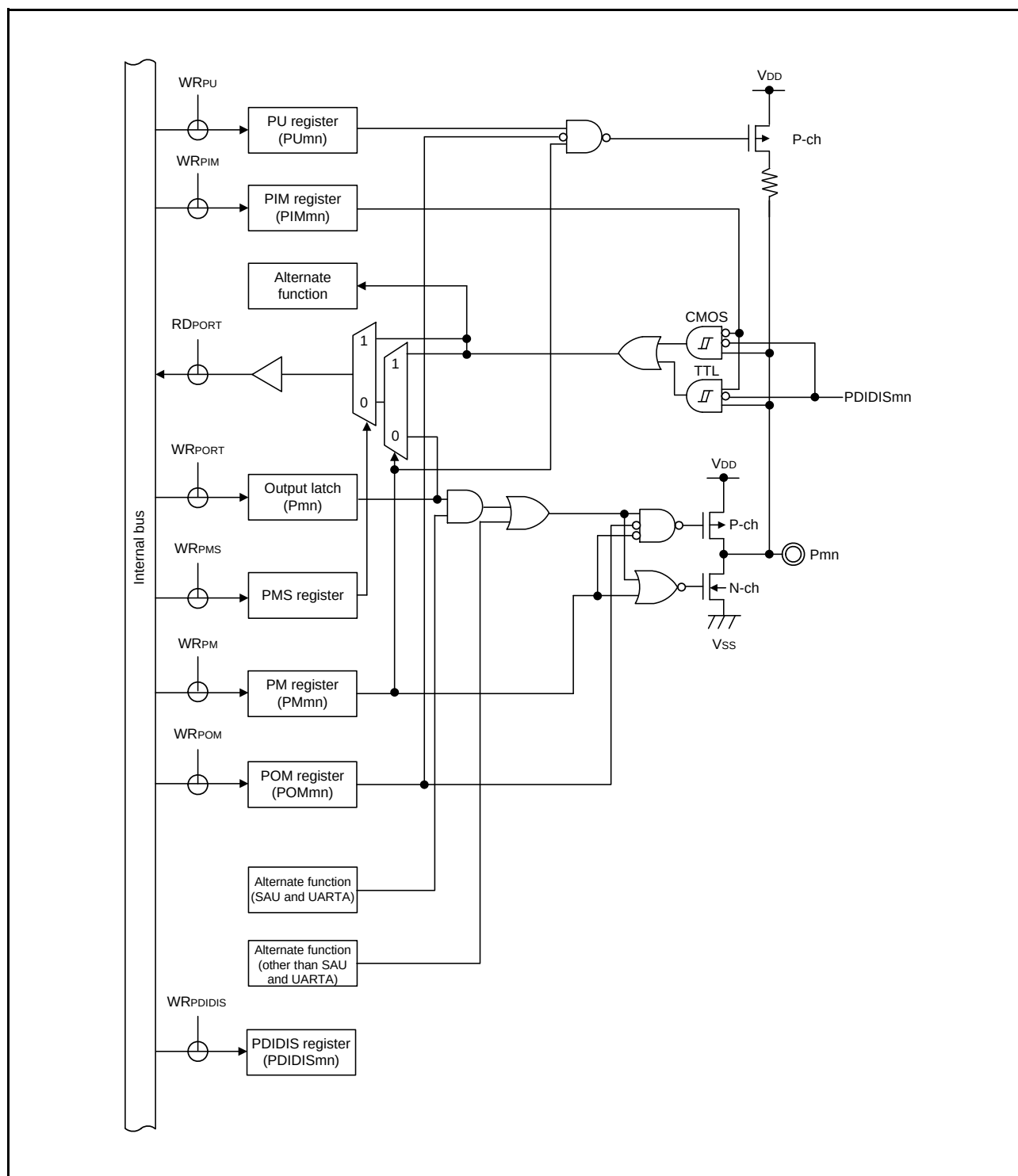


Caution When the type 8-1-3 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-1-3 pin due to the configuration of the TTL input buffer. Drive the type 8-1-3 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 19 Pin Block Diagram for Pin Type 8-1-11



(Cautions and Remarks are listed on the next page.)

Caution 1. The input buffer is enabled even if the type 8-1-11 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-1-11 pin when the voltage level on this pin is intermediate.

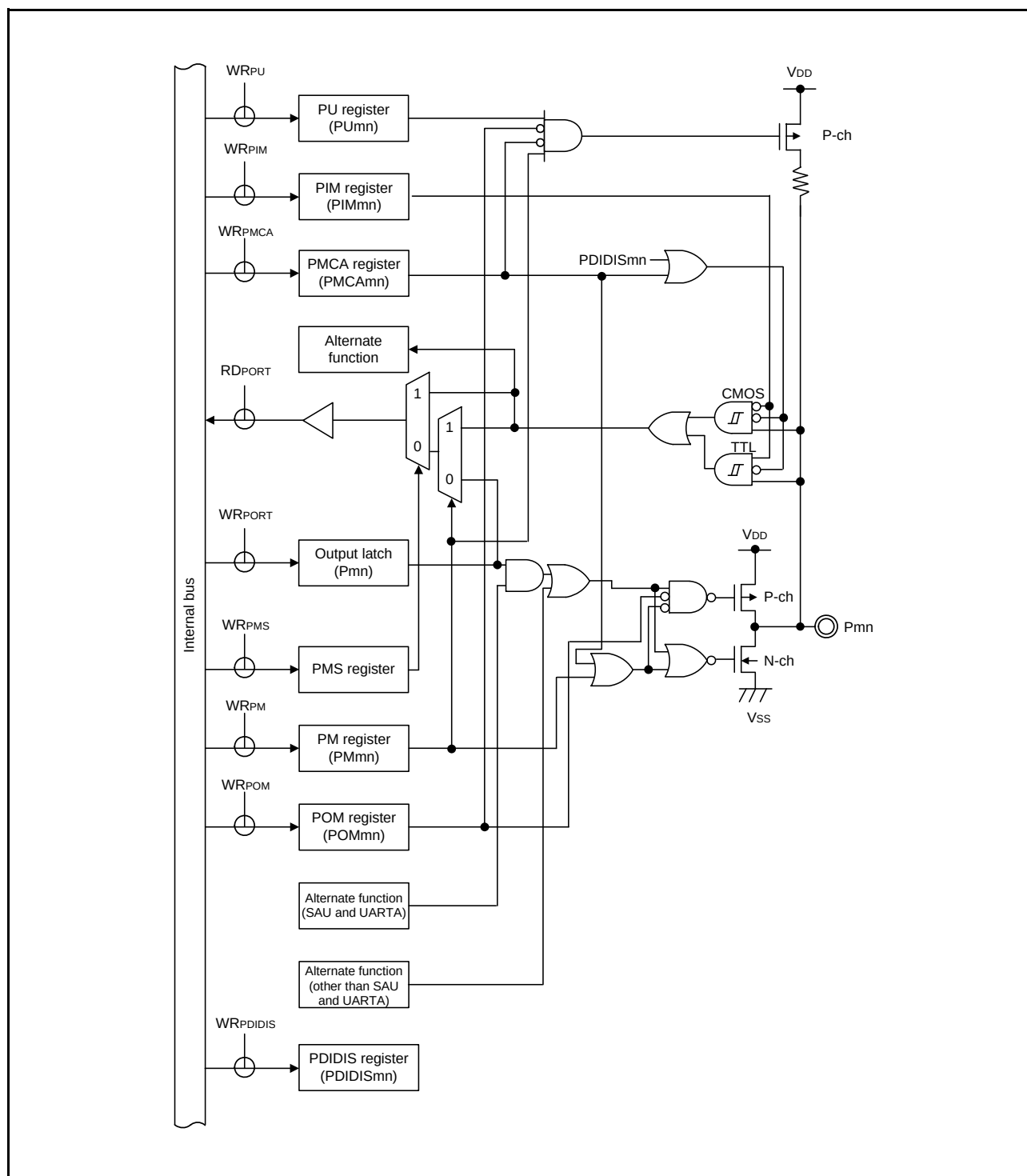
However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Caution 2. When the type 8-1-11 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-1-11 pin due to the configuration of the TTL input buffer. Drive the type 8-1-11 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 20 Pin Block Diagram for Pin Type 8-6-9



(Cautions and Remarks are listed on the next page.)

Caution 1. The input buffer is enabled even if the type 8-6-9 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-6-9 pin when the voltage level on this pin is intermediate.

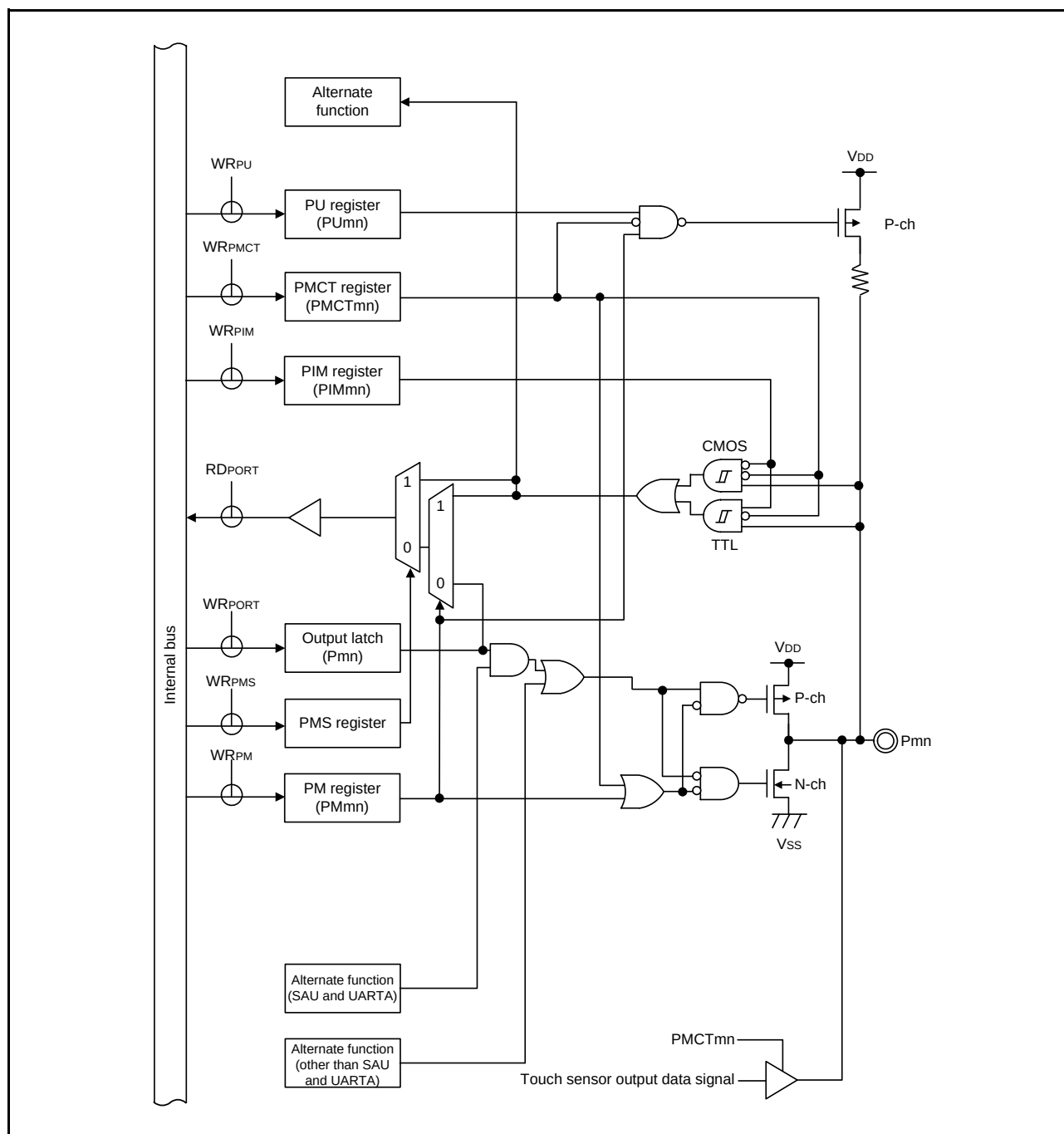
However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Caution 2. When the type 8-6-9 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-6-9 pin due to the configuration of the TTL input buffer. Drive the type 8-6-9 pin low to prevent the through current.

Remark 1. For alternate functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 21 Pin Block Diagram for Pin Type 8-31-1

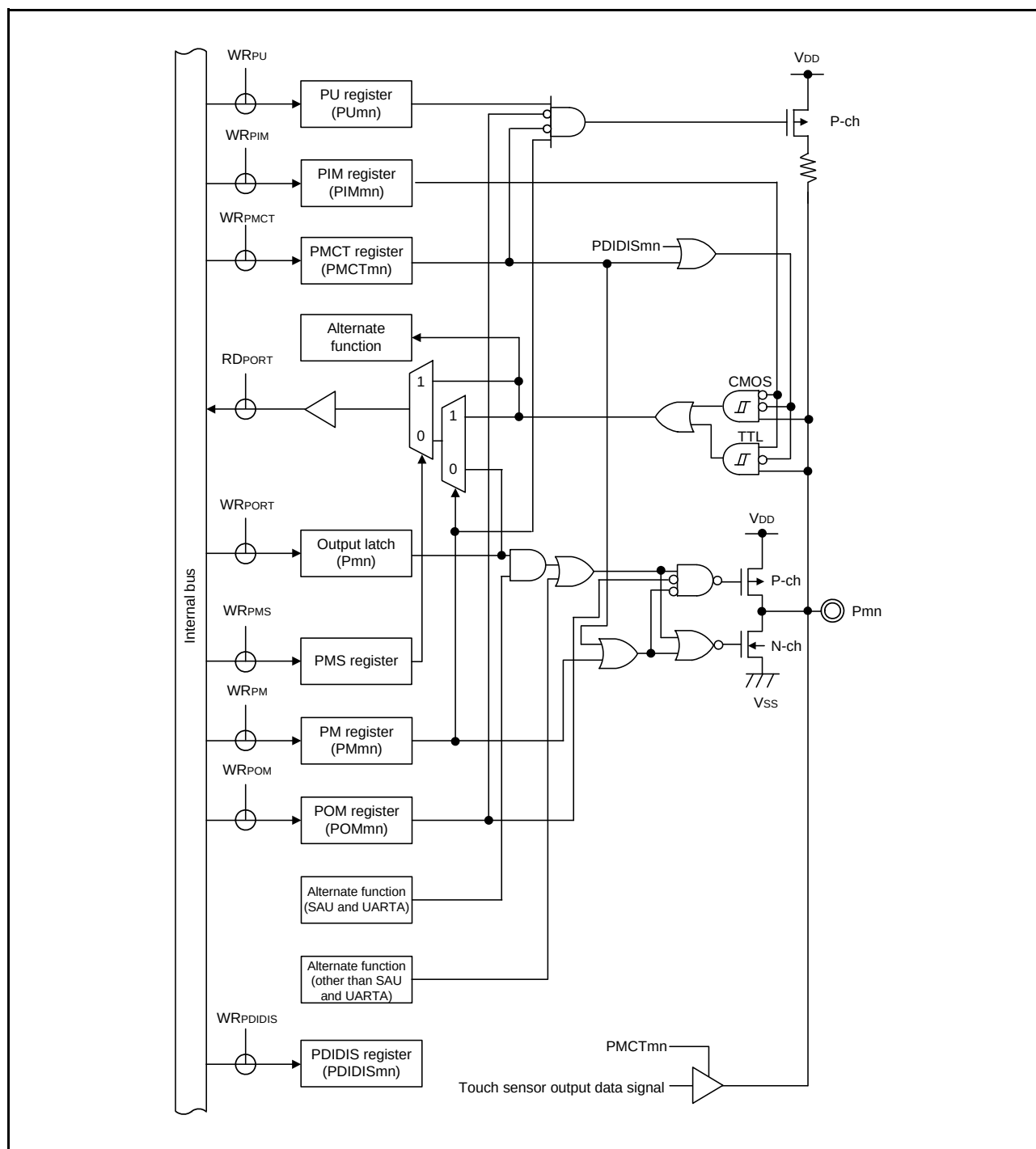


Caution When the type 8-31-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-31-1 pin due to the configuration of the TTL input buffer. Drive the type 8-31-1 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 22 Pin Block Diagram for Pin Type 8-31-2



(Cautions and Remarks are listed on the next page.)

Caution 1. The input buffer is enabled even if the type 8-31-2 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-31-2 pin when the voltage level on this pin is intermediate.

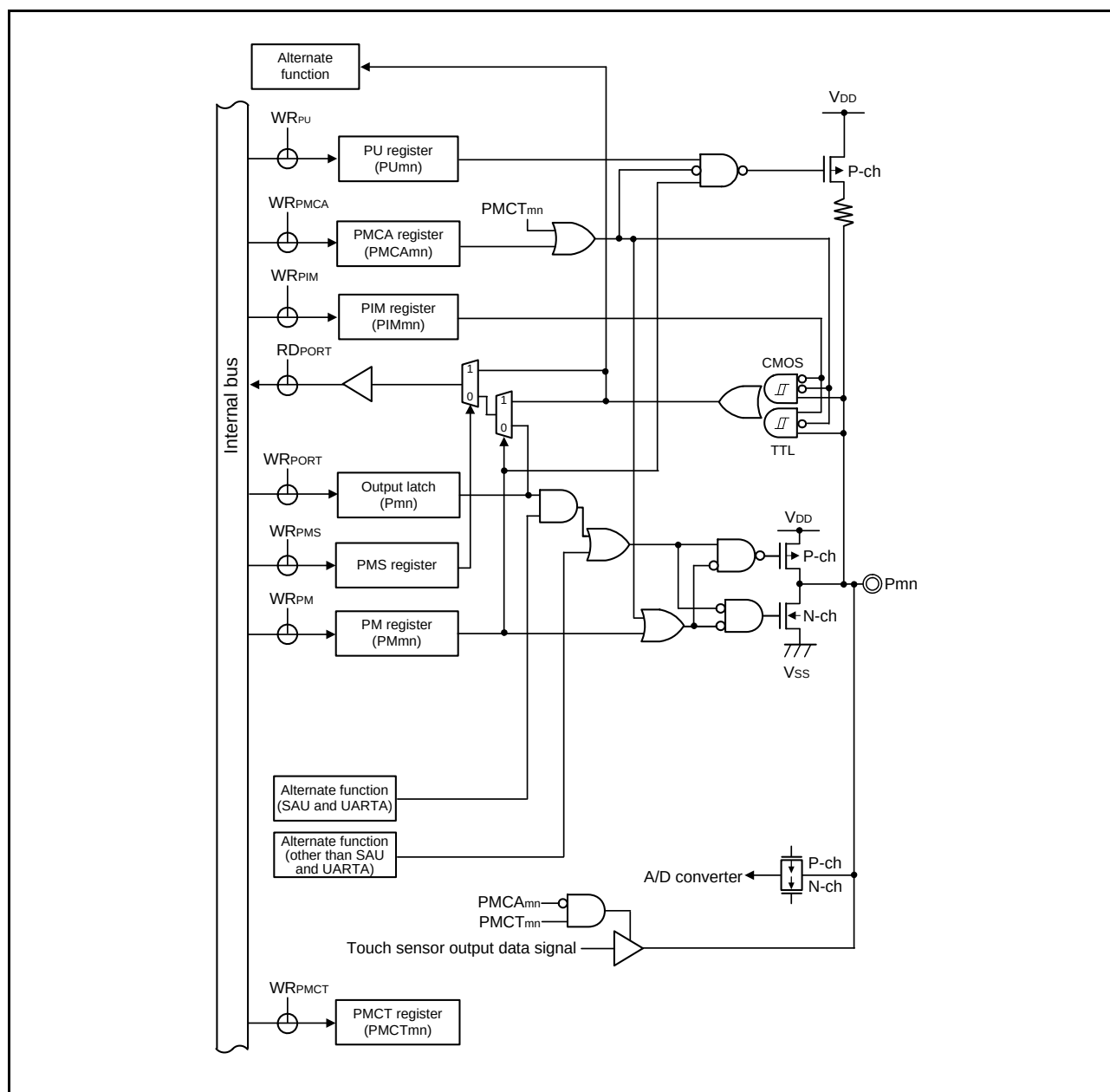
However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Caution 2. When the type 8-31-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-31-2 pin due to the configuration of the TTL input buffer. Drive the type 8-31-2 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 23 Pin Block Diagram for Pin Type 8-33-2

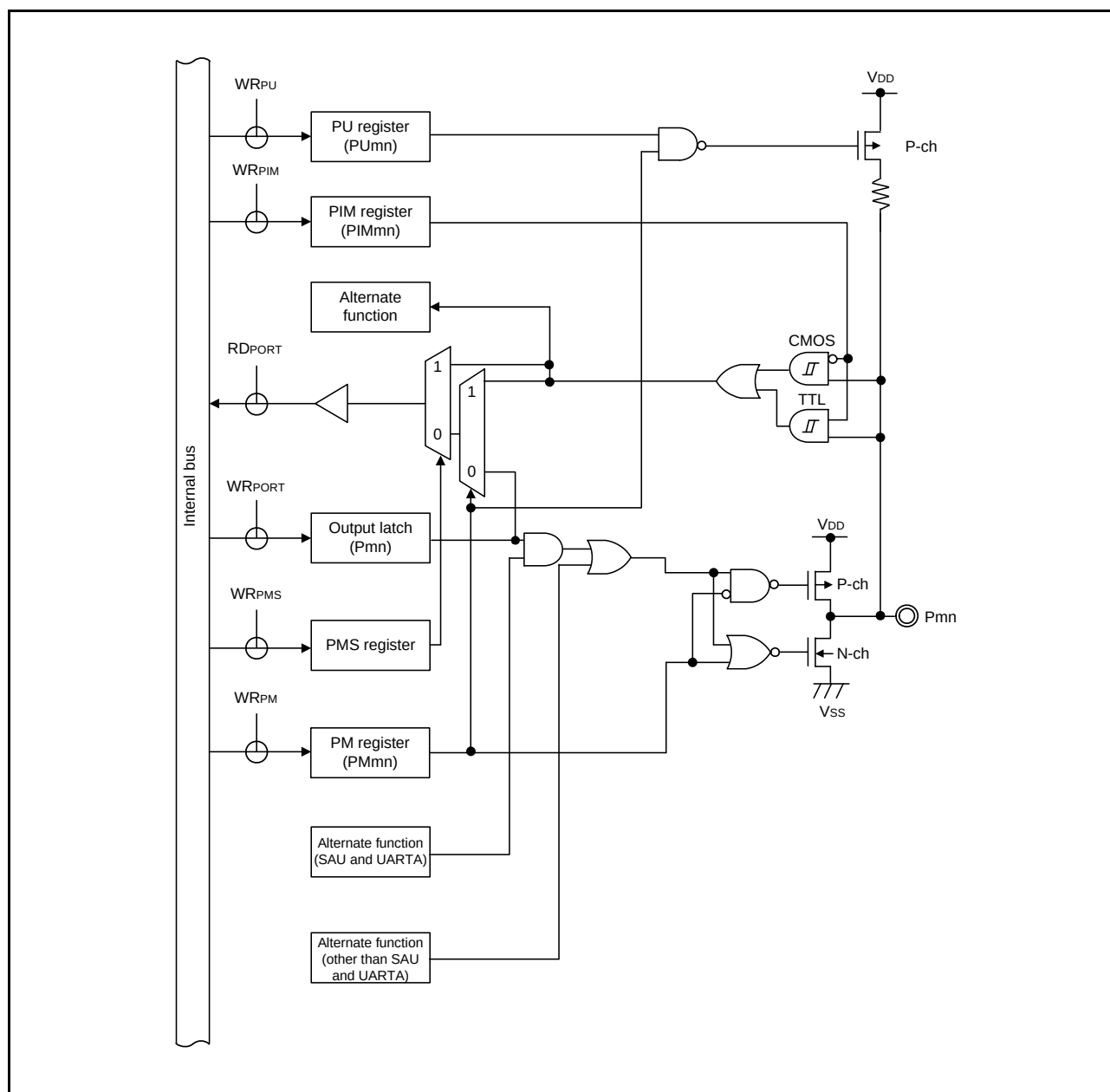


Caution When the type 8-33-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-33-2 pin due to the configuration of the TTL input buffer. Drive the type 8-33-2 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 24 Pin Block Diagram for Pin Type 8-38-1

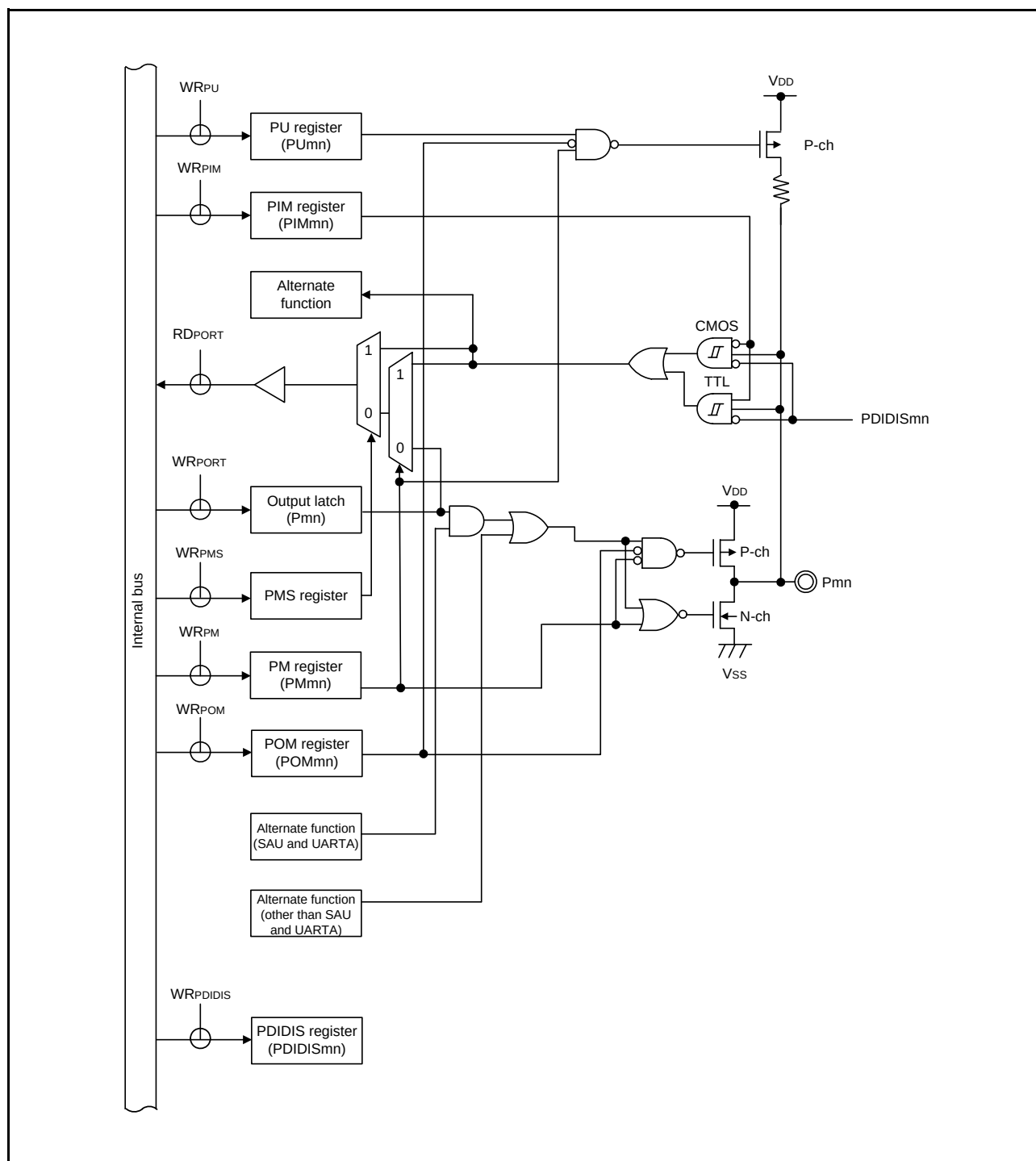


Caution When the type 8-38-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-38-1 pin due to the configuration of the TTL input buffer. Drive the type 8-38-1 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

Figure 2 - 25 Pin Block Diagram for Pin Type 8-38-2



Caution 1. The input buffer is enabled even if the type 8-38-2 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-38-2 pin when the voltage level on this pin is intermediate.

However, setting the corresponding bit of the given PDIDISx register to 1 prevents the flow of a through current.

Caution 2. When the type 8-38-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-38-2 pin due to the configuration of the TTL input buffer. Drive the type 8-38-2 pin low to prevent the through current.

Remark 1. For alternate functions, see 2.1 Functions of Port Pins.

Remark 2. SAU: Serial array unit

[illegible]

Remark 2. SAU: Serial array unit

The schematic diagram illustrates the internal architecture of the PM module. An **Internal bus** on the left provides control signals: **WRPER0**, **RDPORT**, **WRPORT**, **WRPMS**, and **WRPm**. These signals are connected to several registers and functional blocks:

- PER0 (IICAjEN)**: Receives **WRPER0** and outputs to the **Alternate function** block.
- Output latch (Pmn)**: Receives **WRPORT** and its output is connected to a multiplexer.
- PMS register**: Receives **WRPMS** and its output is connected to a multiplexer and an AND gate.
- PM register (PMmn)**: Receives **WRPm** and its output is connected to an AND gate.
- Alternate function (SAU and UARTA)**: Receives signals from the **Alternate function** block and the **PM register**.
- Alternate function (other than SAU and UARTA)**: Receives signals from the **Alternate function** block and the **PM register**.

The output driver consists of an **N-ch** transistor connected to **Vss**. The gate of the transistor is driven by the output of an AND gate, which is the logical AND of the outputs of the **PMS register** and the **PM register**. The source of the transistor is connected to **Vss**, and the drain is connected to the **Pmn** pin. The **Pmn** pin is also connected to the **Alternate function** block.

Remark 2. SAU: Serial array unit

Section 3 CPU Architecture

The RL78/G22 is a microcontroller that has the RL78-S3 CPU core.

The CPU core in the RL78-S3 employs the Harvard architecture which has independent instruction fetch bus, address bus and data bus. In addition, through the adoption of three-stage pipeline control of fetch, decode, and memory access, the operation efficiency is remarkably improved over the conventional CPU core. The CPU core features high performance and highly functional instruction processing, and can be suited for use in various applications that require high speed and highly functional processing.

- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time: One instruction per clock cycle
- General-purpose registers: Eight 8-bit registers
- Type of instruction: 81

The following multiply/divide instructions are available only in the RL78-S3 CPU core.

MULHU (unsigned 16-bit multiplication)

MULH (signed 16-bit multiplication)

DIVHU (unsigned 16-bit division)

DIVWU (unsigned 32-bit division)

MACHU (unsigned multiplication/accumulation (16 bits × 16 bits) + 32 bits)

MACH (signed multiplication/accumulation (16 bits × 16 bits) + 32 bits)

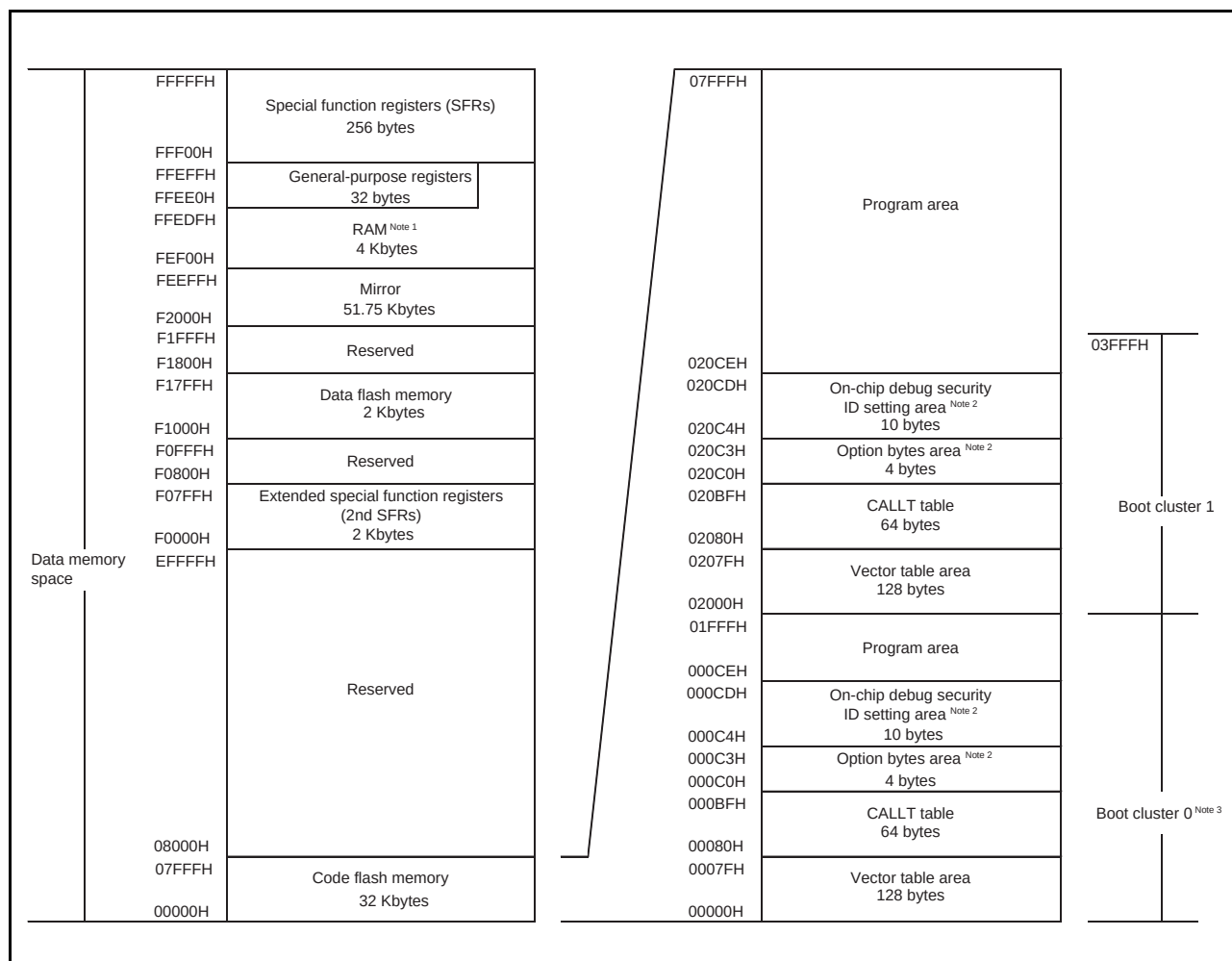
- Data allocation: Little endian

The RL78/G22 does not support OCD tracing.

3.1 Memory Space

Products in the RL78/G22 can access a 1 MB address space. **Figures 3 - 1** and **3 - 2** show the memory maps.

Figure 3 - 1 Memory Map (R7F102GxC (x = 4, 6, 7, 8, A, B, C, E, F, G))



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

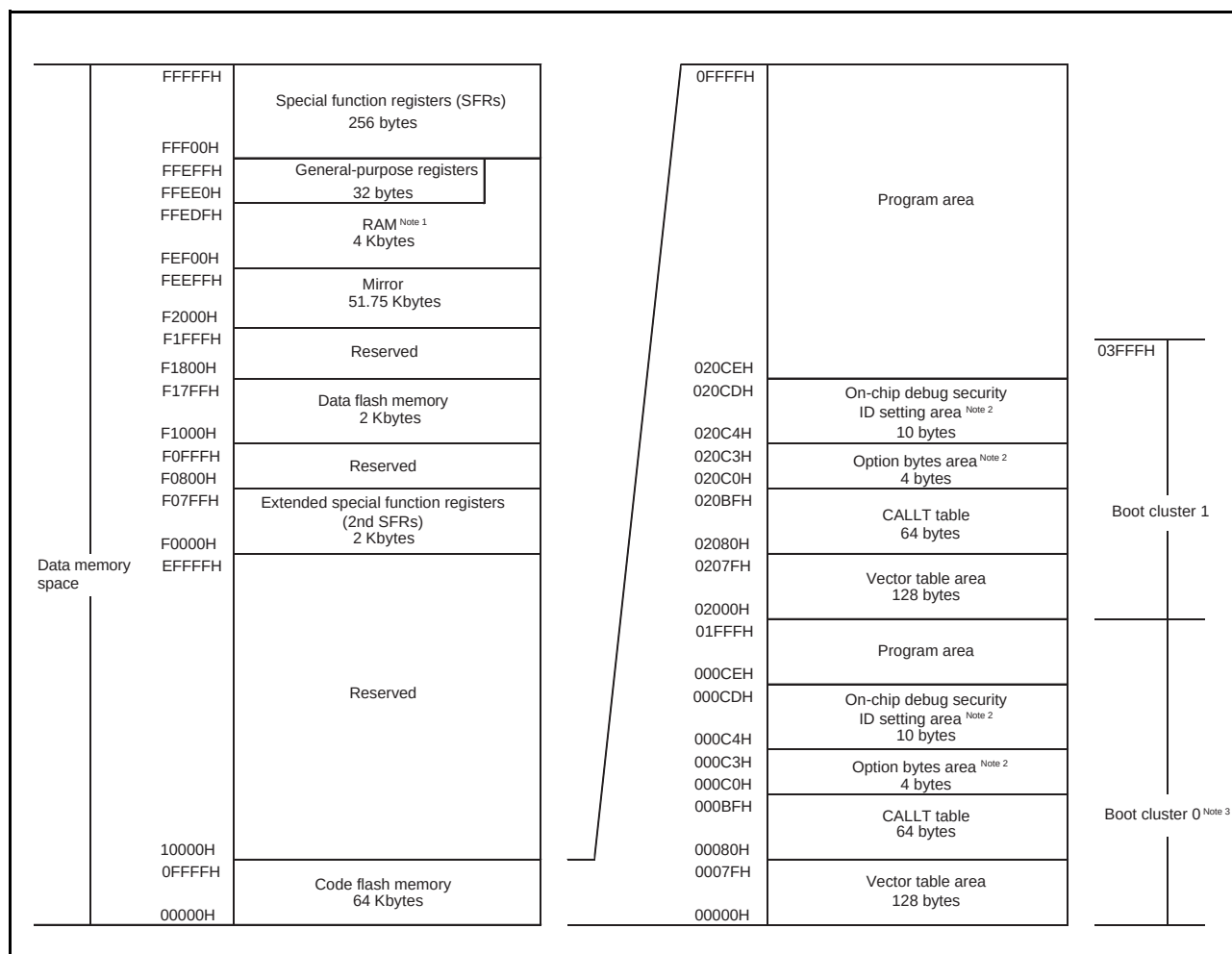
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

Note 3. Writing boot cluster 0 can be prohibited depending on the setting of security. See **30.9 Security Settings**.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see **24.3.4 RAM parity error detection**.

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1**.

Figure 3 - 2 Memory Map (R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G))



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

Note 3. Writing boot cluster 0 can be prohibited depending on the setting of security. See **30.9 Security Settings**.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.
RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see **24.3.4 RAM parity error detection**.

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1**.

Correspondence between the addresses and block numbers in the flash memory are shown below.

Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory

Address	Block Number
00000H to 007FFH	000H
00800H to 00FFFH	001H
01000H to 017FFH	002H
01800H to 01FFFH	003H
02000H to 027FFH	004H
02800H to 02FFFH	005H
03000H to 037FFH	006H
03800H to 03FFFH	007H
04000H to 047FFH	008H
04800H to 04FFFH	009H
05000H to 057FFH	00AH
05800H to 05FFFH	00BH
06000H to 067FFH	00CH
06800H to 06FFFH	00DH
07000H to 077FFH	00EH
07800H to 07FFFH	00FH
08000H to 087FFH	010H
08800H to 08FFFH	011H
09000H to 097FFH	012H
09800H to 09FFFH	013H
0A000H to 0A7FFH	014H
0A800H to 0AFFFH	015H
0B000H to 0B7FFH	016H
0B800H to 0BFFFH	017H
0C000H to 0C7FFH	018H
0C800H to 0CFFFH	019H
0D000H to 0D7FFH	01AH
0D800H to 0DFFFH	01BH
0E000H to 0E7FFH	01CH
0E800H to 0EFFFH	01DH
0F000H to 0F7FFH	01EH
0F800H to 0FFFFH	01FH

Remark R7F102GxC (x = 4, 6, 7, 8, A, B, C, E, F, G): Block numbers 000H to 00FH
R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G): Block numbers 000H to 01FH

3.1.1 Internal program memory space

The internal program memory holds the program and table data.

The RL78/G22 products incorporate the internal ROM (flash memory) with the capacity shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F102GxC (x = 4, 6, 7, 8, A, B, C, E, F, G)	Flash memory	32768 × 8 bits (00000H to 07FFFH)
R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G)		65536 × 8 bits (00000H to 0FFFFH)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump address is a 64 K address of 00000H to 0FFFFH, because the vector code is assumed to be 2 bytes.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

To use the boot swap function, set a vector table also at 02000H to 0207FH.

Table 3 - 3 lists the vector table. “√” indicates an interrupt source which is supported. “—” indicates an interrupt source which is not supported.

The vector table address can be changed to an address in RAM when self-programming is to proceed. For details, see **30.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

Table 3 - 3 Vector Table (1/2)

Vector Table Address	Interrupt Source	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√	√	√	√	√	√	√	√
00004H	INTWDTI	√	√	√	√	√	√	√	√	√	√
00006H	INTLVI	√	√	√	√	√	√	√	√	√	√
00008H	INTP0	√	√	√	√	√	√	√	√	√	√
0000AH	INTP1	√	√	√	√	√	√	√	√	—	—
0000CH	INTP2	√	√	√	√	√	√	—	—	—	—
0000EH	INTP3	√	√	√	√	√	√	√	√	√	√
00010H	INTP4	√	√	√	√	√	√	√	√	—	—
00012H	INTP5	√	√	√	√	√	√	√	√	√	—
00014H	INTST2/INTCSI20/INTIIC20	√	√	√	√	√	√	—	—	—	—
00016H	INTSR2/INTCSI21/INTIIC21	√	√	√	√	Note 1	Note 1	—	—	—	—
00018H	INTSRE2	√	√	√	√	√	√	—	—	—	—
0001AH	—	—	—	—	—	—	—	—	—	—	—
0001CH	INTSMSE	√	√	√	√	√	√	√	√	√	√
0001EH	INTST0/INTCSI00/INTIIC00	√	√	√	√	√	√	√	√	√	√
00020H	INTTM00	√	√	√	√	√	√	√	√	√	—
00022H	INTSRE0	√	√	√	√	√	√	√	√	√	√
	INTTM01H	√	√	√	√	√	√	√	√	√	—
00024H	INTST1	√	√	√	√	√	√	√	√	√	√
00026H	INTSR1/INTCSI11/INTIIC11	√	√	√	√	√	√	√	√	√	√
00028H	INTSRE1	√	√	√	√	√	√	√	√	√	—
	INTTM03H	√	√	√	√	√	√	√	√	—	—
0002AH	INTIICA0	√	√	√	√	√	√	√	√	—	—
0002CH	INTSR0/INTCSI01/INTIIC01	√	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2	Note 2
0002EH	INTTM01	√	√	√	√	√	√	√	√	√	—
00030H	INTTM02	√	√	√	√	√	√	√	√	√	√
00032H	INTTM03	√	√	√	√	√	√	√	√	—	—
00034H	INTAD	√	√	√	√	√	√	√	√	√	√
00036H	INTRTC	√	√	√	√	√	√	√	√	√	√
00038H	INTITL	√	√	√	√	√	√	√	√	√	√
0003AH	INTKR	√	√	√	—	—	—	—	—	—	—
0003CH	—	—	—	—	—	—	—	—	—	—	—
0003EH	—	—	—	—	—	—	—	—	—	—	—
00040H	—	—	—	—	—	—	—	—	—	—	—
00042H	INTTM04	√	√	√	√	√	√	—	—	—	—
00044H	INTTM05	√	√	√	√	√	√	—	—	—	—
00046H	INTTM06	√	√	√	√	√	√	—	—	—	—
00048H	INTTM07	√	√	√	√	√	√	—	—	—	—
0004AH	INTP6	√	—	—	—	—	—	—	—	—	—

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
0004CH	—	—	—	—	—	—	—	—	—	—	—
0004EH	INTP8	√	—	—	—	—	—	—	—	—	—
00050H	INTP9	√	—	—	—	—	—	—	—	—	—
00052H	INTFL	√	√	√	√	√	√	√	√	√	√
00054H	—	—	—	—	—	—	—	—	—	—	—
00056H	—	—	—	—	—	—	—	—	—	—	—
00058H	INTURE0	√	√	√	√	—	—	—	—	—	—
0005AH	—	—	—	—	—	—	—	—	—	—	—
0005CH	—	—	—	—	—	—	—	—	—	—	—
0005EH	—	—	—	—	—	—	—	—	—	—	—
00060H	INTCTSUWR	√	√	√	√	√	√	√	√	√	√
00062H	—	—	—	—	—	—	—	—	—	—	—
00064H	INTCTSURD	√	√	√	√	√	√	√	√	√	√
00066H	INTCTSUFN	√	√	√	√	√	√	√	√	√	√
00068H	—	—	—	—	—	—	—	—	—	—	—
0006AH	INTUTO	√	√	√	√	—	—	—	—	—	—
0006CH	INTURO	√	√	√	√	—	—	—	—	—	—
0006EH	—	—	—	—	—	—	—	—	—	—	—
00070H	—	—	—	—	—	—	—	—	—	—	—
00072H	—	—	—	—	—	—	—	—	—	—	—
00074H	—	—	—	—	—	—	—	—	—	—	—
00076H	—	—	—	—	—	—	—	—	—	—	—
00078H	—	—	—	—	—	—	—	—	—	—	—
0007EH	BRK	√	√	√	√	√	√	√	√	√	√

Note 1. INTSR2 is only present in this product.

Note 2. INTSR0 is only present in this product.

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

To use the boot swap function, set a CALLT instruction table also at 02080H to 020BFH.

(3) Option bytes area

A 4-byte area of 000C0H to 000C3H can be used as an option bytes area. Set the option byte at 020C0H to 020C3H when the boot swap is used. For details, see **Section 29 Option Bytes**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 020C4H to 020CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and at 020C4H to 020CDH when the boot swap is used. For details, see **Section 31 On-chip Debugging**.

3.1.2 Mirror area

In products of the RL78/G22, the code flash area from 00000H to 0FFFFH is mirrored at F0000H to FFFFFH. The setting of the processor mode control register (PMC) determines whether the code flash area from 00000H to 0FFFFH is mirrored.

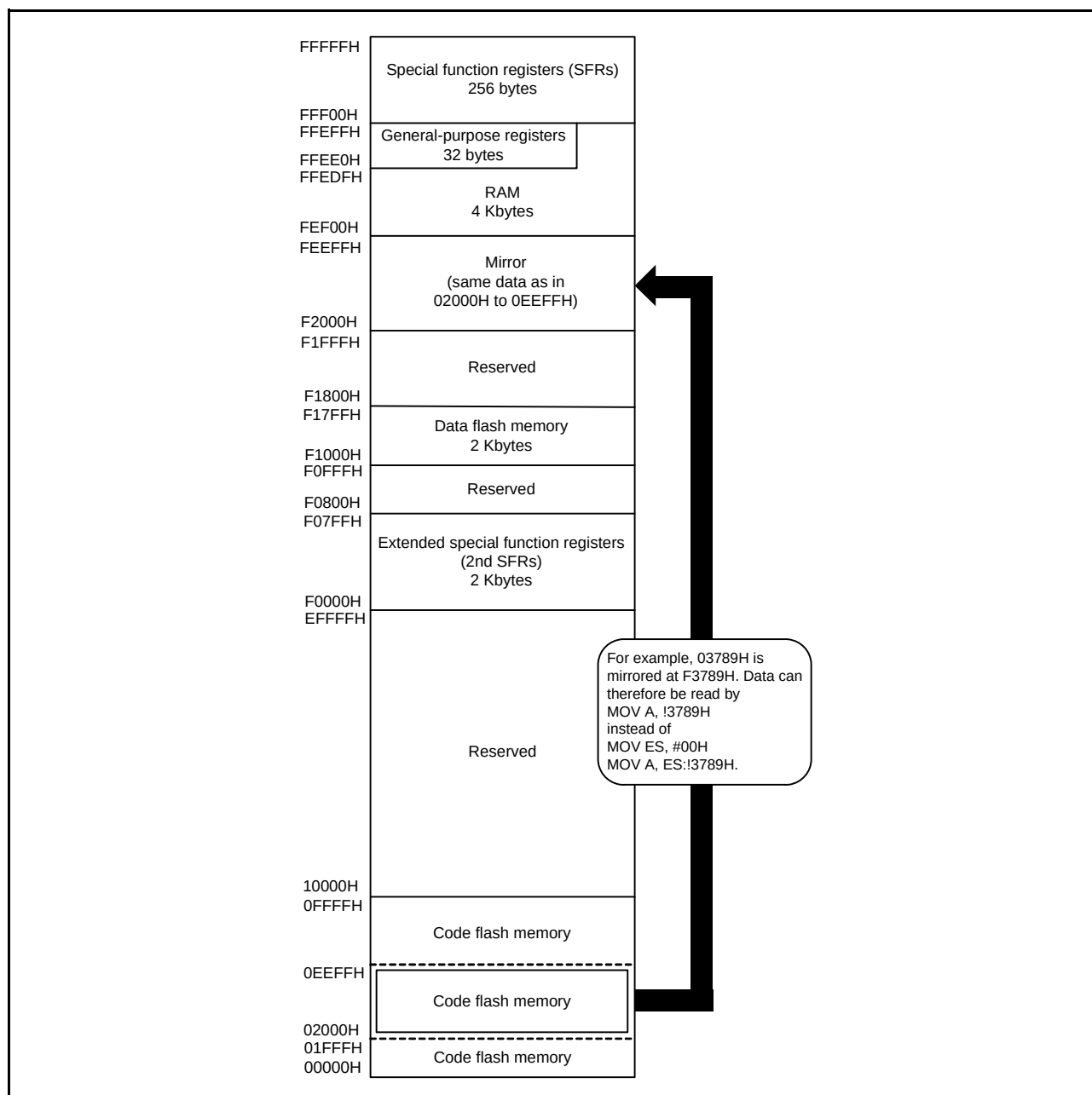
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function registers (SFRs), extended special function registers (2nd SFRs), RAM, data flash memory, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Figure 3 - 3 R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G) (Flash memory: 64 Kbytes, RAM: 4 Kbytes)



The PMC register is described below.

- Processor mode control register (PMC)

This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 3 - 4 Format of Processor Mode Control Register (PMC)

Address: FFFFEH

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA
MAA	Flash memory area from F0000H to FFFFFH mirror usage selection							
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH							
1	Setting prohibited							

Caution 1. After setting the PMC register, wait for at least one instruction and access the mirror area.

Caution 2. When boot swapping is executed while bit 0 (MAA) is 0, data at 02000H to 03FFFH are mirrored at F2000H to F3FFFH after boot swapping.

3.1.3 Internal data memory space

The RL78/G22 products incorporate the RAM with the capacity shown below.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R7F102GxC (x = 4, 6, 7, 8, A, B, C, E, F, G)	4096 × 8 bits (FEF00H to FFEFFH)
R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G)	4096 × 8 bits (FEF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (the space to which the general-purpose registers are allocated cannot be used for instruction fetching). Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.

The internal RAM is used as stack memory.

Caution The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH. See **Table 3 - 5** in **3.2.4 Special function registers (SFRs)**.

Caution Do not access addresses to which SFRs are not assigned.

3.1.5 Extended special function register (2nd SFR: 2nd special function register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH. See **Table 3 - 6** in **3.2.5 Extended special function registers (2nd SFRs)**.

Caution 1. Do not access addresses to which extended SFRs are not assigned.

Caution 2. In the area for the extended special function registers (2nd SFRs), the registers of the capacitive sensing unit (CTSU2La) are allocated to the address range from F0500H to F0535H and the registers of the true random number generator (TRNG) are allocated to F0540H and F0542H. The CPU is placed in the wait state and does not proceed to the next instruction during access to the register range for either module. Accordingly, the CPU entering this state lengthens the number of clock cycles to execute an instruction by the number of cycles of waiting. Specifically, the CPU waits for one clock cycle during access (whether reading or writing) to the registers of the capacitive sensing unit (CTSU2La) and true random number generator (TRNG).

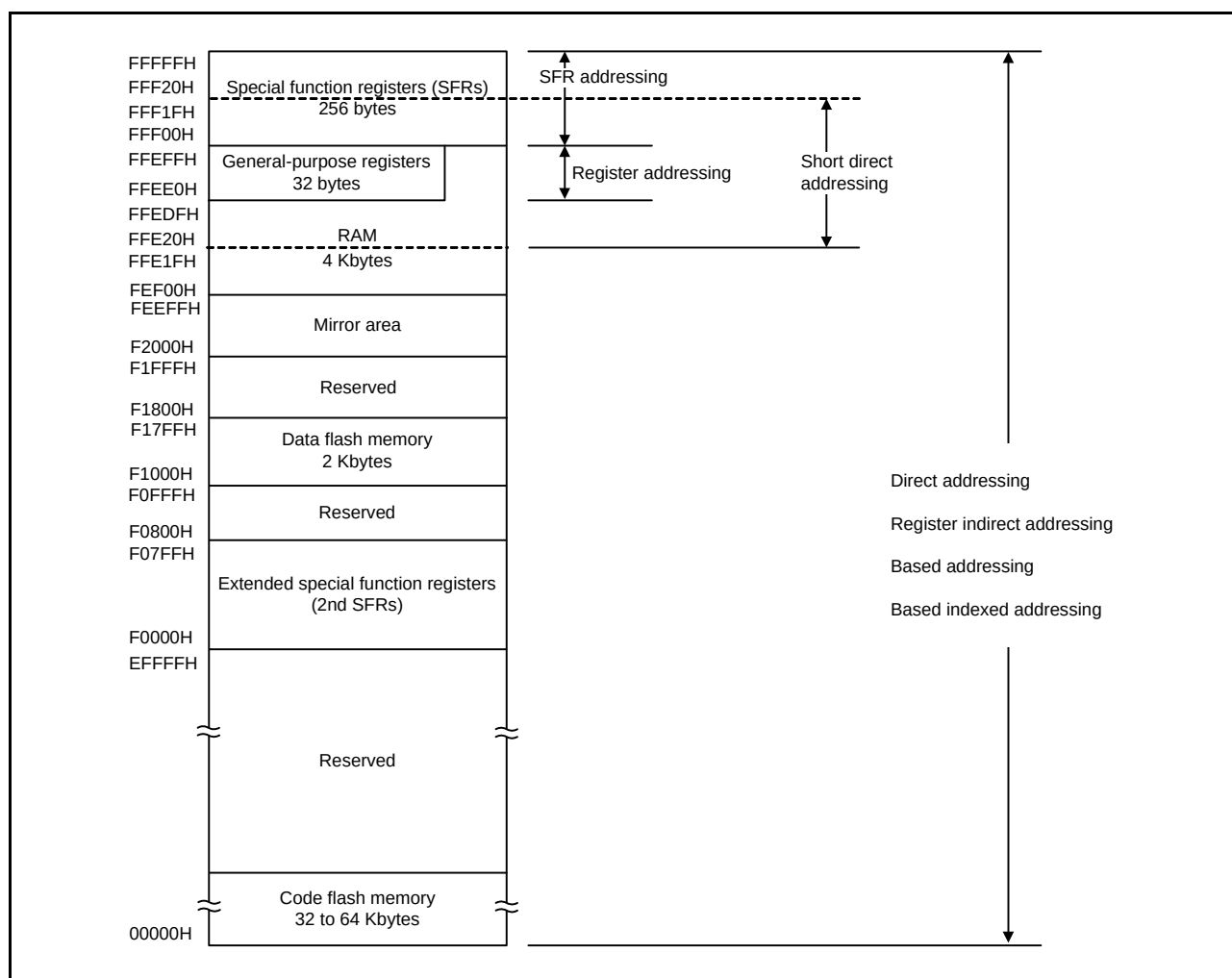
3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/G22, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFRs) and general-purpose registers are available for use. **Figure 3 - 5** shows correspondence between data memory and addressing.

For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 5 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/G22 products incorporate the following processor registers.

3.2.1 Control registers

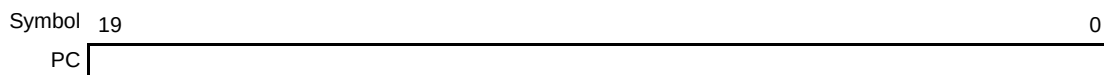
The control registers control the program sequence, state, and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

The program counter is a 20-bit register that holds the address information of the next program to be executed. In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

The values of the reset vector table at addresses 0000H and 0001H are set in the program counter following a reset.

Figure 3 - 6 Format of Program Counter

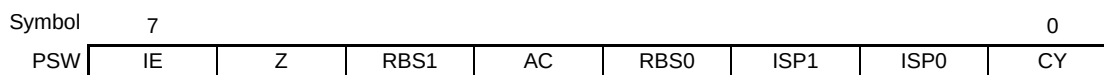


(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. The value of the PSW following a reset is 06H.

Figure 3 - 7 Format of Program Status Word



(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

(c) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flags (ISP1, ISP0)

These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H, PRn3L, and PRn3H; see **18.3.3 Priority specification flag registers**) cannot be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

Figure 3 - 8 Format of Stack Pointer

Symbol	15	14	13	12	11	10	9	8
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8
	7	6	5	4	3	2	1	0
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

Caution 1. Since the contents of the SP become undefined following a reset, be sure to initialize the SP before using the stack.

Caution 2. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consist of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

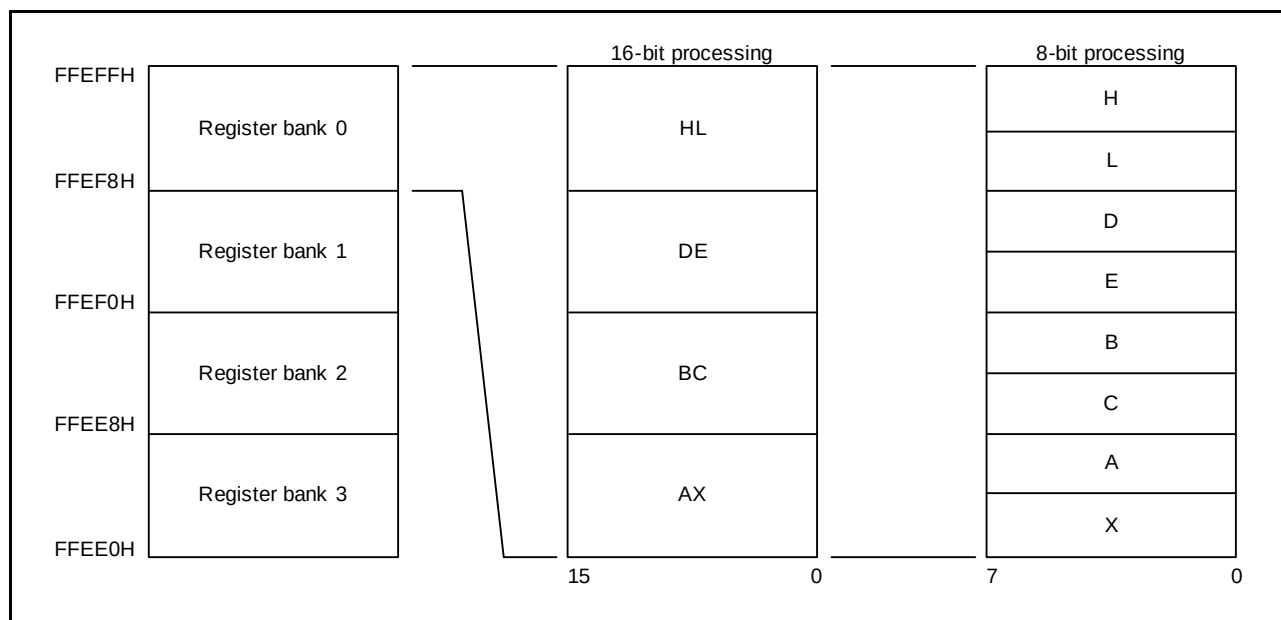
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt processing for each bank.

Caution The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

Figure 3 - 9 Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register indirect addressing), respectively.

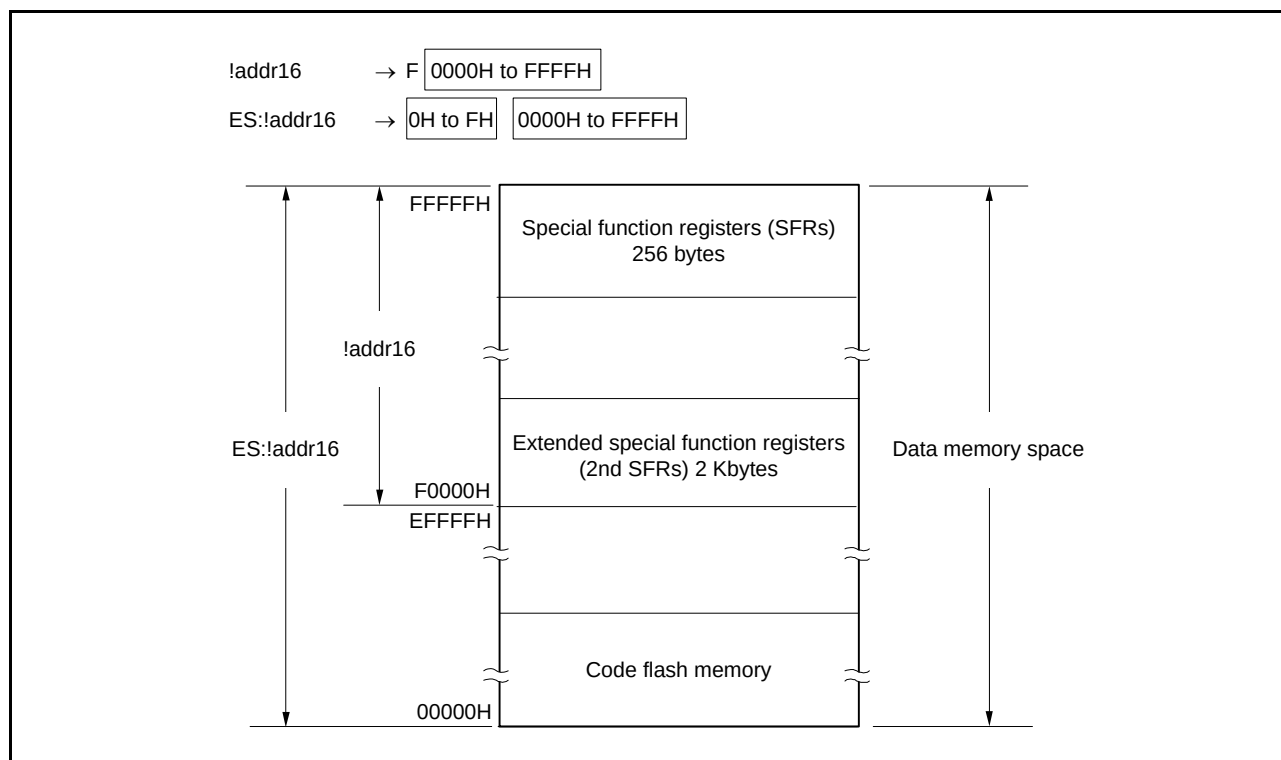
The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3 - 10 Configuration of ES and CS Registers

Symbol	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0
Symbol	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	ES1	ES0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3 - 11 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions.

The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
Describe as follows for the 1-bit manipulation instruction operand (sfr.bit).
When the bit name is defined: <Bit name>
When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 16-bit manipulation
Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Table 3 - 5 give lists of the SFRs. The meanings of items in the table are as follows.

- Symbol
This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
This item indicates whether the corresponding SFR can be read or written.
R/W: Read/write enable
R: Read only
W: Write only
- Manipulable bit units
“√” indicates the manipulable bit unit (1, 8, or 16). “—” indicates a bit unit for which manipulation is not possible.
- After reset
Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution Do not access addresses to which SFRs are not assigned.

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs)**.

Table 3 - 5 List of Special Function Registers (SFRs) (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF00H	Port register 0	P0		R/W	√	√	—	00H
FFF01H	Port register 1	P1		R/W	√	√	—	00H
FFF02H	Port register 2	P2		R/W	√	√	—	00H
FFF03H	Port register 3	P3		R/W	√	√	—	00H
FFF04H	Port register 4	P4		R/W	√	√	—	00H
FFF05H	Port register 5	P5		R/W	√	√	—	00H
FFF06H	Port register 6	P6		R/W	√	√	—	00H
FFF07H	Port register 7	P7		R/W	√	√	—	00H
FFF0CH	Port register 12	P12		R/W	√	√	—	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	—	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	—	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	√	√	00H
FFF1BH		TDR01H			—	√	—	
FFF1EH	10-bit A/D conversion result register	ADCR		R	—	—	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	√	—	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	—	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	—	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	—	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	—	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	—	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	—	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	—	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	—	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF34H	Key return control register	KRCTL		R/W	√	√	—	00H
FFF35H	Key return flag register	KRF		R/W	—	√	—	00H
FFF37H	Key return mode register 0	KRM0		R/W	√	√	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	—	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	—	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		

Table 3 - 5 List of Special Function Registers (SFRs) (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	—	√	√	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	RXD2/ SIO21	SDR11	R/W	—	√	√	0000H
FFF4BH		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	√	—	00H
FFF51H	IICA status register 0	IICS0		R	√	√	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	√	√	—	00H
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√	—	00H
FFF68H	Timer data register 04	TDR04		R/W	—	—	√	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	—	—	√	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	—	—	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	—	—	√	0000H
FFF6FH								
FFFA0H	Clock operation mode control register	CMC		R/W	—	√	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	√	—	07H
FFFA4H	System clock control register	CKC		R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	—	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	—	00H
FFFA7H	Subsystem clock select register	CKSEL		R/W	√	√	—	00H
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00HNote 2
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	—	19H
FFFABH	Watchdog timer enable register	WDTE		R/W	—	√	—	9AH/ 1AHNote 3
FFFACH	CRC input register	CRCIN		R/W	—	√	—	00H
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	√	√	√	00H
FFFD1H		IF2H		R/W	√	√	—	00H
FFFD2H	Interrupt request flag register 3	IF3L	IF3	R/W	√	√	√	00H
FFFD3H		—		—	—	—	—	00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H		MK2H		R/W	√	√	—	FFH
FFFD6H	Interrupt mask flag register 3	MK3L	MK3	R/W	√	√	√	FFH
FFFD7H		—		—	—	—	—	FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H		PR02H		R/W	√	√	—	FFH
FFFDAH	Priority specification flag register 03	PR03L	PR03	R/W	√	√	√	FFH
FFFDDBH		—		—	—	—	—	FFH

Table 3 - 5 List of Special Function Registers (SFRs) (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	√	√	√	FFH
FFFDH		PR12H		R/W	√	√		FFH
FFFDEH	Priority specification flag register 13	PR13L	PR13	R/W	√	√	√	FFH
FFDFH		—		—	—		FFH	
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	√	√	√	00H
FFFE1H		IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	√	√	√	00H
FFFE3H		IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H		MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H		MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H		PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH		PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10	PR10L	PR10	R/W	√	√	√	FFH
FF FEDH		PR10H		R/W	√	√		FFH
FFFE EH	Priority specification flag register 11	PR11L	PR11	R/W	√	√	√	FFH
FFFE FH		PR11H		R/W	√	√		FFH
FFFF0H	Multiplication and accumulation register (L)	MACRL		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	Multiplication and accumulation register (H)	MACRH		R/W	—	—	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	—	00H

Note 1. For the reset resources, see **Section 21 Reset Function**.

Note 2. The initial value depends on the source of the reset. See **23.3.1 Voltage detection register (LVIM)**.

Note 3. The reset value of the WDTE register is determined by the setting of the option byte.

Remark For extended SFRs (2nd SFRs), see **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)**.

3.2.5 Extended special function registers (2nd SFRs)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation

Describe as follows for the 1-bit manipulation instruction operand (!addr16.bit)

When the bit name is defined: <Bit name>

When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>

- 8-bit manipulation

Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.

- 16-bit manipulation

Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3 - 6 gives lists of the extended SFRs. The meanings of items in the table are as follows.

- Symbol

This item indicates the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.

- R/W

This item indicates whether the corresponding extended SFR can be read or written.

R/W: Read/write enable

R: Read only

W: Write only

- Manipulable bit units

“√” indicates the manipulable bit unit (1, 8, or 16). “—” indicates a bit unit for which manipulation is not possible.

- After reset

Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution Do not access addresses to which extended SFRs (2nd SFRs) are not assigned.

Remark For SFRs in the SFR area, see **3.1.4 Special function register (SFR) area**.

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (1/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	—	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	—	√	—	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	—	√	—	00H
F0013H	A/D test register	ADTES	R/W	—	√	—	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	—	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	—	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	—	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	—	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	—	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	—	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	—	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	—	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	—	00H
F0044H	Port input mode register 4	PIM4	R/W	√	√	—	00H
F0047H	Port input mode register 7	PIM7	R/W	√	√	—	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	—	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	—	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	—	00H
F0057H	Port output mode register 7	POM7	R/W	√	√	—	00H
F005CH	Port output mode register 12	POM12	R/W	√	√	—	00H
F0060H	Port mode control A register 0	PMCA0	R/W	√	√	—	FFH
F0062H	Port mode control A register 2	PMCA2	R/W	√	√	—	FFH
F006CH	Port mode control A register 12	PMCA12	R/W	√	√	—	FFH
F006EH	Port mode control A register 14	PMCA14	R/W	√	√	—	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	—	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	—	00H
F0073H	Input switch control register	ISC	R/W	√	√	—	00H
F0074H	Timer input select register 0	TIS0	R/W	—	√	—	00H
F0075H	Timer input select register 1	TIS1	R/W	—	√	—	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F0079H	UART loopback select register	ULBS	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	—	√	—	Undefined Note 1
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	√	—	Undefined Note 2
F00AAH	Flash operating mode select register	FLMODE	R/W	√	√	—	40H/80H/C0H Note 3
F00ABH	Flash operating mode protect register	FLMWRP	R/W	√	√	—	00H
F00B0H	Flash security flag monitoring register	FLSEC	R	—	—	√	Undefined
F00B2H	Flash FSW monitoring register S	FLFSWS	R	—	—	√	Undefined
F00B4H	Flash FSW monitoring register E	FLFSWE	R	—	—	√	Undefined
F00B6H	Flash memory sequencer initial setting register	FSSET	R/W	—	√	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (2/10)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00B7H	Flash extra area sequencer control register	FSSE		R/W	√	√	—	00H
F00C0H	Flash protect command register	PFCMD		W	—	√	—	Undefined
F00C1H	Flash status register	PFS		R	√	√	—	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	—	00H
F00F1H	Peripheral reset control register 0	PRR0		R/W	√	√	—	00H
F00F2H	Middle-speed on-chip oscillator frequency select register	MOCODIV		R/W	—	√	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	√	√	—	Undefined
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	—	00H
F00F9H	Power-on-reset status register	PORSR		R/W	√	√	—	00H
F00FAH	Peripheral enable register 1	PER1		R/W	√	√	—	00H
F00FBH	Peripheral reset control register 1	PRR1		R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	√	—	Undefined
F00FFH	Interrupt vector jump enable register	VECTCTRL		R/W	—	√	—	00H
F0100H	Serial status register 00	SSR00L	SSR00	R	—	√	√	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	√	√	0000H
F0103H		—			—	—		
F0104H	Serial status register 02	SSR02L	SSR02	R	—	√	√	0000H
F0105H		—			—	—		
F0106H	Serial status register 03	SSR03L	SSR03	R	—	√	√	0000H
F0107H		—			—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	√	√	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	√	√	0000H
F010BH		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (3/10)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	√	√	0000H
F0139H		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	√	√	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	√	√	0000H
F0143H		—			—	—		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	√	√	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	√	√	0000H
F014BH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	√	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	√	0087H
F015BH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	√	√	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	√	√	0000H
F0175H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	√	FFFFH
F0183H								

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (4/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0184H	Timer counter register 02	TCR02	R	—	—	√	FFFFH
F0185H				—	—	√	
F0186H	Timer counter register 03	TCR03	R	—	—	√	FFFFH
F0187H				—	—	√	
F0188H	Timer counter register 04	TCR04	R	—	—	√	FFFFH
F0189H				—	—	√	
F018AH	Timer counter register 05	TCR05	R	—	—	√	FFFFH
F018BH				—	—	√	
F018CH	Timer counter register 06	TCR06	R	—	—	√	FFFFH
F018DH				—	—	√	
F018EH	Timer counter register 07	TCR07	R	—	—	√	FFFFH
F018FH				—	—	√	
F0190H	Timer mode register 00	TMR00	R/W	—	—	√	0000H
F0191H				—	—	√	
F0192H	Timer mode register 01	TMR01	R/W	—	—	√	0000H
F0193H				—	—	√	
F0194H	Timer mode register 02	TMR02	R/W	—	—	√	0000H
F0195H				—	—	√	
F0196H	Timer mode register 03	TMR03	R/W	—	—	√	0000H
F0197H				—	—	√	
F0198H	Timer mode register 04	TMR04	R/W	—	—	√	0000H
F0199H				—	—	√	
F019AH	Timer mode register 05	TMR05	R/W	—	—	√	0000H
F019BH				—	—	√	
F019CH	Timer mode register 06	TMR06	R/W	—	—	√	0000H
F019DH				—	—	√	
F019EH	Timer mode register 07	TMR07	R/W	—	—	√	0000H
F019FH				—	—	√	
F01A0H	Timer status register 00	TSR00L	R	—	√	√	0000H
F01A1H		—		—	—		
F01A2H	Timer status register 01	TSR01L	R	—	√	√	0000H
F01A3H		—		—	—		
F01A4H	Timer status register 02	TSR02L	R	—	√	√	0000H
F01A5H		—		—	—		
F01A6H	Timer status register 03	TSR03L	R	—	√	√	0000H
F01A7H		—		—	—		
F01A8H	Timer status register 04	TSR04L	R	—	√	√	0000H
F01A9H		—		—	—		
F01AAH	Timer status register 05	TSR05L	R	—	√	√	0000H
F01ABH		—		—	—		
F01ACH	Timer status register 06	TSR06L	R	—	√	√	0000H
F01ADH		—		—	—		
F01AEH	Timer status register 07	TSR07L	R	—	√	√	0000H
F01AFH		—		—	—		
F01B0H	Timer channel enable status register 0	TE0L	R	√	√	√	0000H
F01B1H		—		—	—		
F01B2H	Timer channel start register 0	TS0L	R/W	√	√	√	0000H
F01B3H		—		—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (5/10)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	√	√	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—	—		
F0212H	Middle-speed on-chip oscillator trimming register	MIOTRM		R/W	—	√	—	90H
F0213H	Low-speed on-chip oscillator trimming register	LIOTRM		R/W	—	√	—	80H
F0214H	High-speed system clock division register	MOSCDIV		R/W	—	√	—	00H
F0215H	Standby mode release setting register	WKUPMD		R/W	√	√	—	00H
F0216H	Reserved	—		—	—	—	—	00H
F0218H	LVD detection flag clearing register	LVDFCLR		R/W	√	√	—	00H
F0220H	Second count register	SEC		R/W	—	√	—	Undefined
F0221H	Minute count register	MIN		R/W	—	√	—	Undefined
F0222H	Hour count register	HOUR		R/W	—	√	—	Undefined
F0223H	Day-of-week count register	WEEK		R/W	—	√	—	Undefined
F0224H	Day count register	DAY		R/W	—	√	—	Undefined
F0225H	Month count register	MONTH		R/W	—	√	—	Undefined
F0226H	Year count register	YEAR		R/W	—	√	—	Undefined
F0227H	Time error correction register	SUBCUD		R/W	—	√	—	00H
F0228H	Alarm minute register	ALARMWM		R/W	—	√	—	Undefined
F0229H	Alarm hour register	ALARMWH		R/W	—	√	—	Undefined
F022AH	Alarm day-of-week register	ALARMWW		R/W	—	√	—	Undefined
F022BH	Realtime clock control register 0	RTCC0		R/W	√	√	—	00H
F022CH	Realtime clock control register 1	RTCC1		R/W	√	√	—	00H
F0230H	IICA control register 00	IICCTL00		R/W	√	√	—	00H
F0231H	IICA control register 01	IICCTL01		R/W	√	√	—	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	—	√	—	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	—	√	—	FFH
F0234H	Slave address register 0	SVA0		R/W	—	√	—	00H
F0240H	Event output destination select register 00	ELSELR00		R/W	—	√	—	00H
F0241H	Event output destination select register 01	ELSELR01		R/W	—	√	—	00H
F0242H	Event output destination select register 02	ELSELR02		R/W	—	√	—	00H
F0243H	Event output destination select register 03	ELSELR03		R/W	—	√	—	00H
F0244H	Event output destination select register 04	ELSELR04		R/W	—	√	—	00H
F0245H	Event output destination select register 05	ELSELR05		R/W	—	√	—	00H
F0246H	Event output destination select register 06	ELSELR06		R/W	—	√	—	00H
F0247H	Event output destination select register 07	ELSELR07		R/W	—	√	—	00H
F0248H	Event output destination select register 08	ELSELR08		R/W	—	√	—	00H
F024AH	Event output destination select register 10	ELSELR10		R/W	—	√	—	00H
F024BH	Event output destination select register 11	ELSELR11		R/W	—	√	—	00H
F024CH	Event output destination select register 12	ELSELR12		R/W	—	√	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (6/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F024DH	Event output destination select register 13	ELSELR13	R/W	—	√	—	00H
F024EH	Event output destination select register 14	ELSELR14	R/W	—	√	—	00H
F024FH	Event output destination select register 15	ELSELR15	R/W	—	√	—	00H
F0250H	Event output destination select register 16	ELSELR16	R/W	—	√	—	00H
F0251H	Event output destination select register 17	ELSELR17	R/W	—	√	—	00H
F0252H	Event output destination select register 18	ELSELR18	R/W	—	√	—	00H
F0253H	Event output destination select register 19	ELSELR19	R/W	—	√	—	00H
F0260H	Port mode control T register 0	PMCT0	R/W	√	√	—	00H
F0261H	Port mode control T register 1	PMCT1	R/W	√	√	—	00H
F0262H	Port mode control T register 2	PMCT2	R/W	√	√	—	00H
F0263H	Port mode control T register 3	PMCT3	R/W	√	√	—	00H
F0265H	Port mode control T register 5	PMCT5	R/W	√	√	—	00H
F0267H	Port mode control T register 7	PMCT7	R/W	√	√	—	00H
F026DH	Port mode control T register 13	PMCT13	R/W	√	√	—	00H
F026EH	Port mode control T register 14	PMCT14	R/W	√	√	—	00H
F02ABH	Port function output enable register 1	PFOE1	R/W	√	√	—	FFH
F02B0H	Port digital input disable register 0	PDIDIS0	R/W	√	√	—	00H
F02B1H	Port digital input disable register 1	PDIDIS1	R/W	√	√	—	00H
F02B5H	Port digital input disable register 5	PDIDIS5	R/W	√	√	—	00H
F02B7H	Port digital input disable register 7	PDIDIS7	R/W	√	√	—	00H
F02BCH	Port digital input disable register 12	PDIDIS12	R/W	√	√	—	00H
F02BDH	Port digital input disable register 13	PDIDIS13	R/W	√	√	—	00H
F02C0H	Flash programming mode control register	FLPMC	R/W	—	√	—	08H
F02C1H	Flash area selection register	FLARS	R/W	√	√	—	00H
F02C2H	Flash address pointer register L	FLAPL	R/W	—	—	√	0000H
F02C4H	Flash address pointer register H	FLAPH	R/W	—	√	—	00H
F02C5H	Flash memory sequencer control register	FSSQ	R/W	√	√	—	00H
F02C6H	Flash end address pointer register L	FLSEDL	R/W	—	—	√	0000H
F02C8H	Flash end address pointer register H	FLSEDH	R/W	—	√	—	00H
F02C9H	Flash registers initialization register	FLRST	R/W	√	√	—	00H
F02CAH	Flash memory sequencer status register L	FSASTL	R	√	√	—	00H
F02CBH	Flash memory sequencer status register H	FSASTH	R	√	√	—	00H
F02CCH	Flash write buffer register L	FLWL	R/W	—	—	√	0000H
F02CEH	Flash write buffer register H	FLWH	R/W	—	—	√	0000H
F02E0H	DTC base address register	DTCBAR	R/W	—	√	—	FDH
F02E8H	DTC activation enable register 0	DTCEN0	R/W	√	√	—	00H
F02E9H	DTC activation enable register 1	DTCEN1	R/W	√	√	—	00H
F02EAH	DTC activation enable register 2	DTCEN2	R/W	√	√	—	00H
F02EBH	DTC activation enable register 3	DTCEN3	R/W	√	√	—	00H
F02ECH	DTC activation enable register 4	DTCEN4	R/W	√	√	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	—	—	√	0000H
F02FAH	CRC data register	CRCD	R/W	—	—	√	0000H
F0300H	Transmit buffer register 0	TXBA0	R/W	—	√	—	FFH
F0301H	Receive buffer register 0	RXBA0	R	—	√	—	FFH
F0302H	Operation mode setting register 00	ASIMA00	R/W	√	√	—	01H
F0303H	Operation mode setting register 01	ASIMA01	R/W	√	√	—	1AH
F0304H	Baud rate generator control register 0	BRGCA0	R/W	—	√	—	FFH

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (7/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0305H	Status register 0	ASISA0	R	—	√	—	00H
F0306H	Status clear trigger register 0	ASCTA0	R/W	√	√	—	00H
F0310H	UARTA clock select register 0	UTA0CK	R/W	√	√	—	00H
F0360H	Interval timer compare register 00	ITLCMP000	R/W	—	√	√	FFFFH
F0361H		ITLCMP001		—	√		
F0362H	Interval timer compare register 01	ITLCMP012	R/W	—	√	√	FFFFH
F0363H		ITLCMP013		—	√		
F0364H	Interval timer capture register 00	ITLCAP00	R	—	—	√	0000H
F0365H				—	—		
F0366H	Interval timer control register	ITLCTL0	R/W	√	√	—	00H
F0367H	Interval timer clock select register 0	ITLCSEL0	R/W	—	√	—	00H
F0368H	Interval timer frequency division register 0	ITLFDIV00	R/W	—	√	—	00H
F0369H	Interval timer frequency division register 1	ITLFDIV01	R/W	—	√	—	00H
F036AH	Interval timer capture control register 0	ITLCC0	R/W	√	√	—	00H
F036BH	Interval timer status register	ITLS0	R/W	—	√	—	00H
F036CH	Interval timer match detection mask register	ITLMKF0	R/W	—	√	—	00H
F0380H	Sequencer instruction register 0	SMSI0	R/W	—	—	√	0000H
F0381H				—	—		
F0382H	Sequencer instruction register 1	SMSI1	R/W	—	—	√	0000H
F0383H				—	—		
F0384H	Sequencer instruction register 2	SMSI2	R/W	—	—	√	0000H
F0385H				—	—		
F0386H	Sequencer instruction register 3	SMSI3	R/W	—	—	√	0000H
F0387H				—	—		
F0388H	Sequencer instruction register 4	SMSI4	R/W	—	—	√	0000H
F0389H				—	—		
F038AH	Sequencer instruction register 5	SMSI5	R/W	—	—	√	0000H
F038BH				—	—		
F038CH	Sequencer instruction register 6	SMSI6	R/W	—	—	√	0000H
F038DH				—	—		
F038EH	Sequencer instruction register 7	SMSI7	R/W	—	—	√	0000H
F038FH				—	—		
F0390H	Sequencer instruction register 8	SMSI8	R/W	—	—	√	0000H
F0391H				—	—		
F0392H	Sequencer instruction register 9	SMSI9	R/W	—	—	√	0000H
F0393H				—	—		
F0394H	Sequencer instruction register 10	SMSI10	R/W	—	—	√	0000H
F0395H				—	—		
F0396H	Sequencer instruction register 11	SMSI11	R/W	—	—	√	0000H
F0397H				—	—		
F0398H	Sequencer instruction register 12	SMSI12	R/W	—	—	√	0000H
F0399H				—	—		
F039AH	Sequencer instruction register 13	SMSI13	R/W	—	—	√	0000H
F039BH				—	—		
F039CH	Sequencer instruction register 14	SMSI14	R/W	—	—	√	0000H
F039DH				—	—		
F039EH	Sequencer instruction register 15	SMSI15	R/W	—	—	√	0000H
F039FH				—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (8/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F03A0H	Sequencer instruction register 16	SMSI16	R/W	—	—	√	0000H
F03A1H				—	—	√	
F03A2H	Sequencer instruction register 17	SMSI17	R/W	—	—	√	0000H
F03A3H				—	—	√	
F03A4H	Sequencer instruction register 18	SMSI18	R/W	—	—	√	0000H
F03A5H				—	—	√	
F03A6H	Sequencer instruction register 19	SMSI19	R/W	—	—	√	0000H
F03A7H				—	—	√	
F03A8H	Sequencer instruction register 20	SMSI20	R/W	—	—	√	0000H
F03A9H				—	—	√	
F03AAH	Sequencer instruction register 21	SMSI21	R/W	—	—	√	0000H
F03ABH				—	—	√	
F03ACH	Sequencer instruction register 22	SMSI22	R/W	—	—	√	0000H
F03ADH				—	—	√	
F03AEH	Sequencer instruction register 23	SMSI23	R/W	—	—	√	0000H
F03AFH				—	—	√	
F03B0H	Sequencer instruction register 24	SMSI24	R/W	—	—	√	0000H
F03B1H				—	—	√	
F03B2H	Sequencer instruction register 25	SMSI25	R/W	—	—	√	0000H
F03B3H				—	—	√	
F03B4H	Sequencer instruction register 26	SMSI26	R/W	—	—	√	0000H
F03B5H				—	—	√	
F03B6H	Sequencer instruction register 27	SMSI27	R/W	—	—	√	0000H
F03B7H				—	—	√	
F03B8H	Sequencer instruction register 28	SMSI28	R/W	—	—	√	0000H
F03B9H				—	—	√	
F03BAH	Sequencer instruction register 29	SMSI29	R/W	—	—	√	0000H
F03BBH				—	—	√	
F03BCH	Sequencer instruction register 30	SMSI30	R/W	—	—	√	0000H
F03BDH				—	—	√	
F03BEH	Sequencer instruction register 31	SMSI31	R/W	—	—	√	0000H
F03BFH				—	—	√	
F03C0H	Sequencer general-purpose register 0	MSG0	R	—	—	√	0000H
F03C1H				—	—	√	
F03C2H	Sequencer general-purpose register 1	MSG1	R/W	—	—	√	0000H
F03C3H				—	—	√	
F03C4H	Sequencer general-purpose register 2	MSG2	R/W	—	—	√	0000H
F03C5H				—	—	√	
F03C6H	Sequencer general-purpose register 3	MSG3	R/W	—	—	√	0000H
F03C7H				—	—	√	
F03C8H	Sequencer general-purpose register 4	MSG4	R/W	—	—	√	0000H
F03C9H				—	—	√	
F03CAH	Sequencer general-purpose register 5	MSG5	R/W	—	—	√	0000H
F03CBH				—	—	√	
F03CCH	Sequencer general-purpose register 6	MSG6	R/W	—	—	√	0000H
F03CDH				—	—	√	
F03CEH	Sequencer general-purpose register 7	MSG7	R/W	—	—	√	0000H
F03CFH				—	—	√	

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (9/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F03D0H	Sequencer general-purpose register 8	MSG8	R/W	—	—	√	0000H
F03D1H				—	—	√	
F03D2H	Sequencer general-purpose register 9	MSG9	R/W	—	—	√	0000H
F03D3H				—	—	√	
F03D4H	Sequencer general-purpose register 10	MSG10	R/W	—	—	√	0000H
F03D5H				—	—	√	
F03D6H	Sequencer general-purpose register 11	MSG11	R/W	—	—	√	0000H
F03D7H				—	—	√	
F03D8H	Sequencer general-purpose register 12	MSG12	R/W	—	—	√	0000H
F03D9H				—	—	√	
F03DAH	Sequencer general-purpose register 13	MSG13	R/W	—	—	√	0000H
F03DBH				—	—	√	
F03DCH	Sequencer general-purpose register 14	MSG14	R/W	—	—	√	0000H
F03DDH				—	—	√	
F03DEH	Sequencer general-purpose register 15	MSG15	R	—	—	√	FFFFH
F03DFH				—	—	√	
F03E0H	Sequencer control register	SMSC	R/W	√	√	—	00H
F03E1H	Sequencer status register	SMSS	R	√	√	—	00H
F0480H	Interrupt vector change register 0	FLSIVC0	R/W	—	—	√	0000H
F0481H				—	—	√	
F0482H	Interrupt vector change register 1	FLSIVC1	R/W	—	—	√	000FH
F0483H				—	—	√	
F0488H	Code flash memory guard register	GFLASH0	R/W	—	—	√	0000H
F0489H				—	—	√	
F048AH	Data flash memory guard register	GFLASH1	R/W	—	—	√	0000H
F048BH				—	—	√	
F048CH	Flash security area guard register	GFLASH2	R/W	—	—	√	0000H
F048DH				—	—	√	
F048EH	Guard register of IAWCTL register	GIAWCTL	R/W	—	—	√	0000H
F048FH				—	—	√	
F0500H	CTSU control register AL	CTSUCR0	R/W	—	√	√	0000H
F0501H		CTSUCR1		—	√		
F0502H	CTSU control register AH	CTSUCR2	R/W	—	√	√	0000H
F0503H		CTSUCR3		—	√		
F0504H	CTSU control register BL	CTSUSDP	R/W	—	√	√	0000H
F0505H		CTSUSST		—	√		
F0506H	CTSU control register BH	—	R/W	—	—	√	0000H
F0507H		CTSUDCLKC		—	√		
F0508H	CTSU measurement channel register L	CTSUMCH0	R/W	—	√	√	3F3FH
F0509H		CTSUMCH1		—	√		
F050AH	CTSU measurement channel register H	CTSUMFAF	R/W	—	√	√	0000H
F050BH		—		—	—		
F050CH	CTSU channel enable control register AL	CTSUCHAC0	R/W	—	√	√	0000H
F050DH		CTSUCHAC1		—	√		
F050EH	CTSU channel enable control register AH	CTSUCHAC2	R/W	—	√	√	0000H
F050FH		CTSUCHAC3		—	√		
F0510H	CTSU channel enable control register BL	CTSUCHAC4	R/W	—	√	√	0000H
F0511H		CTSUCHAC5		—	√		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (10/10)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0512H	CTSU channel enable control register BH	CTSUCHAC6	R/W	—	√	√	0000H
F0513H		CTSUCHAC7		—	√		
F0514H	CTSU channel transmit/receive control register AL	CTSUCHTRC0	R/W	—	√	√	0000H
F0515H		CTSUCHTRC1		—	√		
F0516H	CTSU channel transmit/receive control register AH	CTSUCHTRC2	R/W	—	√	√	0000H
F0517H		CTSUCHTRC3		—	√		
F0518H	CTSU channel transmit/receive control register BL	CTSUCHTRC4	R/W	—	√	√	0000H
F0519H		CTSUCHTRC5		—	√		
F051AH	CTSU channel transmit/receive control register BH	CTSUCHTRC6	R/W	—	√	√	0000H
F051BH		CTSUCHTRC7		—	√		
F051CH	CTSU status register L	CTSUST1	R/W	—	√	√	0000H
F051DH		CTSUST		—	√		
F0520H	CTSU sensor offset register 0	CTSUSO0	R/W	—	—	√	0000H
F0521H							
F0522H	CTSU sensor offset register 1	CTSUSO1	R/W	—	—	√	0000H
F0523H							
F0524H	CTSU sensor counter register L	CTSUSC	R	—	—	√	0000H
F0525H							
F0526H	CTSU sensor counter register H	CTSUUC	R	—	—	√	0000H
F0527H							
F0528H	CTSU calibration register L	CTSUDBGR0	R/W	—	—	√	0000H
F0529H							
F052AH	CTSU calibration register H	CTSUDBGR1	R/W	—	—	√	0000H
F052BH							
F052CH	CTSU sensor unit clock control register AL	CTSUSUCLK0	R/W	—	—	√	0000H
F052DH							
F052EH	CTSU sensor unit clock control register AH	CTSUSUCLK1	R/W	—	—	√	0000H
F052FH							
F0530H	CTSU sensor unit clock control register BL	CTSUSUCLK2	R/W	—	—	√	0000H
F0531H							
F0532H	CTSU sensor unit clock control register BH	CTSUSUCLK3	R/W	—	—	√	0000H
F0533H							
F0540H	Random number seed data register	TRNGSDR	R	—	√	—	00H
F0542H	Random number command register 0	TRNGSCR0	R/W	√	√	—	00H
F0600H	CTSU trimming register AL	RTRIM	R/W	—	√	√	Undefined Note 1
F0601H		DACTRIM		—	√		
F0602H	CTSU trimming register AH	SUADJD	R/W	—	√	√	Undefined Note 1
F0603H		TRESULT4		—	√		
F0604H	CTSU trimming register BL	TRESULT0	R/W	—	√	√	Undefined Note 1
F0605H		TRESULT1		—	√		
F0606H	CTSU trimming register BH	TRESULT2	R/W	—	√	√	Undefined Note 1
F0607H		TRESULT3		—	√		

(Notes and Remark are listed on the next page.)

- Note 1.** The value after a reset is adjusted at the time of shipment.
- Note 2.** The value after a reset is a value set in FRQSEL2 to FRQSEL0 of the option byte (000C2H).
- Note 3.** The initial value of the FLMODE register is set to the value of the MODE1 and MODE0 bits updated with the set value of the CMODE1 and CMODE0 bits in the option byte at address 000C2H.
- Remark** For SFRs in the SFR area, see **Table 3 - 5 List of Special Function Registers (SFRs)**.

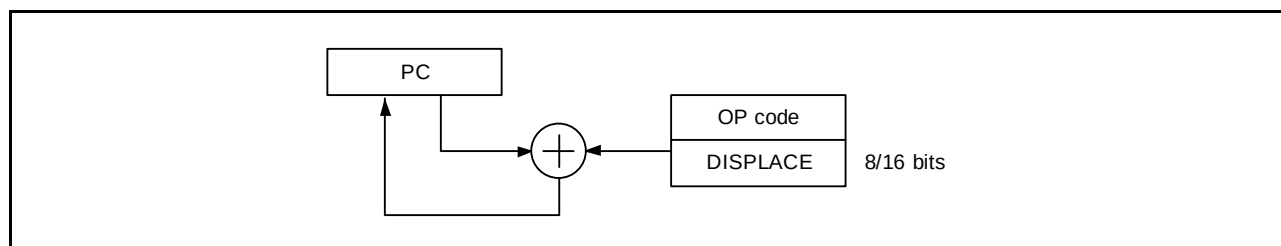
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: -128 to $+127$ or -32768 to $+32767$) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3 - 12 Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3 - 13 Example of CALL !!addr20/BR !!addr20

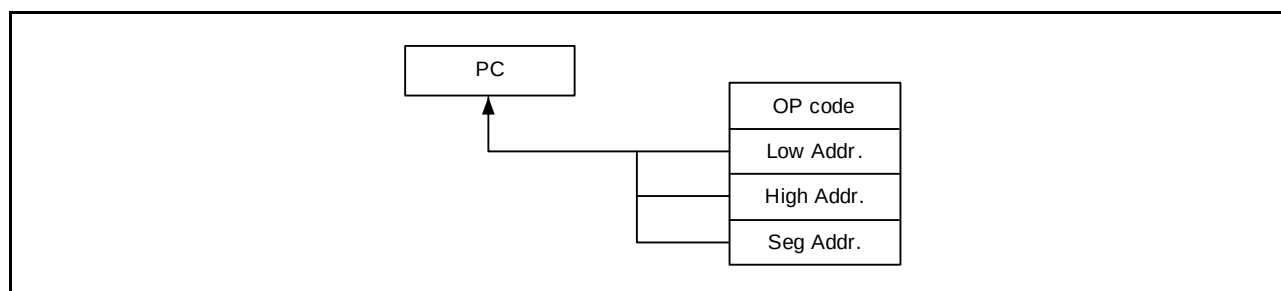
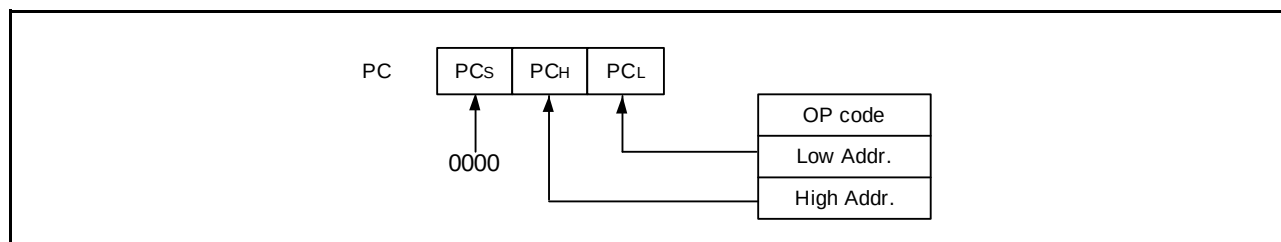


Figure 3 - 14 Example of CALL !addr16/BR !addr16



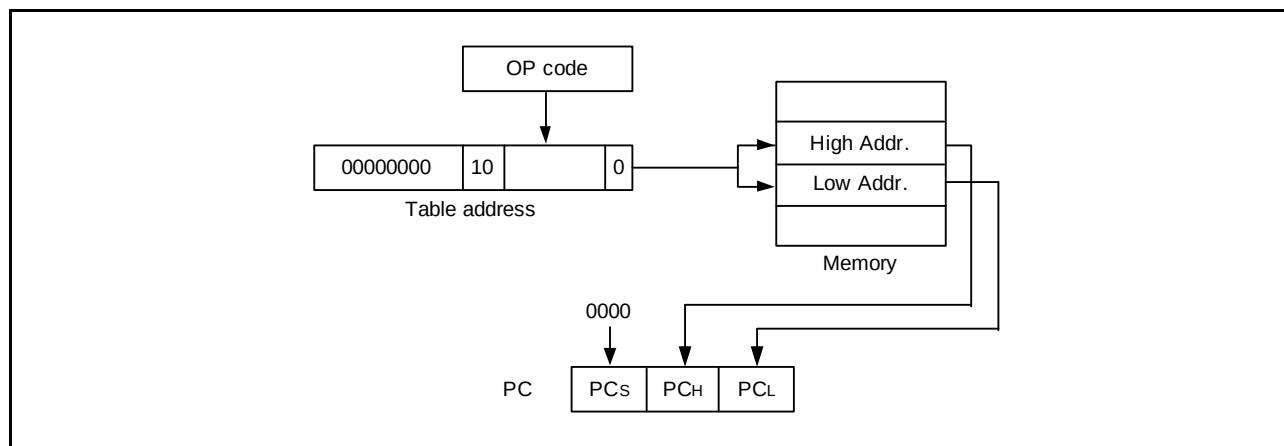
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64-Kbyte space from 00000H to 0FFFFH.

Figure 3 - 15 Outline of Table Indirect Addressing

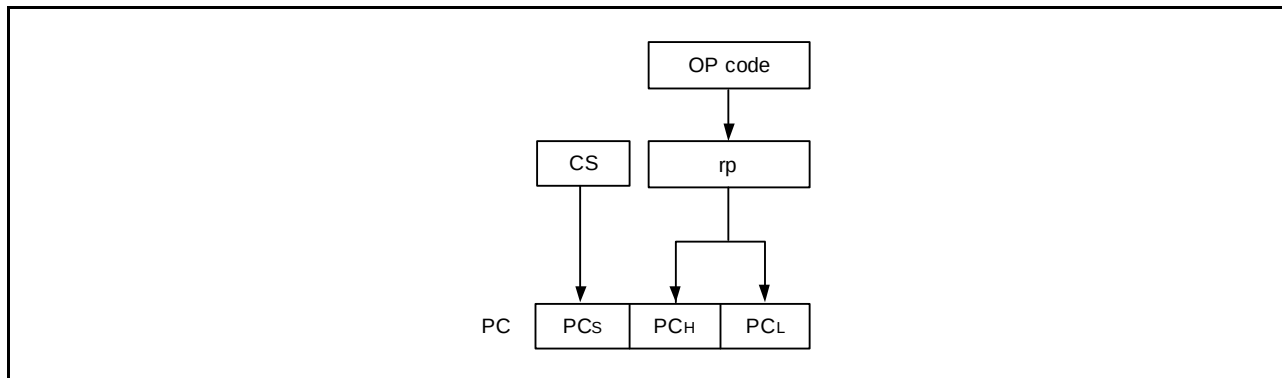


3.3.4 Register indirect addressing

[Function]

Register indirect addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3 - 16 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

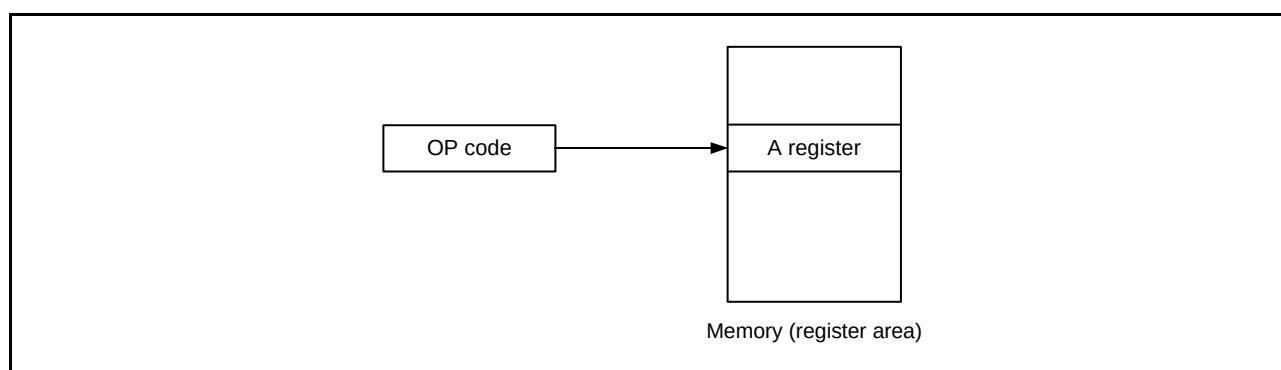
[Function]

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Implied addressing can be applied only to MULU X.

Figure 3 - 17 Outline of Implied Addressing



3.4.2 Register addressing

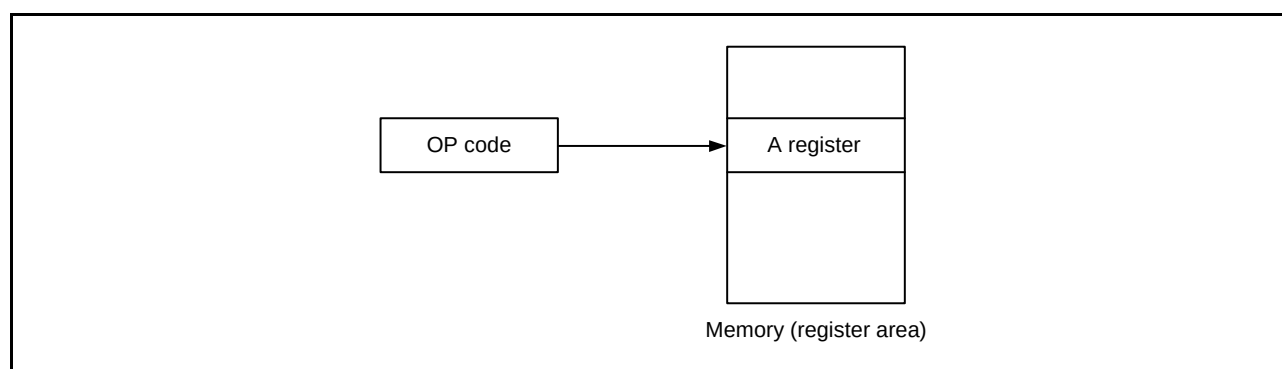
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3 - 18 Outline of Register Addressing



3.4.3 Direct addressing

[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 19 Example of !addr16

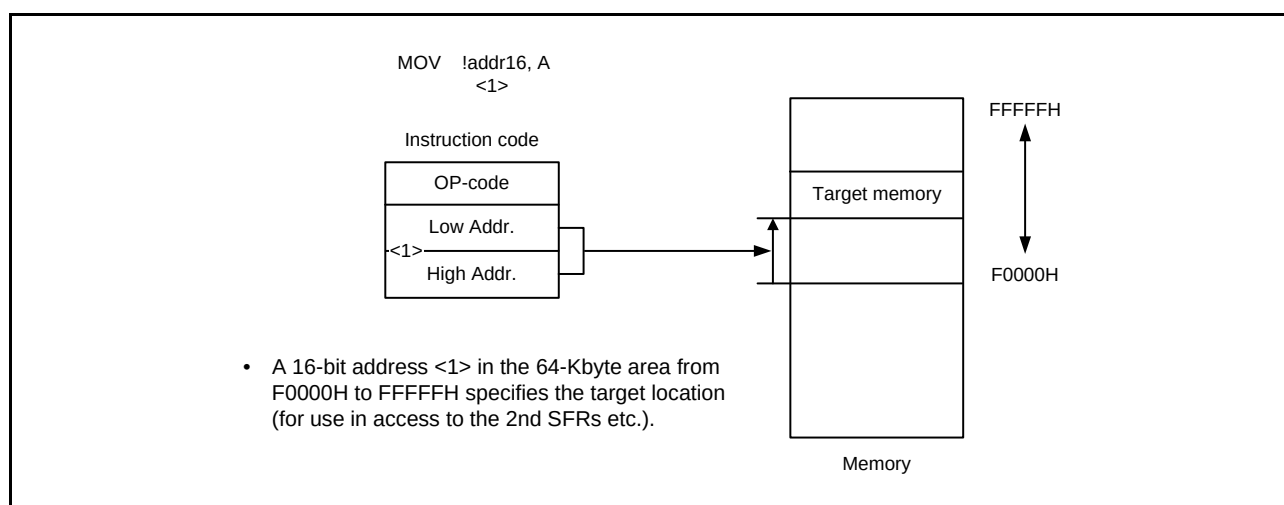
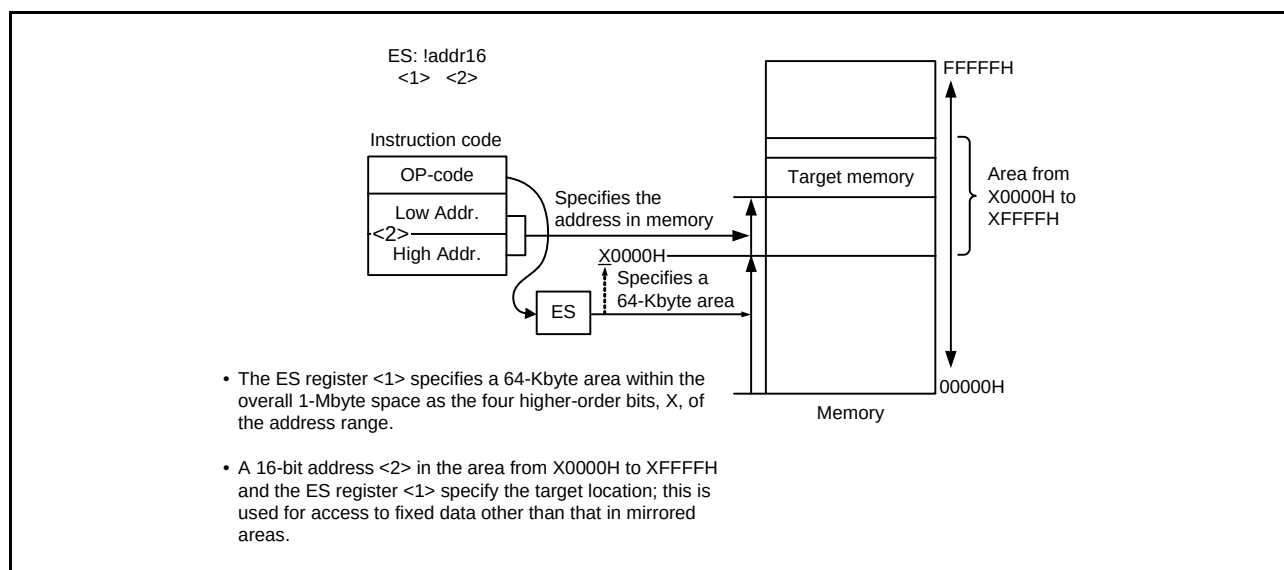


Figure 3 - 20 Example of ES:!addr16



3.4.4 Short direct addressing

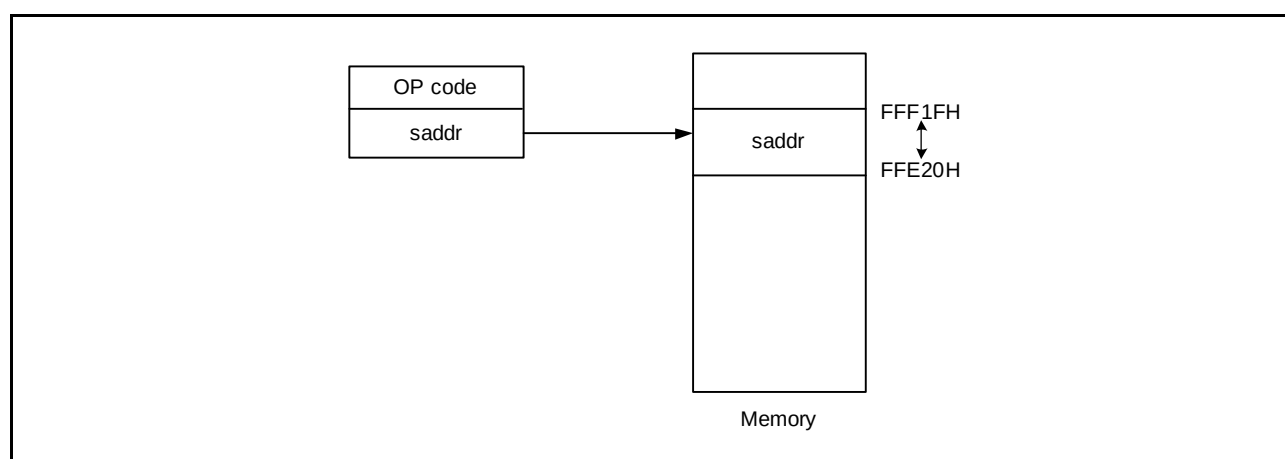
[Function]

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

[Operand format]

Identifier	Description
SADDR	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (only the space from FFE20H to FFF1FH is specifiable)
SADDRP	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3 - 21 Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data. Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

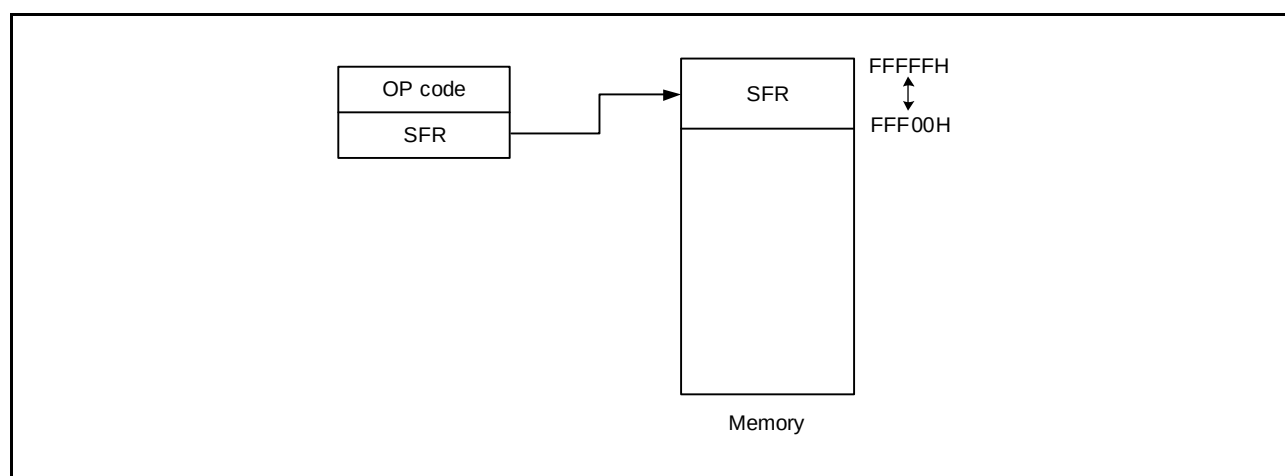
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address)

Figure 3 - 22 Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

Identifier	Description
—	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 23 Example of [DE], [HL]

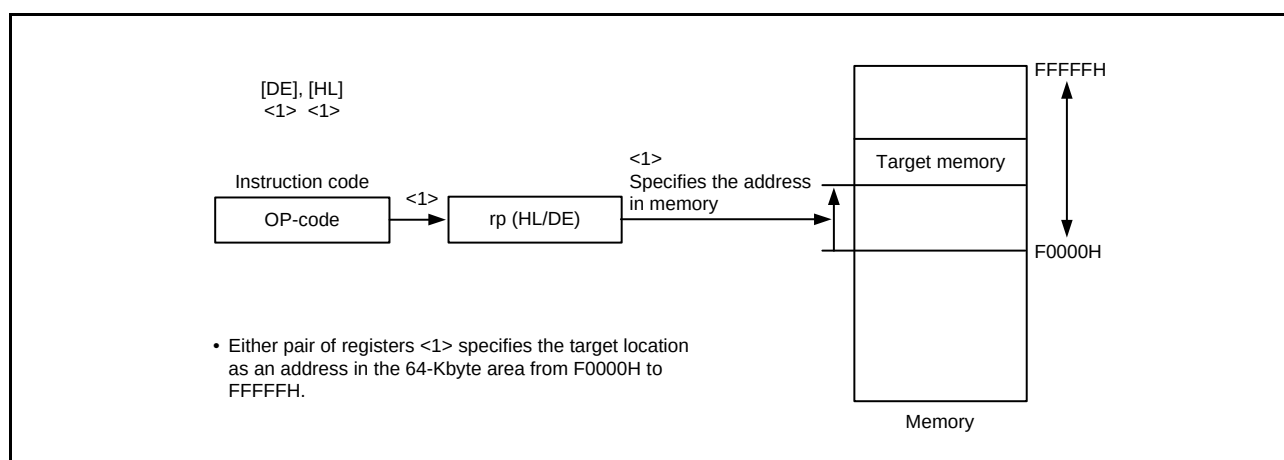
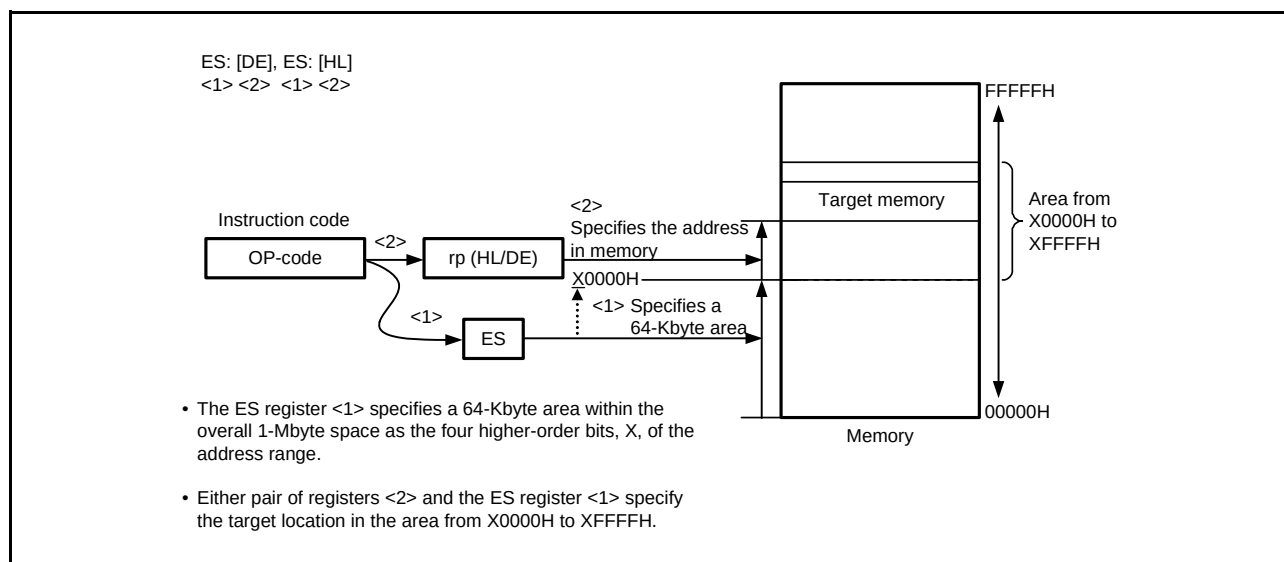


Figure 3 - 24 Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

Identifier	Description
—	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
—	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
—	word[BC] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register)
—	ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register)
—	ES:word[BC] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 25 Example of [SP + byte]

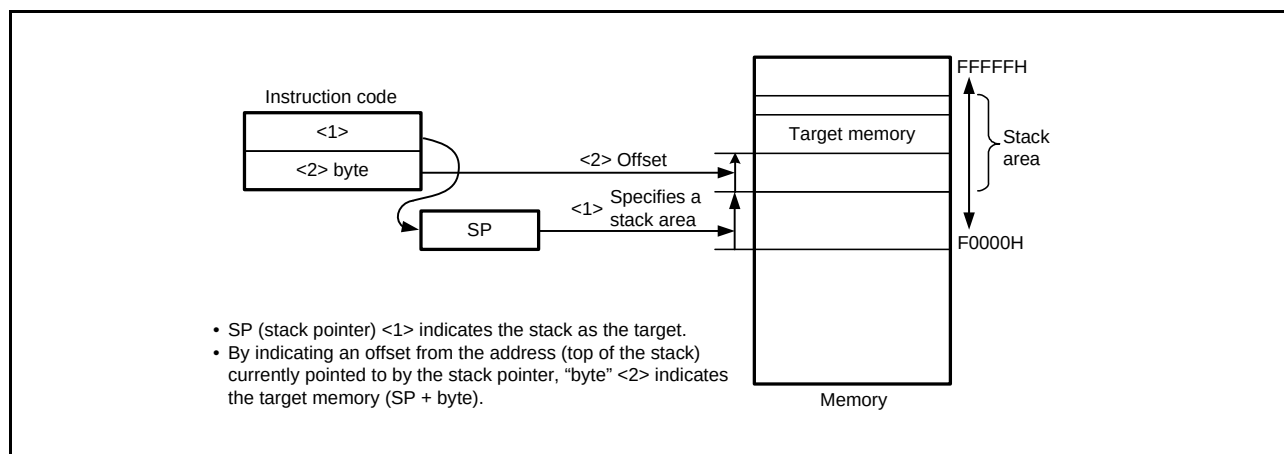


Figure 3 - 26 Example of [HL + byte], [DE + byte]

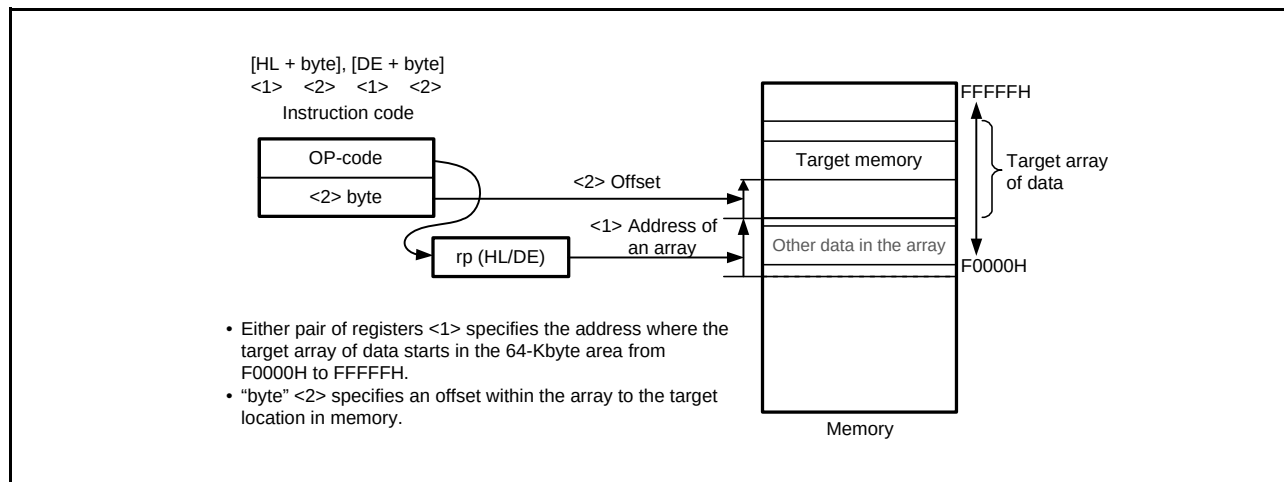


Figure 3 - 27 Example of word[B], word[C]

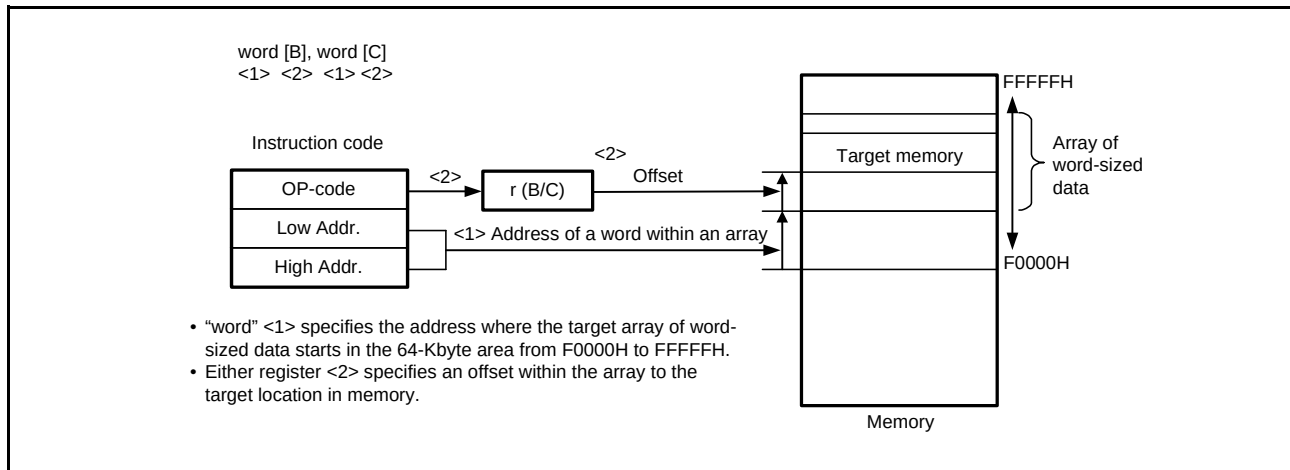


Figure 3 - 28 Example of word[BC]

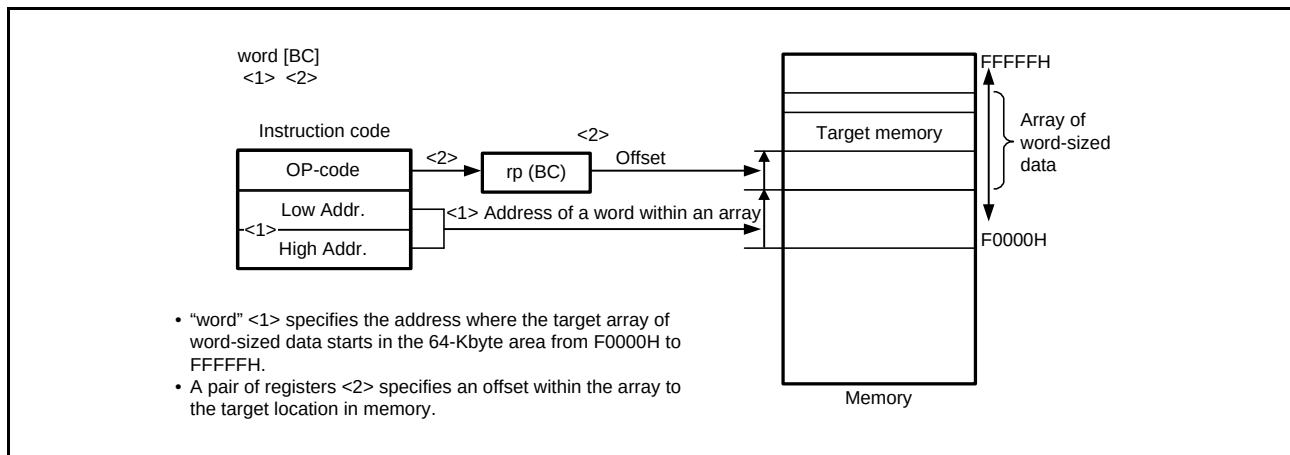


Figure 3 - 29 Example of ES:[HL + byte], ES:[DE + byte]

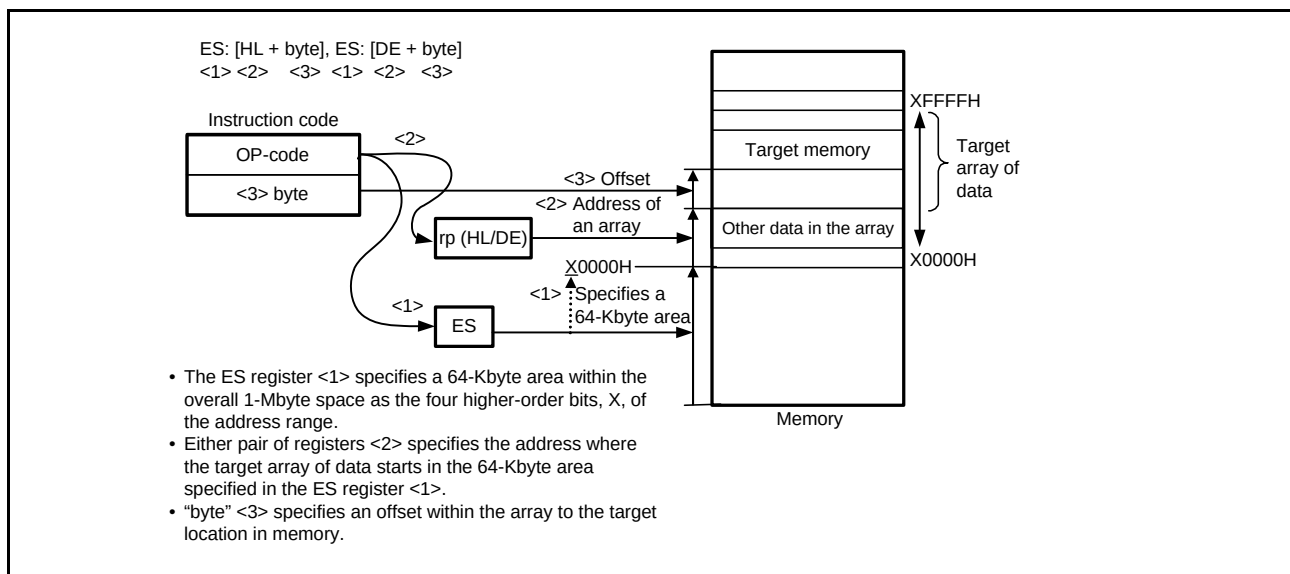


Figure 3 - 30 Example of ES:word[B], ES:word[C]

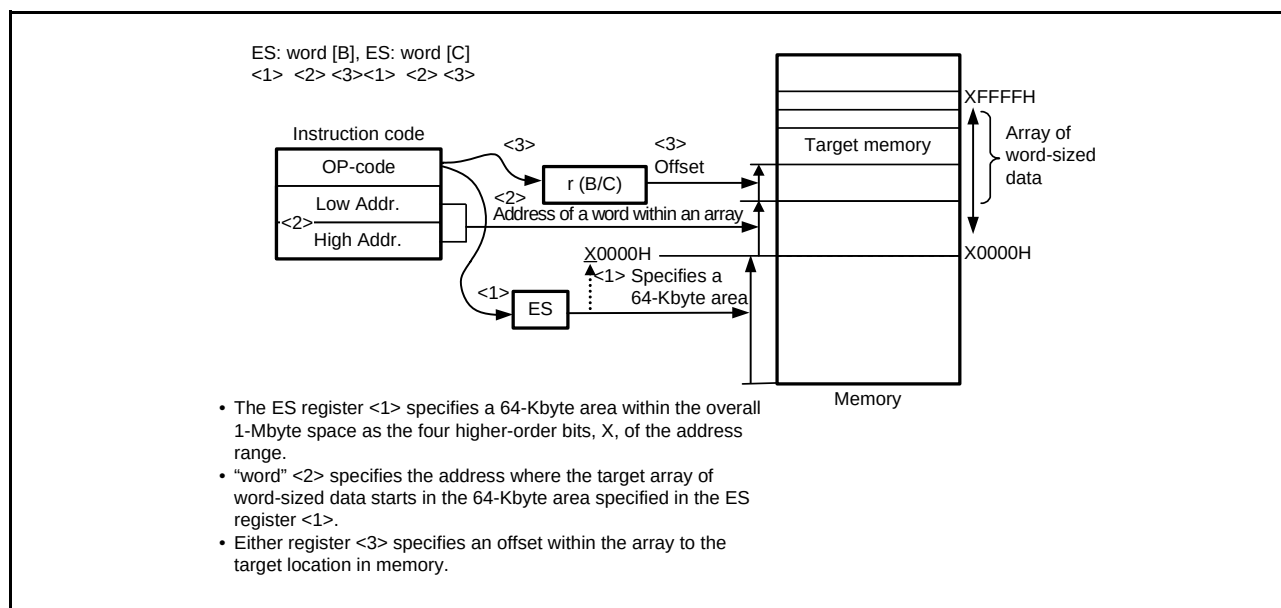
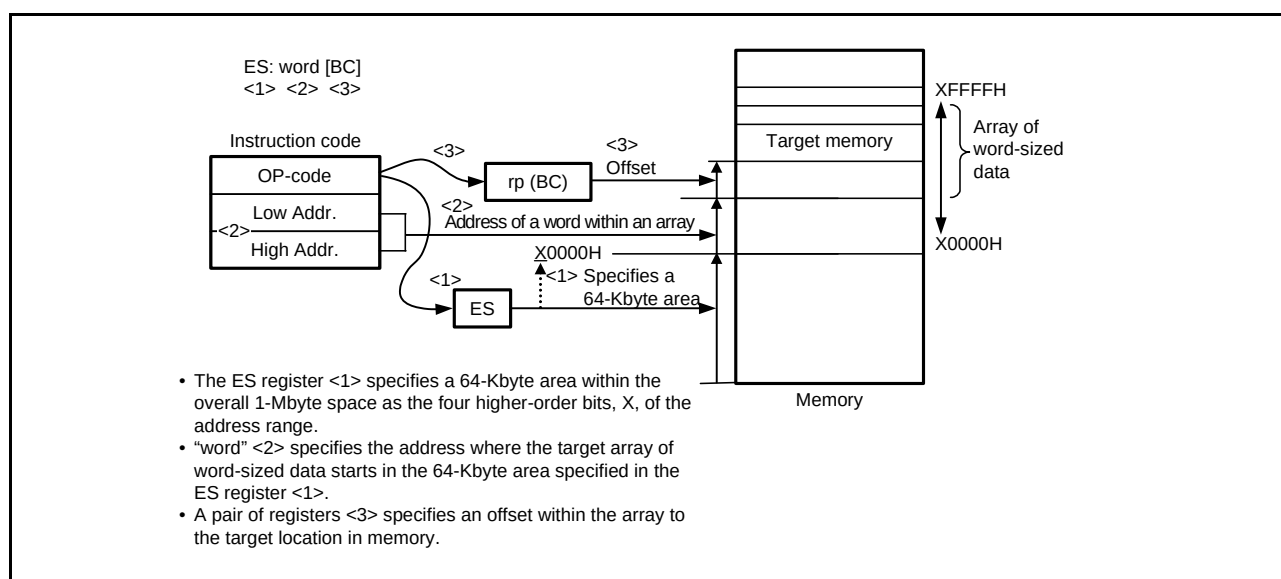


Figure 3 - 31 Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

Identifier	Description
—	[HL + B], [HL + C] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + B], ES:[HL + C] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 32 Example of [HL + B], [HL + C]

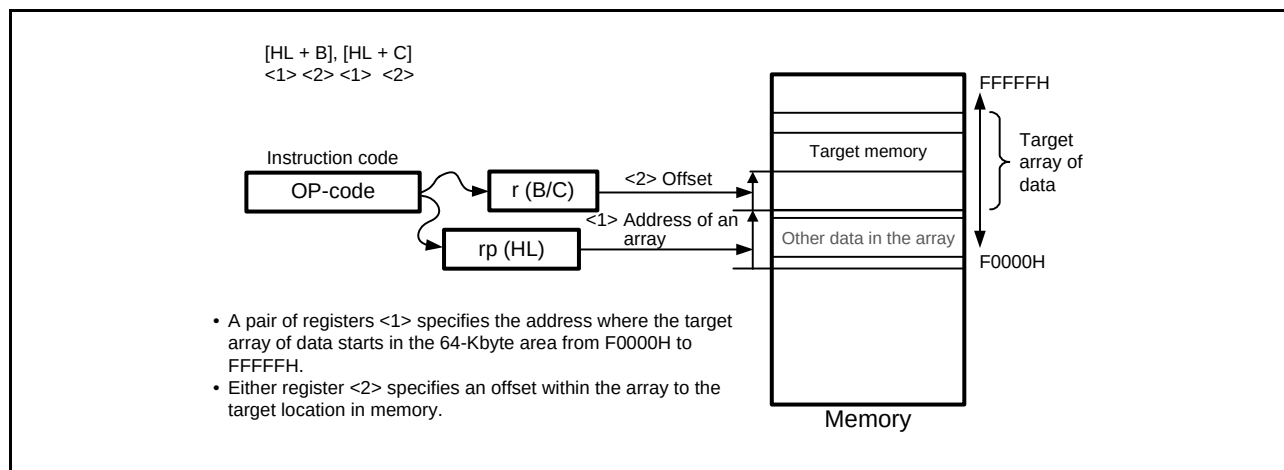
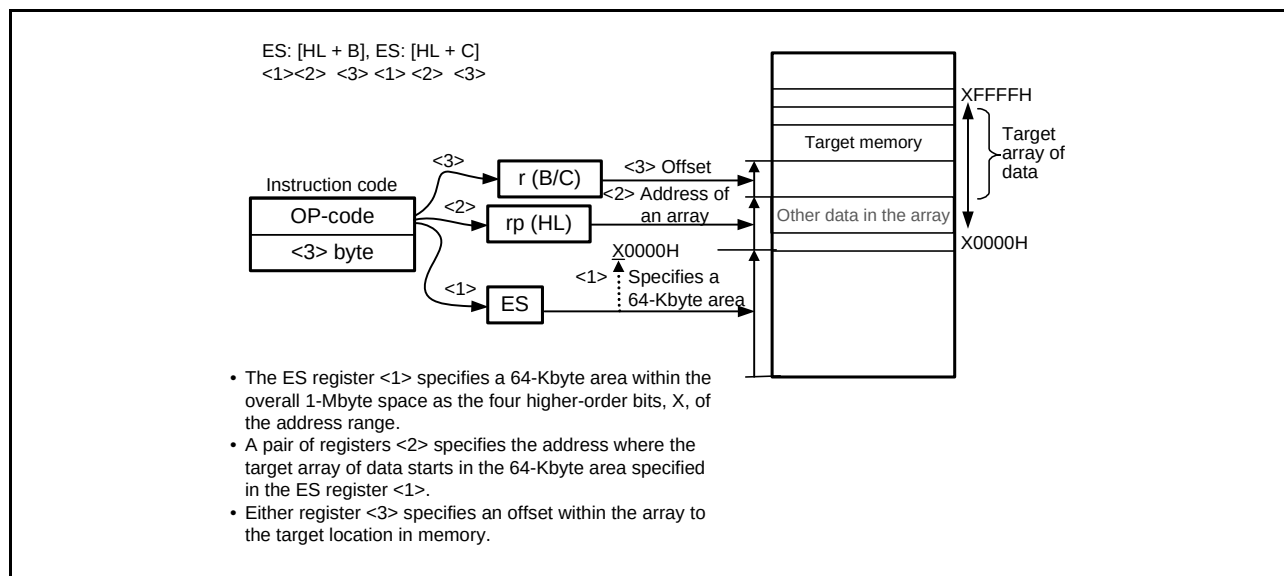


Figure 3 - 33 Example of ES:[HL + B], ES:[HL + C]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) values. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

Only the internal RAM area can be set as the stack area.

[Operand format]

Identifier	Description
—	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

The data to be saved/restored by each stack operation is shown in **Figure 3 - 34** to **Figure 3 - 39**.

Figure 3 - 34 Example of PUSH rp

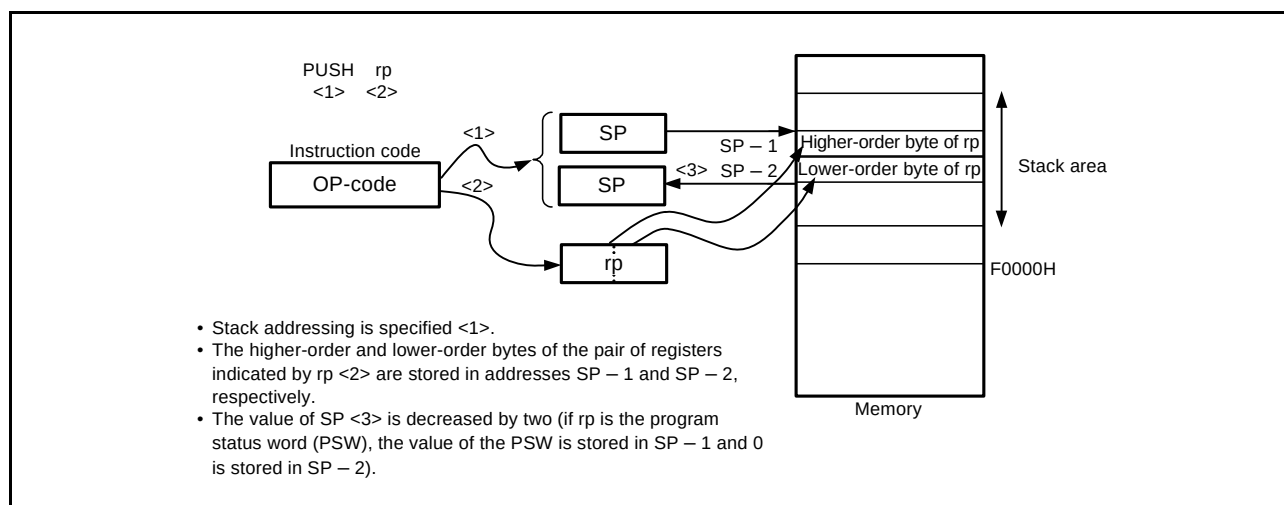


Figure 3 - 35 Example of POP

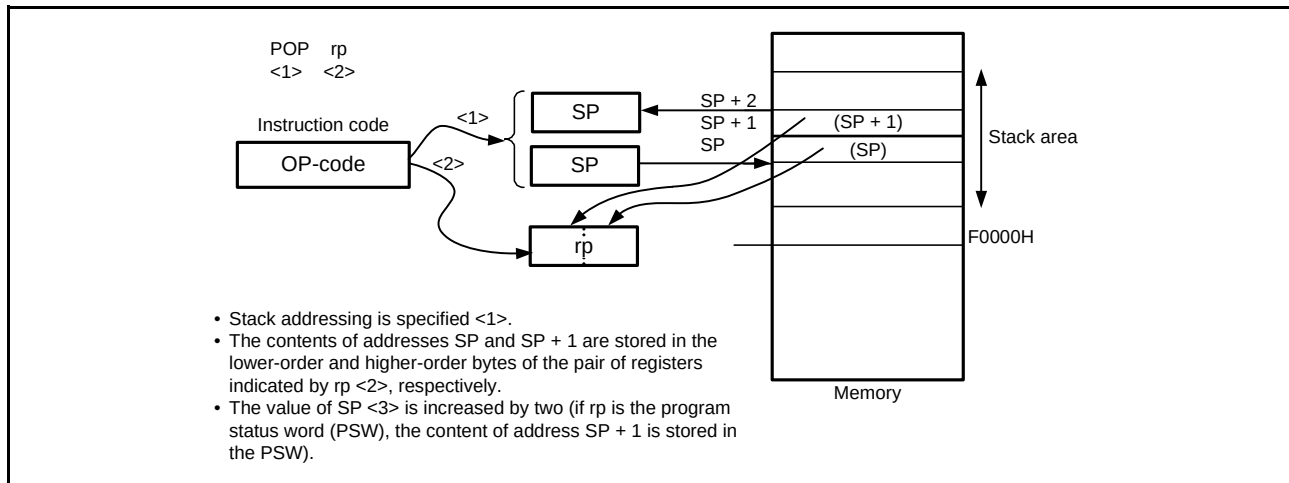


Figure 3 - 36 Example of CALL, CALLT

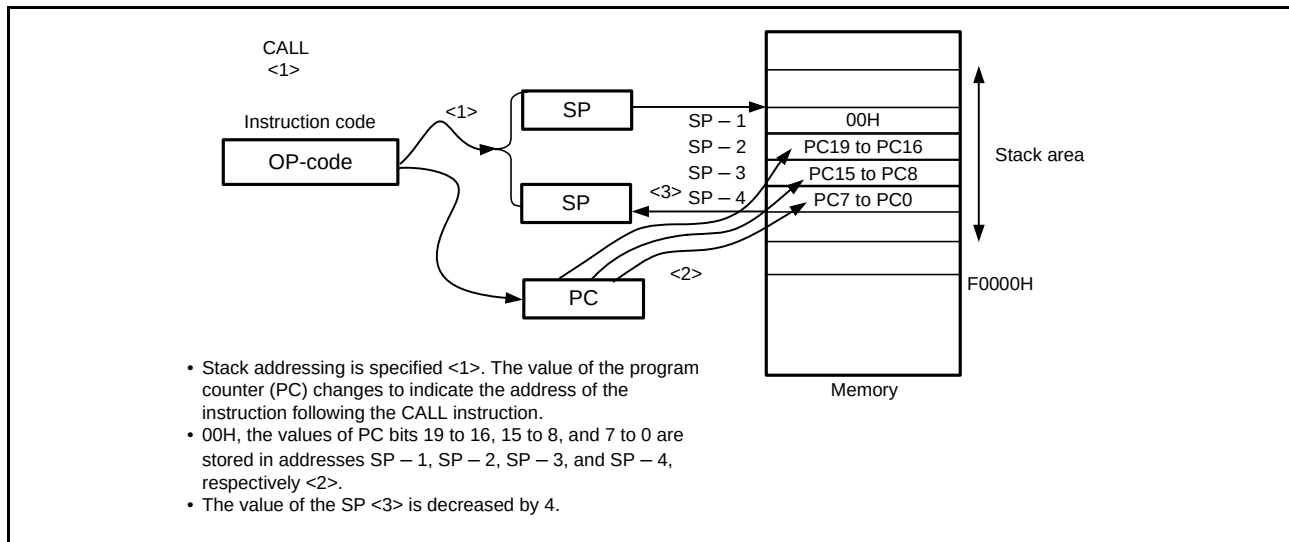


Figure 3 - 37 Example of RET

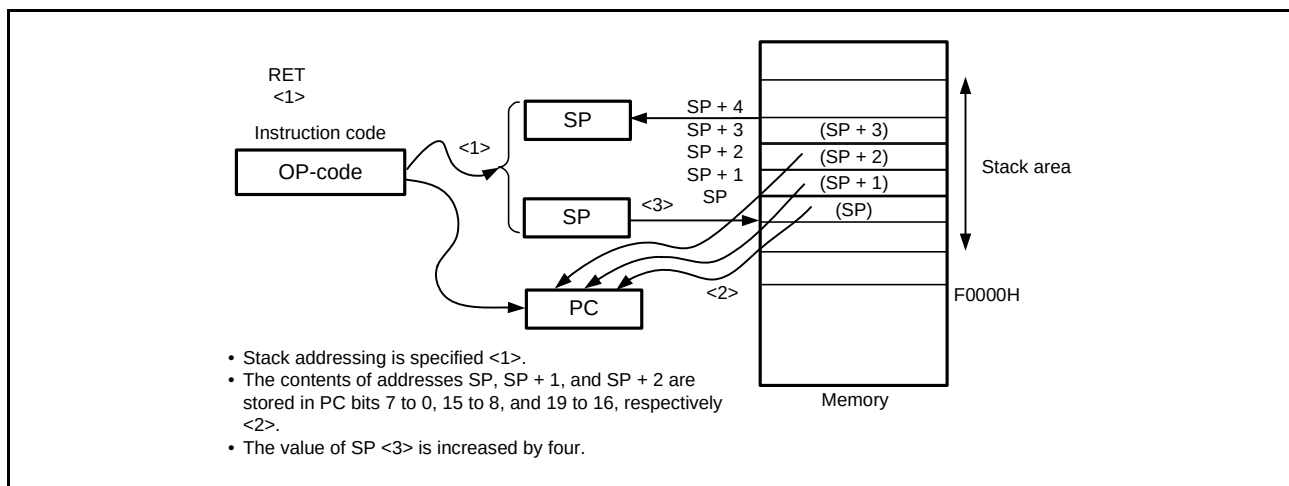


Figure 3 - 38 Example of Interrupt, BRK

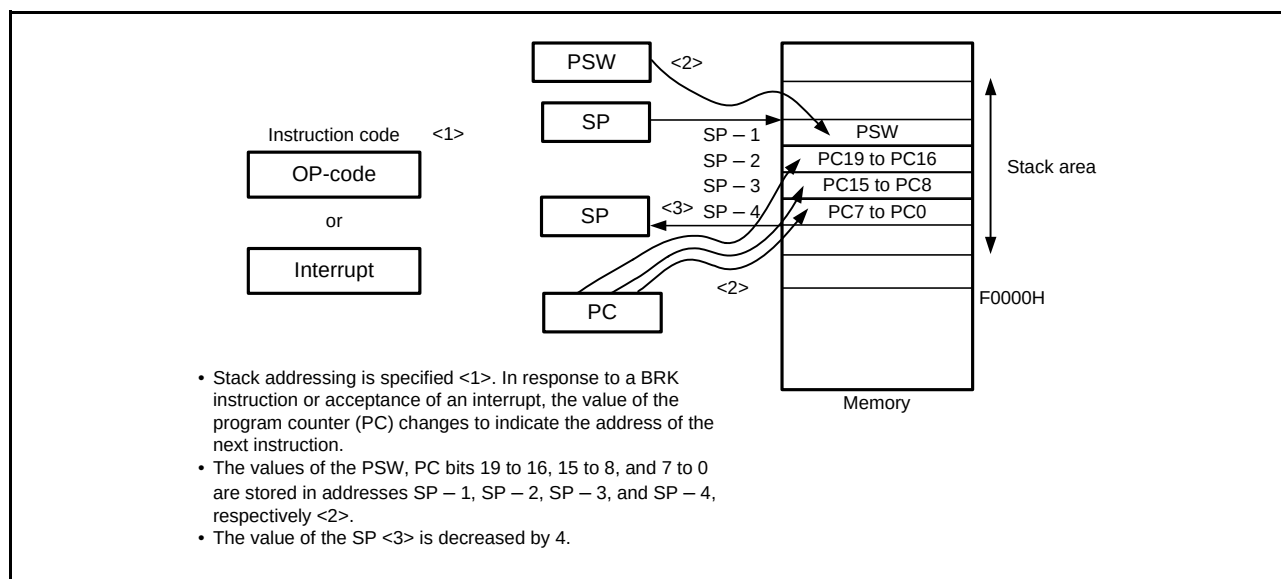
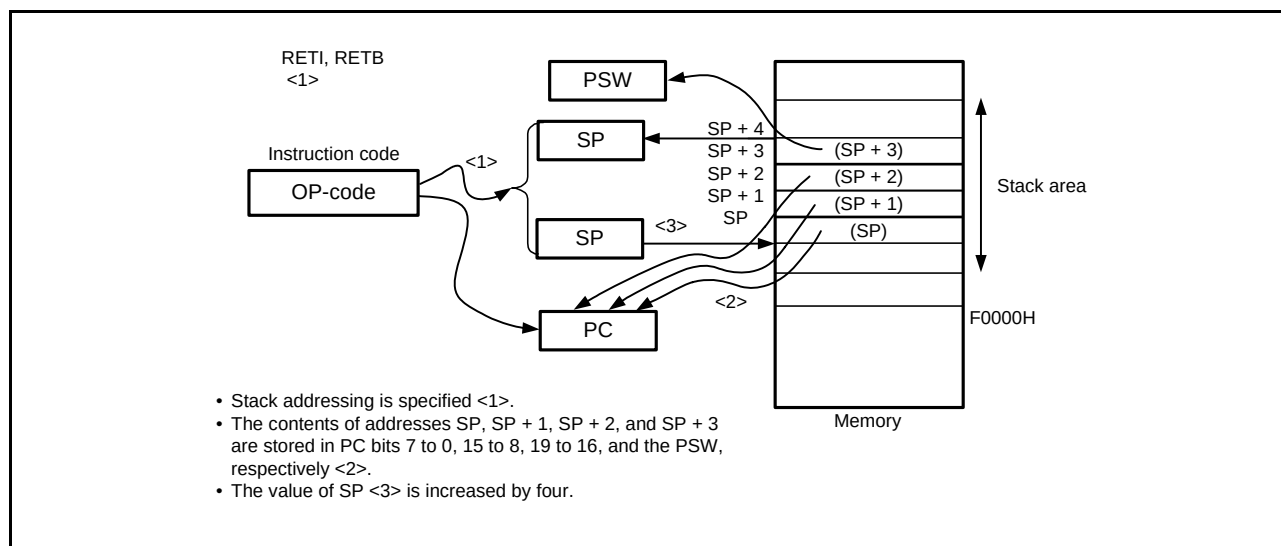


Figure 3 - 39 Example of RETI, RETB



Section 4 Port Functions

4.1 Port Functions

The RL78/G22 microcontrollers are provided with digital I/O ports, which enable variety of control operations. In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **Section 2 Pin Functions**.

4.2 Port Configuration

Ports include the following hardware.

Table 4 - 1 Port Configuration (1/2)

Item	Configuration
Control registers	Port mode registers (PM0 to PM7, PM12, PM14) Port registers (P0 to P7, P12 to P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14) Port input mode registers (PIM0, PIM1, PIM4, PIM7) Port output mode registers (POM0, POM1, POM5, POM7, POM12) Port digital input disable registers (PDIDIS0, PDIDIS1, PDIDIS5, PDIDIS7, PDIDIS12, PDIDIS13) Port mode control A registers (PMCA0, PMCA2, PMCA12, PMCA14) Port mode control T registers (PMCT0 to PMCT3, PMCT5, PMCT7, PMCT13, PMCT14) Peripheral I/O redirection register (PIOR) Port function output enable register 1 (PFOE1) Port mode select register (PMS)
Port	16-pin products Total: 12 (CMOS I/O: 11 (N-ch open drain I/O [withstand voltage of V_{DD}]: 4), CMOS input: 1) 20-pin products Total: 16 (CMOS I/O: 15 (N-ch open drain I/O [withstand voltage of V_{DD}]: 5), CMOS input: 1) 24-pin products Total: 20 (CMOS I/O: 17 (N-ch open drain I/O [withstand voltage of V_{DD}]: 6), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) 25-pin products Total: 21 (CMOS I/O: 17 (N-ch open drain I/O [withstand voltage of V_{DD}]: 6), CMOS input: 1, CMOS output: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) 30-pin products Total: 26 (CMOS I/O: 23 (N-ch open drain I/O [withstand voltage of V_{DD}]: 10), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 2) 32-pin products Total: 28 (CMOS I/O: 24 (N-ch open drain I/O [withstand voltage of V_{DD}]: 10), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 3) 36-pin products Total: 32 (CMOS I/O: 28 (N-ch open drain I/O [withstand voltage of V_{DD}]: 12), CMOS input: 1, N-ch open drain I/O [withstand voltage of 6 V]: 3) 40-pin products Total: 36 (CMOS I/O: 30 (N-ch open drain I/O [withstand voltage of V_{DD}]: 12), CMOS input: 3, N-ch open drain I/O [withstand voltage of 6 V]: 3) 44-pin products Total: 40 (CMOS I/O: 33 (N-ch open drain I/O [withstand voltage of V_{DD}]: 12), CMOS input: 3, N-ch open drain I/O [withstand voltage of 6 V]: 4) 48-pin products Total: 44 (CMOS I/O: 36 (N-ch open drain I/O [withstand voltage of V_{DD}]: 13), CMOS input: 3, CMOS output: 1, N-ch open drain I/O [withstand voltage of 6 V]: 4)

Table 4 - 1 Port Configuration (2/2)

Item	Configuration
Pull-up resistor	• 16-pin products Total: 8 • 20-pin products Total: 12 • 24-pin products Total: 14 • 25-pin products Total: 14 • 30-pin products Total: 19 • 32-pin products Total: 20 • 36-pin products Total: 22 • 40-pin products Total: 23 • 44-pin products Total: 25 • 48-pin products Total: 28

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units by port mode register 0 (PM0). When the P00 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P01 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 0 (PIM0).

Output from the P00 pin can be specified as N-ch open-drain output (withstand voltage of V_{DD}) in 1-bit units by port output mode register 0 (POM0).

This port can also be used for timer I/O, A/D converter analog input, serial interface data I/O and clock I/O, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

The settings of the port pins following a reset are as follows.

- P00 and P01 pins of the 20- to 32-pin products ... Analog input
- P00 and P01 pins of the 36- to 48-pin products ... Input mode

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units by port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10, P11, and P13 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 1 (PIM1).

Output from the P10 to P15 and P17 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}) in 1-bit units by port output mode register 1 (POM1).

This port can also be used for serial interface data I/O and clock I/O, UART data transmission and reception for external device connection when programming flash memory, clock/buzzer output, timer I/O, external interrupt request input, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 1 is set to input mode following a reset.

4.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units by port mode register 2 (PM2).

This port can also be used for A/D converter analog input, A/D converter reference voltage input (+ side and - side), and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 2 is set to analog input following a reset.

4.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units by port mode register 3 (PM3). When the P30 and P31 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, realtime clock correction clock output, clock/buzzer output, timer I/O, serial interface clock I/O, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**. Port 3 is set to input mode following a reset.

4.2.5 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units by port mode register 4 (PM4). When the P40 and P41 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

Input to the P41 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 4 (PIM4).

This port can also be used for data I/O for a flash memory programmer/debugger. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 4 is set to input mode following a reset.

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units by port mode register 5 (PM5). When the P50 and P51 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Output from the P50 pin can be specified as N-ch open-drain output (withstand voltage of V_{DD}) in 1-bit units by port output mode register 5 (POM5).

This port can also be used for external interrupt request input, serial interface data I/O, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins.

For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 5 is set to input mode following a reset.

4.2.7 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units by port mode register 6 (PM6).

Output from the P60 to P63 pins is N-ch open-drain output (withstand voltage of 6 V).

This port can also be used for serial interface data I/O and clock I/O, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 6 is set to input mode following a reset.

4.2.8 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units by port mode register 7 (PM7). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7). Input to the P71 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units by port input mode register 7 (PIM7).

Output from the P71, P72, and P74 pins can be specified as N-ch open-drain output (withstand voltage of V_{DD}) in 1-bit units by port output mode register 7 (POM7).

This port can also be used for key interrupt input, serial interface data I/O and clock I/O, external interrupt request input, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Port 7 is set to input mode following a reset.

4.2.9 Port 12

P120 to P122 are 6-bit I/O ports with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units by port mode register 12 (PM12). When this port is used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

Output from the P120 pin can be specified as N-ch open-drain output (withstand voltage of V_{DD}) by port output mode register 12 (POM12).

P123 and P124 are 2-bit input-only ports.

This port can also be used for A/D converter analog input, connection of a resonator for the main system clock, connection of a resonator for the subsystem clock, external clock input for the main system clock, external clock input for the subsystem clock, and power supply for battery backup. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

P120 is set to analog input and P121 to P124 are set to input mode following a reset.

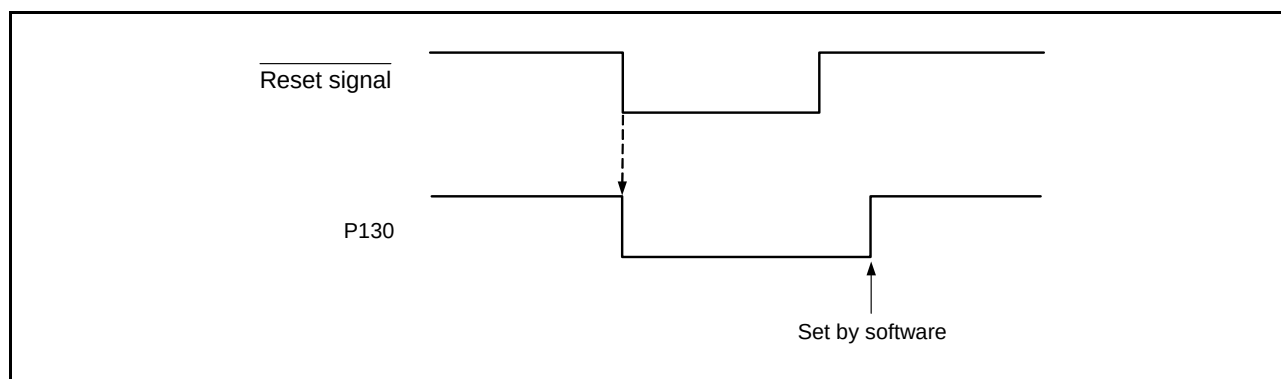
4.2.10 Port 13

P130 is a 1-bit output-only port with an output latch. P137 is a 1-bit input-only port.

P130 is fixed to output mode, and P137 is fixed to input mode.

This port can also be used for external interrupt request input and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

Remark When a reset signal is applied, P130 outputs a low-level signal. If P130 is set to a mode for outputting a high-level signal before a reset signal is applied, the P130 signal can be used to indicate a CPU reset.



4.2.11 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units by port mode register 14 (PM14). When the P140, P146, and P147 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for clock/buzzer output, external interrupt request input, A/D converter analog input, and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 4**.

P140 and P146 are set to input mode and P147 is set to analog input following a reset.

4.3 Registers for Controlling the Port Function

The following registers are used to control the port functions.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port digital input disable registers (PDIDISxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Peripheral I/O redirection register (PIOR)
- Port function output enable register 1 (PFOE1)
- Port mode select register (PMS)

Caution Which registers and bits are included depends on the product. For registers and bits implemented in each product, see Table 4 - 2. Be sure to set the bits that are not implemented to their initial values.

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, and PMCTxx Registers and the Bits Implemented in Each Product (16-Pin to 48-Pin Products with 32-Kbyte or 64-Kbyte Flash Memory) (1/3)

Port		Bit name								48 pins	44 pins	40 pins	36 pins	32 pins	30 pins	25 pins	24 pins	20 pins	16 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register										
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	PMCA00 ^{Note}	PMCT00	√	√	√	√	√	√	√	√	√	—
	1	PM01	P01	PU01	PIM01	—	—	PMCA01 ^{Note}	PMCT01	√	√	√	√	√	√	√	√	√	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PDIDIS10	—	PMCT10	√	√	√	√	√	√	√	√	√	√
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	—	PMCT11	√	√	√	√	√	√	√	√	√	√
	2	PM12	P12	PU12	—	POM12	PDIDIS12	—	PMCT12	√	√	√	√	√	√	√	√	√	√
	3	PM13	P13	PU13	PIM13	POM13	PDIDIS13	—	PMCT13	√	√	√	√	√	√	—	—	—	—
	4	PM14	P14	PU14	PIM14	POM14	PDIDIS14	—	PMCT14	√	√	√	√	√	√	—	—	—	—
	5	PM15	P15	PU15	PIM15	POM15	PDIDIS15	—	PMCT15	√	√	√	√	√	√	—	—	—	—
	6	PM16	P16	PU16	PIM16	—	—	—	PMCT16	√	√	√	√	√	√	√	√	√	—
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	—	PMCT17	√	√	√	√	√	√	√	√	√	√
Port 2	0	PM20	P20	—	—	—	—	PMCA20	—	√	√	√	√	√	√	√	√	√	√
	1	PM21	P21	—	—	—	—	PMCA21	—	√	√	√	√	√	√	√	√	√	√
	2	PM22	P22	—	—	—	—	PMCA22	PMCT22	√	√	√	√	√	√	√	√	√	√
	3	PM23	P23	—	—	—	—	PMCA23	PMCT23	√	√	√	√	√	√	—	—	—	—
	4	PM24	P24	—	—	—	—	PMCA24	PMCT24	√	√	√	√	—	—	—	—	—	—
	5	PM25	P25	—	—	—	—	PMCA25	PMCT25	√	√	√	√	—	—	—	—	—	—
	6	PM26	P26	—	—	—	—	PMCA26	PMCT26	√	√	√	—	—	—	—	—	—	—
	7	PM27	P27	—	—	—	—	PMCA27	PMCT27	√	√	—	—	—	—	—	—	—	—
Port 3	0	PM30	P30	PU30	—	—	—	—	PMCT30	√	√	√	√	√	√	√	√	√	√
	1	PM31	P31	PU31	—	—	—	—	PMCT31	√	√	√	√	√	√	√	√	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√
	1	PM41	P41	PU41	PIM41	—	—	—	—	√	√	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, and PMCTxx Registers and the Bits Implemented in Each Product (16-Pin to 48-Pin Products with 32-Kbyte or 64-Kbyte Flash Memory) (2/3)

Port		Bit name							48 pins	44 pins	40 pins	36 pins	32 pins	30 pins	25 pins	24 pins	20 pins	16 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register										
Port 5	0	PM50	P50	PU50	—	POM50	PDIDIS50	—	PMCT50	√	√	√	√	√	√	√	—	—
	1	PM51	P51	PU51	—	—	—	—	PMCT51	√	√	√	√	√	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—
	1	PM61	P61	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—
	2	PM62	P62	—	—	—	—	—	—	√	√	√	√	√	—	—	—	—
	3	PM63	P63	—	—	—	—	—	—	√	√	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	—	PMCT70	√	√	√	√	√	—	—	—	—
	1	PM71	P71	PU71	PIM71	POM71	PDIDIS71	—	PMCT71	√	√	√	√	—	—	—	—	—
	2	PM72	P72	PU72	—	POM72	PDIDIS72	—	PMCT72	√	√	√	√	—	—	—	—	—
	3	PM73	P73	PU73	—	—	—	—	PMCT73	√	√	√	—	—	—	—	—	—
	4	PM74	P74	PU74	—	POM74	PDIDIS74	—	PMCT74	√	—	—	—	—	—	—	—	—
	5	PM75	P75	PU75	—	—	—	—	PMCT75	√	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	POM120	PDIDIS120	PMCA120	—	√	√	√	√	√	√	—	—	—
	1	PM121	P121	PU121	—	—	—	—	—	√	√	√	√	√	√	√	√	√
	2	PM122	P122	PU122	—	—	—	—	—	√	√	√	√	√	√	√	√	√
	3	—	P123	—	—	—	—	—	—	√	√	√	—	—	—	—	—	—
	4	—	P124	—	—	—	—	—	—	√	√	√	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 13	0	—	P130	—	—	—	—	—	PMCT130	√	—	—	—	—	√	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS137	—	—	√	√	√	√	√	√	√	√	√

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, and PMCTxx Registers and the Bits Implemented in Each Product (16-Pin to 48-Pin Products with 32-Kbyte or 64-Kbyte Flash Memory) (3/3)

Port		Bit name								48 pins	44 pins	40 pins	36 pins	32 pins	30 pins	25 pins	24 pins	20 pins	16 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register										
Port 14	0	PM140	P140	PU140	—	—	—	—	PMCT140	√	—	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	—	PMCT146	√	√	—	—	—	—	—	—	—	—
	7	PM147	P147	PU147	—	—	—	PMCA147	PMCT147	√	√	√	√	√	√	√	√	√	—
Port 15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note This bit is only present in the 20- to 32-pin products.

4.3.1 Port mode registers (PMxx)

The PMxx registers specify input or output mode for the ports in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each PMxx register following a reset is FFH.

To use an alternate function of a port pin, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4 - 1 Format of Port Mode Registers (PMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	1	1	1	1	1	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits that are not implemented to their initial values.

4.3.2 Port registers (Pxx)

The Pxx registers set the output latch values of ports.

If data is read in the input mode, the pin level is read. If data is read in the output mode, the output latch value is read^{Note}.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each Pxx register following a reset is 00H.

Note If P00, P01, P13, P20 to P27, P120, or P147 is set up as an analog function port, when the port is read in the input mode, 0 is always returned instead of the pin level.

Caution If P00, P01, P10 to P17, P22 to P27, P30, P31, P50, P51, P70 to P75, P130, P140, P146, or P147 is set up as a capacitance measurement port, when the port is read in the input mode, 0 is always returned instead of the pin level.

Figure 4 - 2 Format of Port Registers (Pxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W ^{Note 1}
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Note 2	R/W ^{Note 1}
P14	P147	P146	0	0	0	0	0	P140	FFF0EH	00H (output latch)	R/W

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note 1. P123, P124, and P137 are read-only.

Note 2. P137: Undefined
P130: 0 (output latch)

Caution Be sure to set bits that are not implemented to their initial values.

Remark m = 0 to 7, 12 to 14; n = 0 to 7

4.3.3 Pull-up resistor option registers (PUxx)

The PUxx registers specify whether to use the on-chip pull-up resistors. On-chip pull-up resistors can be used in 1-bit units only for the bits set to both normal output mode (POMmn = 0) and input mode (PMmn = 1) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors are not connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the PU4 register following a reset is 01H, and that of the other PUxx is 00H.

Caution When data is input from a device operating at a different voltage to the TTL buffer for a port with the PIMn register, set PUMn = 0 and pull up to the power supply of the device operating at a different voltage via an external resistor.

Figure 4 - 3 Format of Pull-up Resistor Option Registers (PUxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	PU122	PU121	PU120	F003CH	00H	R/W
PU14	PU147	PU146	0	0	0	0	0	PU140	F003EH	00H	R/W

PUMn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Caution Be sure to set bits that are not implemented to their initial values.

4.3.4 Port input mode registers (PIMxx)

The PIMxx registers set the input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device operating at a different voltage.

Port input mode registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each PIMxx register following a reset is 00H.

Figure 4 - 4 Format of Port Input Mode Registers (PIMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	PIM13	0	PIM11	PIM10	F0041H	00H	R/W
PIM4	0	0	0	0	0	0	PIM41	0	F0044H	00H	R/W
PIM7	0	0	0	0	0	0	PIM71	0	F0047H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 4, 7; n = 0, 1, 3 to 7)
0	Normal input buffer
1	TTL input buffer

Caution Be sure to set bits that are not implemented to their initial values.

4.3.5 Port output mode registers (POMxx)

The POMxx registers set the output mode in 1-bit units.

N-ch open drain output (withstand voltage of V_{DD}) mode can be selected during serial communication with an external device operating at a different voltage, and for the SDA00, SDA01, SDA11, SDA20, and SDA21 pins during simplified I²C communication with an external device operating at the same voltage.

In addition, POMxx registers are used in combination with PUxx registers to specify whether to use on-chip pull-up resistors.

The POMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each POMxx register following a reset is 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (withstand voltage of V_{DD}) mode (POMmn = 1) is set.

Figure 4 - 5 Format of Port Output Mode Registers (POMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W
POM5	0	0	0	0	0	0	0	POM50	F0055H	00H	R/W
POM7	0	0	0	POM74	0	POM72	POM71	0	F0057H	00H	R/W
POM12	0	0	0	0	0	0	0	POM120	F005CH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 5, 7, 12; n = 0 to 5, 7)
0	Normal output mode
1	N-ch open-drain output (withstand voltage of V _{DD}) mode

Caution Be sure to set bits that are not implemented to their initial values.

4.3.6 Port digital input disable registers (PDIDISxx)

The PDIDISxx registers are used to prevent through-current flowing into input buffers.

When N-ch open drain output is selected for serial communications with an external device operating at a different voltage or an input port is not used, low power consumption can be achieved by setting the corresponding PDIDIS register bit to 1.

The PDIDISxx registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each PDIDISxx register following a reset is 00H.

Figure 4 - 6 Format of Port Digital Input Disable Registers (PDIDISxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PDIDIS0	0	0	0	0	0	0	0	PDIDIS 00	F02B0H	00H	R/W
PDIDIS1	PDIDIS 17	0	PDIDIS 15	PDIDIS 14	PDIDIS 13	PDIDIS 12	PDIDIS 11	PDIDIS 10	F02B1H	00H	R/W
PDIDIS5	0	0	0	0	0	0	0	PDIDIS 50	F02B5H	00H	R/W
PDIDIS7	0	0	0	PDIDIS 74	0	PDIDIS 72	PDIDIS 71	0	F02B7H	00H	R/W
PDIDIS12	0	0	0	0	0	0	0	PDIDIS 120	F02BCH	00H	R/W
PDIDIS13	PDIDIS 137	0	0	0	0	0	0	0	F02BDH	00H	R/W

PDIDISmn	Setting of input buffers (m = 0, 1, 5, 7, 12, 13; n = 0 to 5, 7)
0	Input to the input buffer is enabled (default)
1	Input to the input buffer is disabled. Through-current flowing into the input buffer is prevented.

Caution Be sure to set bits that are not implemented to their initial values.

Remark For P123 and P124, low power consumption can be achieved by setting the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC) and setting the XTSTOP bit to 1 in the clock operation status control register (CSC).

4.3.7 Port mode control A registers (PMCAxx)

The PMCAxx registers specify the digital I/O or analog input function in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each PMCAxx register following a reset is FFH.

Figure 4 - 7 Format of Port Mode Control A Registers (PMCAxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	1	1	PMCA0 1	PMCA0 0	F0060H	FFH	R/W
PMCA2	PMCA2 7	PMCA2 6	PMCA2 5	PMCA2 4	PMCA2 3	PMCA2 2	PMCA2 1	PMCA2 0	F0062H	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA1 20	F006CH	FFH	R/W
PMCA14	PMCA1 47	1	1	1	1	1	1	1	F006EH	FFH	R/W

PMCAmn	Selection of digital I/O or analog input function for Pmn pin (m = 0, 2, 12, 14; n = 0 to 7)
0	Digital I/O
1	Analog input function

Caution 1. Select input mode by using port mode register 0, 2, 12, or 14 (PM0, PM2, PM12, or PM14) for the port which is set to the analog input function by the PMCAxx register.

Caution 2. Do not set the pin that is specified as digital I/O by the PMCAxx register to the analog function by the analog input channel specification register (ADS).

Caution 3. Be sure to set bits that are not implemented to their initial values.

4.3.8 Port mode control T registers (PMCTxx)

The PMCTxx registers specify the digital I/O or capacitance measurement function in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of each PMCTxx register following a reset is 00H.

Figure 4 - 8 Format of Port Mode Control T Registers (PMCTxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCT0	0	0	0	0	0	0	PMCT0 1	PMCT0 0	F0260H	00H	R/W
PMCT1	PMCT1 7	PMCT1 6	PMCT1 5	PMCT1 4	PMCT1 3	PMCT1 2	PMCT1 1	PMCT1 0	F0261H	00H	R/W
PMCT2	PMCT2 7	PMCT2 6	PMCT2 5	PMCT2 4	PMCT2 3	PMCT2 2	0	0	F0262H	00H	R/W
PMCT3	0	0	0	0	0	0	PMCT3 1	PMCT3 0	F0263H	00H	R/W
PMCT5	0	0	0	0	0	0	PMCT5 1	PMCT5 0	F0265H	00H	R/W
PMCT7	0	0	PMCT7 5	PMCT7 4	PMCT7 3	PMCT7 2	PMCT7 1	PMCT7 0	F0267H	00H	R/W
PMCT13	0	0	0	0	0	0	0	PMCT1 30	F026DH	00H	R/W
PMCT14	PMCT1 47	PMCT1 46	0	0	0	0	0	PMCT1 40	F026EH	00H	R/W

PMCTmn	Selection of digital I/O or capacitance measurement function for Pmn pin (m = 0 to 3, 5, 7, 13, 14; n = 0 to 7)
0	Digital I/O
1	Capacitance measurement function

Caution Be sure to set bits that are not implemented to their initial values.

4.3.9 Peripheral I/O redirection register (PIOR)

The PIOR register is used to specify whether to enable or disable the peripheral I/O redirection.

This function is used to switch the assignments of multiplexed functions to port pins.

Enable a pin function that has been redirected after having used the PIOR register to assign the pin function to a port pin.

Note that the settings for redirection can only be changed before the pin function is enabled.

The PIOR register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 4 - 9 Format of Peripheral I/O Redirection Register (PIOR)

Address: F0077H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	PIOR3	PIOR2	PIOR1	PIOR0

Bit	Alternative function	48-pin		44-pin		40-, 36-, 32-, and 30-pin	
		Setting value		Setting value		Setting value	
		0	1	0	1	0	1
PIOR3	PCLBUZ0	P140	P31	These functions are not available for use. Set this bit to 0 (default value).			
PIOR2	SCLA0	P60	P14	P60	P14	P60	P14
	SDAA0	P61	P13	P61	P13	P61	P13
PIOR1	TxD2	P13	—	P13	—	P13	—
	RxD2	P14	—	P14	—	P14	—
	SCL20	P15	—	P15	—	P15	—
	SDA20	P14	—	P14	—	P14	—
	SI20	P14	—	P14	—	P14	—
	SO20	P13	—	P13	—	P13	—
	SCK20	P15	—	P15	—	P15	—
	TxD0	P12	P17	P12	P17	P12	P17
	RxD0	P11	P16	P11	P16	P11	P16
	SCL00	P10	—	P10	—	P10	—
	SDA00	P11	—	P11	—	P11	—
	SI00	P11	—	P11	—	P11	—
	SO00	P12	—	P12	—	P12	—
	SCK00	P10	—	P10	—	P10	—
PIOR0	TI02/TO02	P17	P15	P17	P15	P17	P15
	TI03/TO03	P31	P14	P31	P14	P31	P14
	TI04/TO04	—	P13	—	P13	—	P13
	TI05/TO05	—	P12	—	P12	—	P12
	TI06/TO06	—	P11	—	P11	—	P11
	TI07/TO07	P41	P10	P41	P10	—	P10

Remark —: These functions are not available for use.

4.3.10 Port function output enable register 1 (PFOE1)

The PFOE1 register is used to enable serial clock output, serial data output, and clock output for pins on which these functions are multiplexed.

The PFOE1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is FFH.

Figure 4 - 10 Format of Port Function Output Enable Register 1 (PFOE1)

Address: F02ABH

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFOE1	1	1	1	PFOE14	PFOE13	PFOE12	PFOE11	PFOE10

PFOE14	TxDA0 pin output enable	
0	Serial data output to the TxD0 pin is disabled. The TxD0 pin is available for use for the port pin or a multiplexed function.	
1	Serial data output to the TxD0 pin is enabled. If UAR0 is not in use, the TxD0 pin is available for use for the port pin or a multiplexed function.	

PFOE13	SCK01 and SCL01 pins output enable	
0	Serial clock output to the SCK01 and SCL01 pins is disabled. The SCK01 and SCL01 pins are available for use for the port pins or multiplexed functions.	
1	Serial clock output to the SCK01 and SCL01 pins is enabled. If channel 1 of SAU0 is not in use, the SCK01 and SCL01 pins are available for use for the port pins or multiplexed functions.	

PFOE12	SCK00 and SCL00 pins output enable	
0	Serial clock output to the SCK00 and SCL00 pins is disabled. The SCK00 and SCL00 pins are available for use for the port pins or multiplexed functions.	
1	Serial clock output to the SCK00 and SCL00 pins is enabled. If channel 0 of SAU0 is not in use, the SCK00 and SCL00 pins are available for use for the port pins or multiplexed functions.	

PFOE11	SO01 pin output enable	
0	Serial data output to the SO01 pin is disabled. The SO01 pin is available for use for the port pin or a multiplexed function.	
1	Serial data output to the SO01 pin is enabled. If channel 1 of SAU0 is not in use, the SO01 pin is available for use for the port pin or a multiplexed function.	

PFOE10	SO00 and TxD0 pins output enable	
0	Serial data output to the SO00 and TxD0 pins is disabled. The SO00 and TxD0 pins are available for use for the port pins or multiplexed functions.	
1	Serial data output to the SO00 and TxD0 pins is enabled. If channel 0 of SAU0 is not in use, the SO00 and TxD0 pins are available for use for the port pins or multiplexed functions.	

4.3.11 Port mode select register (PMS)

The PMS register is used to specify whether the value in the output latch for a port is read or the output level on a port pin is read when the pin is in output mode (the PMmn bit of the port mode register (PMm) is 0). For details, see

24.3.11.1 Port mode select register (PMS).

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is selected, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin. Once data is written to the output latch, it is retained until data is written to the output latch again. The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin state does not change. Therefore, byte data can be written to the ports used for both input and output. Once data is written to the output latch, it is retained until data is written to the output latch again. The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin state is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins. Once data is written to the output latch, it is retained until data is written to the output latch again. The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on the read value. The result of the operation is written to the output latch, but since the output buffer is off, the pin state does not change. Therefore, byte data can be written to the ports used for both input and output. The data of the output latch is cleared when a reset signal is generated.

4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers

The port input mode registers (PIMxx) and port output mode registers (POMxx) can be used to switch the I/O buffers to enable communications with external devices that have different operating voltages (1.8 V, 2.5 V, or 3 V) to this device. To receive input from external devices operating at different voltages, set the relevant bits of port input mode registers 0, 1, 4, and 7 (PIM0, PIM1, PIM4, and PIM7) in 1-bit units to switch from normal input (CMOS) to the TTL input buffers. To output data to external devices operating at different voltages, set the relevant bits of port output mode registers 0, 1, 5, 7, and 12 (POM0, POM1, POM5, POM7, and POM12) in 1-bit units to switch from normal output (CMOS) and N-ch open drain output (withstand voltage of V_{DD}). The following describes connection through a serial interface to an external device operating at a different voltage.

- (1) Procedure for setting input pins of UART0 to UART2, UARTA0, CSI00, CSI01, and CSI20 for use with the TTL input buffers

Use the following port pin or pins for each interface.

P11 (P16) for UART0
P01 for UART1
P14 for UART2
P71 for UARTA0
P10 and P11 for CSI00
P74 for CSI01
P14 and P15 for CSI20

Remark Functions can be assigned to the pins in parentheses via settings in the peripheral I/O redirection register (PIOR).

<1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.

<2> Set the corresponding bit of the PIM0, PIM1, and PIM7 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL} , refer to the DC characteristics when the TTL input buffer is selected.

<3> Enable the operation of the serial array unit and set the mode to the UART/simplified SPI (CSI^{Note}) mode.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

- (2) Procedure for setting output pins of UART0 to UART2, UARTA0, CSI00, CSI01, and CSI20 for use with the N-ch open-drain output mode

P12 (P17) for UART0
P00 for UART1
P13 for UART2
P72 for UARTA0
P10 and P12 for CSI00
P74 for CSI01
P13 and P15 for CSI20

Remark Functions can be assigned to the pins in parentheses via settings in the peripheral I/O redirection register (PIOR).

- <1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
- <2> The port pins are set for input (Hi-Z) after the reset state is released.
- <3> Set the corresponding bit of the PDIDIS0, PDIDIS1, and PDIDIS7 registers to 1 to disable input to the input buffer.
- <4> Set the output latch of the corresponding port to 1.
- <5> Set the corresponding bit of the POM0, POM1, and POM7 registers to 1 to set the N-ch open drain output (withstand voltage of V_{DD}) mode.
- <6> Enable the operation of the serial array unit and set the mode to the UART/simplified SPI (CSI) mode.
- <7> Set the corresponding bit of the PM0, PM1, and PM7 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

- (3) Procedure for setting I/O pins of IIC00, IIC01, and IIC20 for use in connection with a device operating at a different voltage (1.8 V, 2.5 V, or 3 V)

P10 and P11 for IIC00
P74 and P75 for IIC01
P14 and P15 for IIC20

Remark Functions can be assigned to the pins in parentheses via settings in the peripheral I/O redirection register (PIOR).

- <1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
- <2> The port pins are set for input (Hi-Z) after the reset state is released.
- <3> Set the output latch of the corresponding port to 1.
- <4> Set the corresponding bit of the POM0, POM1, and POM7 registers to 1 to set the N-ch open drain output (withstand voltage of V_{DD}) mode.
- <5> Set the corresponding bit of the PIM0, PIM1, and PIM7 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL} , refer to the DC characteristics when the TTL input buffer is selected.
- <6> Enable the operation of the serial array unit and set the mode to the simplified I²C mode.
- <7> Set the corresponding bit of the PM0, PM1, and PM7 registers to the output mode (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

4.5 Register Settings When Using Alternate Function

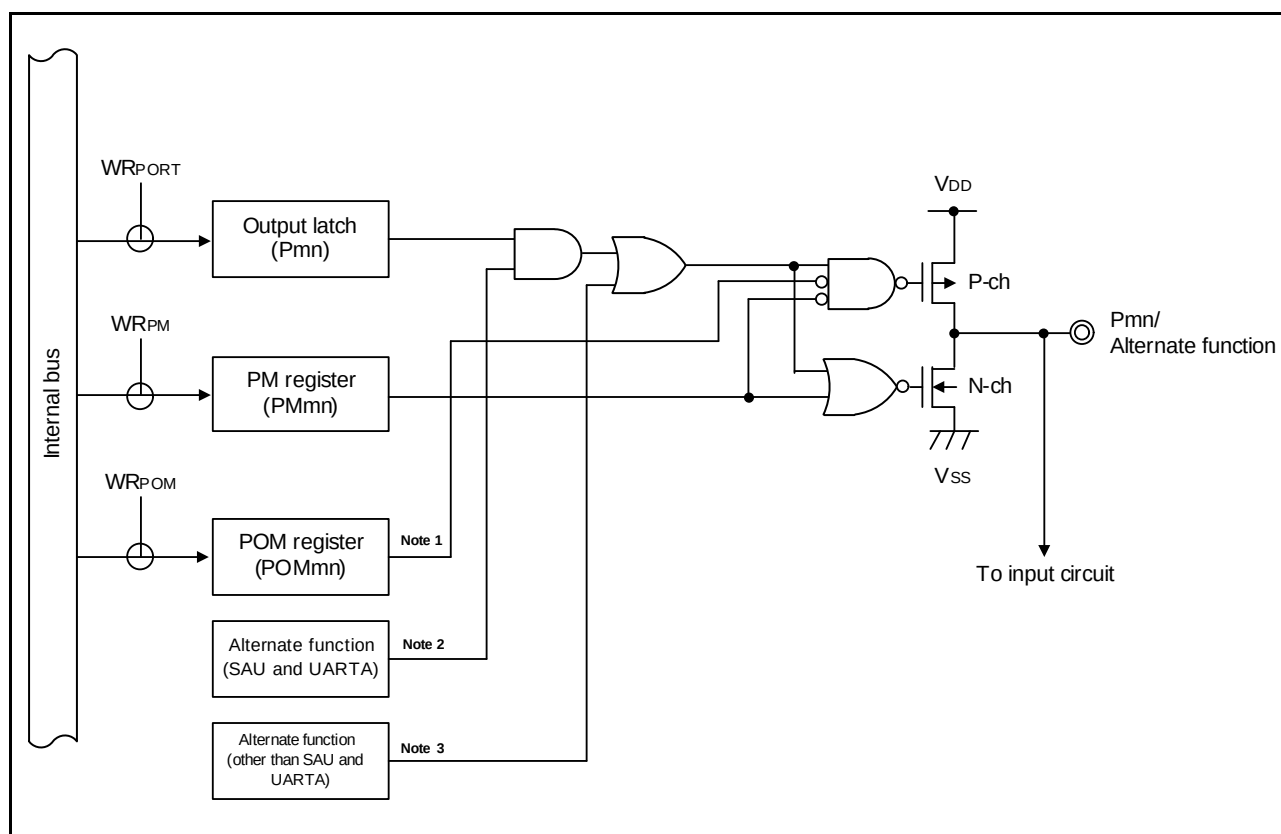
4.5.1 Basic concept when using alternate function

In the beginning, for a pin that is also assigned to the analog function, use the corresponding port mode control A register (PMCAxx) to specify whether to use the pin for the analog function or digital input/output.

For a pin that is also assigned to the capacitance measurement function, use the corresponding port mode control T register (PMCTxx) to specify whether to use the pin for the capacitance measurement function or digital input/output.

Figure 4 - 11 shows the basic configuration of the output circuit for a pin used for digital input/output. The outputs from multiplexed SAU and UARTA functions and the outputs from the output latches for the port-pin functions are input to an AND gate. The output of the AND gate is input to an OR gate. The outputs of multiplexed functions other than the SAU and UARTA (timer array unit, realtime clock, clock/buzzer output, IICA, etc.) are connected to other input pins of the OR gate. When such a pin is used for the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. A concept of basic settings for this case is shown in **Table 4 - 3**.

Figure 4 - 11 Basic Configuration of Output Circuit for a Pin



Note 1. When there is no POM register, this signal should be considered to be low level (0).

Note 2. When there is no alternate function, this signal should be considered to be high level (1).

Note 3. When there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 0 to 7, 12 to 14); n: Bit number (n = 0 to 7)

Table 4 - 3 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Port-pins Output Function	Output Functions of the SAU and UARTA	Output Functions other than those of the SAU and UARTA
Port-pins output function	—	Output is high (1)	Output is low (0)
Output functions of the SAU and UARTA	High (1)	Output is high (1)	Output is low (0)
Output functions other than those of the SAU and UARTA	Low (0)	Don't care	Output is low (0) ^{Note}

Note The output of the multiplexed functions which are not in use must be set to the low level (0) because two or more output functions other than those of the SAU and UARTA may be multiplexed on the same pin. The output of the multiplexed functions which are not in use must be set to the high level (1) because two or more output functions of the SAU and UARTA may be multiplexed on the same pin. For details on the setting method, see **4.5.2 Register settings for alternate function whose output function is not used**.

4.5.2 Register settings for alternate function whose output function is not used

When the output of an alternate function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is applicable to an alternate function, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR). This allows usage of the port function or other alternate function assigned to the target pin.

- (1) SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)

When the serial output (SOp/TxDq) is not used, such as a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOMn bit in serial output register m (SOM) to 1 (high). When SOp and TxDq pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

- (2) SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)

When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOMn and CKOMn bits in serial output register m (SOM) to 1 (high). When SCKp, SDAr, and SCLr pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

- (3) TOMn = 0 (settings when the output of channel n in TAU is not used)

When the TOMn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the corresponding bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

- (4) SDAA0 = 0, SCLA0 = 0 (setting when IICA is not used)

When IICA is not used, set the IICE0 bit in IICA control register 00 (IICCTL00) to 0 (operation stopped). This is the same setting as the initial state.

- (5) PCLBUZn = 0 (setting when clock/buzzer output is not used)

When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

- (6) TxDA0 = 1 (settings when the UARTA is not used)

When the UARTA is not used, set the UARTAEN0, TXEA0, and RXEA0 bits in operation mode setting register 0 (ASIMA00) to 0 (operation disabled). When TxDA0 pin is to be used for the port pin functions, set the PFOE14 bit corresponding to the TxDA0 pin to 1. These are the same settings as the initial state.

4.5.3 Register settings and port pin state

The correspondence between register settings and port pin state is shown in **Table 4 - 4**.

Table 4 - 4 Correspondence between Register Settings and Port Pin State

PMCAxx	PMCTxx	PMxx	Pxx	PUxx	Pin State
1	x	x	x	x	Analog input/output
0	1	x	x	x	Capacitance measurement
0	0	1	x	1	Pulled up
0	0	1	x	0	Hi-Z
0	0	0	1	x	High-level port output
0	0	0	0	x	Hi-Z
0	0	0	0	x	Low-level port output

4.5.4 Examples of register settings for port and alternate functions

Examples of register settings for port and alternate functions are shown in **Table 4 - 5**. The registers used to control the port functions should be set as shown in **Table 4 - 5**. See the following remark for legends used in **Table 4 - 5**.

Remark	—:	Not supported
	x:	Don't care
	PIOR:	Peripheral I/O redirection register
	POMxx:	Port output mode registers
	PMCAxx:	Port mode control A registers
	PMCTxx:	Port mode control T registers
	PMxx:	Port mode registers
	Pxx:	Port output latch

Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (1/15)

Pin Name	Function Used		PIORx	POMxx	PMCAXx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O							SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P00	P00	Input	—	x	0Note	0	1	x	x	—	—	√	√	√	√	√	√	√	√	√
		Output	—	0	0Note	0	0	0/1	TxD1 = 1	—										
		N-ch open drain output	—	1	0Note	0	0	0/1												
	ANI17	Analog input	—	x	1	0	1	x	x	—	—	√	√	√	√	√	—	—	—	—
	TS26	I/O	—	x	0Note	1	1	x	x	—	—	√	√	√	√	√	√	√	√	√
	TI00	Input	—	x	0Note	0	1	x	x	—	—	√	√	√	√	√	√	√	√	√
	TxD1	Output	—	0/1	0Note	0	0	1	x	—	—	√	√	√	√	√	√	√	√	√
P01	P01	Input	—	—	0Note	0	1	x	x	x	—	√	√	√	√	√	√	√	√	√
		Output	—	—	0Note	0	0	0/1	—	TO00 = 0	—									
	ANI16	Analog input	—	—	1	0	1	x	x	x	—	√	√	√	√	√	—	—	—	—
	TS27	I/O	—	—	0Note	1	1	x	x	—	—	√	√	√	√	√	√	√	√	√
	TO00	Output	—	—	0Note	0	0	0	—	x	—	√	√	√	√	√	√	√	√	√
	RxD1	Input	—	—	0Note	0	1	x	—	x	—	√	√	√	√	√	√	√	√	√

Note This setting is only applicable in the 20- to 32-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (2/15)

Pin Name	Function Used		PIORx	POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O						SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P10	P10	Input	—	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SCK00/SCL00 = 1	(TO07) = 0 ^{Note}										
		N-ch open drain output	—	1	0	0	0/1												
	TS11	I/O	—	x	1	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	SCK00	Input	PIOR1 = 0	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	PIOR1 = 0	0/1	0	0	1	x	(TO07) = 0 ^{Note}	√	√	√	√	√	√	√	√	√	√
	SCL00	Output	PIOR1 = 0	0/1	0	0	1	x	(TO07) = 0 ^{Note}	√	√	√	√	√	√	√	√	√	√
	(TI07)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO07)	Output	PIOR0 = 1	0	0	0	0	x	x	—	—	—	—	√	√	√	√	√	√
P11	P11	Input	—	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SDA00 = 1	(TO06) = 0 ^{Note}										
		N-ch open drain output	—	1	0	0	0/1												
	TS12	I/O	—	x	1	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	SI00	Input	PIOR1 = 0	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	RxD0	Input	PIOR1 = 0	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	SDA00	I/O	PIOR1 = 0	1	0	0	1	x	(TO06) = 0 ^{Note}	√	√	√	√	√	√	√	√	√	√
	(TI06)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO06)	Output	PIOR0 = 1	0	0	0	0	x	x	—	—	—	—	√	√	√	√	√	√
P12	P12	Input	—	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SO00/TxD0 = 1	(TO05) = 0 ^{Note}										
		N-ch open drain output	—	1	0	0	0/1												
	TS13	I/O	—	x	1	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	SO00	Output	PIOR1 = 0	0/1	0	0	1	x	(TO05) = 0 ^{Note}	√	√	√	√	√	√	√	√	√	√
	TxD0	Output	PIOR1 = 0	0/1	0	0	1	x	(TO05) = 0 ^{Note}	√	√	√	√	√	√	√	√	√	√
	(TI05)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO05)	Output	PIOR0 = 1	0	0	0	0	x	x	—	—	—	—	√	√	√	√	√	√

Note This setting is only applicable in the 30- to 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (3/15)

Pin Name	Function Used		PIORx	POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O						SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P13	P13	Input	—	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
		Output	—	0	0	0	0/1	Tx/D2/SO20 = 1	(TO04) = 0 (SDAA0) = 0										
		N-ch open drain output	—	1	0	0	0/1												
	TS14	I/O	—	x	1	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	TxD2	Output	PIOR1 = 0	0/1	0	0	1	x	(TO04) = 0 (SDAA0) = 0	—	—	—	—	√	√	√	√	√	√
	SO20	Output	PIOR1 = 0	0/1	0	0	1	x	(TO04) = 0 (SDAA0) = 0	—	—	—	—	√	√	√	√	√	√
	(SDAA0)	I/O	PIOR2 = 1	1	0	0	0	x	(TO04) = 0	—	—	—	—	√	√	√	√	√	√
	(TI04)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO04)	Output	PIOR0 = 1	0	0	0	0	x	(SDAA0) = 0	—	—	—	—	√	√	√	√	√	√
P14	P14	Input	—	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SDA20 = 1	(TO03) = 0 (SCLA0) = 0										
		N-ch open drain output	—	1	0	0	0/1												
	TS15	I/O	—	x	1	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	RxD2	Input	PIOR1 = 0	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	SI20	Input	PIOR1 = 0	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	SDA20	I/O	PIOR1 = 0	1	0	0	1	x	(TO03) = 0 (SCLA0) = 0	—	—	—	—	√	√	√	√	√	√
	(SCLA0)	I/O	PIOR2 = 1	1	0	0	0	x	(TO03) = 0	—	—	—	—	√	√	√	√	√	√
	(TI03)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO03)	Output	PIOR0 = 1	0	0	0	0	x	(SCLA0) = 0	—	—	—	—	√	√	√	√	√	√
P15	P15	Input	—	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SCK20/SCL20 = 1	PCLBUZ1 = 0 Note (TO02) = 0										
		N-ch open drain output	—	1	0	0	0/1												
	TS16	I/O	—	x	1	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	PCLBUZ1	Output	—	0	0	0	0	x	(TO02) = 0	—	—	—	—	√	√	√	√	√	√
	SCK20	Input	PIOR1 = 0	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
		Output	PIOR1 = 0	0/1	0	0	1	x	PCLBUZ1 = 0 Note (TO02) = 0	—	—	—	—	√	√	√	√	√	√
	SCL20	Output	PIOR1 = 0	0/1	0	0	1	x	PCLBUZ1 = 0 Note (TO02) = 0	—	—	—	—	√	√	√	√	√	√
	(TI02)	Input	PIOR0 = 1	x	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
	(TO02)	Output	PIOR0 = 1	0	0	0	0	x	PCLBUZ1 = 0 Note	—	—	—	—	√	√	√	√	√	√

Note This setting is only applicable in the 30- to 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (4/15)

Pin Name	Function Used		PIORx	POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O						SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P16	P16	Input	—	—	0	1	x	x	x	—	√	√	√	√	√	√	√	√	√
		Output	—	—	0	0	0/1	SO11 = 1 ^{Note 2}	TO01 = 0										
	TS17	I/O	—	x	1	1	x	x	x	—	√	√	√	√	√	√	√	√	√
	TI01	Input	—	—	0	1	x	x	x	—	√	√	√	√	√	√	√	√	√
	TO01	Output	—	—	0	0	0	SO11 = 1 ^{Note 2}	x	—	√	√	√	√	√	√	√	√	√
	INTP5	Input	—	—	0	1	x	x	x	—	√	√	√	√	√	√	√	√	√
	SO11	Output	—	—	0	0	1	x	TO01 = 0	—	√	—	—	—	—	—	—	—	—
	(RxD0)	Input	PIOR1 = 1	—	0	1	x	x	x	—	—	—	—	√	√	√	√	√	√
P17	P17	Input	—	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0	0/1	SDA11 = 1 ^{Note 2} SO11 = 1 ^{Note 3} (TxD0) = 1 ^{Note 1}	TO02 = 0										
		N-ch open drain output	—	1	0	0	0/1												
	TS18	I/O	—	x	1	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	TI02	Input	PIOR0 = 0	x	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	TO02	Output	PIOR0 = 0	0	0	0	0	SDA11 = 1 ^{Note 2} SO11 = 1 ^{Note 3} (TxD0) = 1 ^{Note 1}	x	√	√	√	√	√	√	√	√	√	√
	SI11	Input	—	x	0	1	x	x	x	—	√	—	—	—	—	—	—	—	—
	SDA11	I/O	—	1	0	0	1	x	TO02 = 0	√	√	—	—	—	—	—	—	—	—
	SO11	Output	—	0/1	0	0	1	x	TO02 = 0	—	—	√	√	—	—	—	—	—	—
(TxD0)	Output	PIOR1 = 1	0/1	0	0	1	x	TO02 = 0	—	—	—	—	√	√	√	√	√	√	

Note 1. This setting is only applicable in the 30- to 48-pin products.

Note 2. This setting is only applicable in the 20-pin products.

Note 3. This setting is only applicable in the 24- and 25-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (5/15)

Pin Name	Function Used		PMCAxx	PMxx	Pxx	16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O													
P20	P20	Input	0	1	x	√	√	√	√	√	√	√	√	√	√
		Output	0	0	0/1										
	ANI0	Analog input	1	1	x	√	√	√	√	√	√	√	√	√	√
	AVREFP	Reference voltage	1	1	x	√	√	√	√	√	√	√	√	√	√
P21	P21	Input	0	1	x	√	√	√	√	√	√	√	√	√	√
		Output	0	0	0/1										
	ANI1	Analog input	1	1	x	√	√	√	√	√	√	√	√	√	√
	AVREFM	Reference voltage	1	1	x	√	√	√	√	√	√	√	√	√	√
P22	P22	Input	0	1	x	√	√	√	√	√	√	√	√	√	√
		Output	0	0	0/1										
	ANI2	Analog input	1	1	x	√	√	√	√	√	√	√	√	√	√
P23	P23	Input	0	1	x	—	—	—	—	√	√	√	√	√	√
		Output	0	0	0/1										
	ANI3	Analog input	1	1	x	—	—	—	—	√	√	√	√	√	√
P24	P24	Input	0	1	x	—	—	—	—	—	√	√	√	√	√
		Output	0	0	0/1										
	ANI4	Analog input	1	1	x	—	—	—	—	—	√	√	√	√	√
P25	P25	Input	0	1	x	—	—	—	—	—	√	√	√	√	√
		Output	0	0	0/1										
	ANI5	Analog input	1	1	x	—	—	—	—	—	√	√	√	√	√
P26	P26	Input	0	1	x	—	—	—	—	—	—	√	√	√	√
		Output	0	0	0/1										
	ANI6	Analog input	1	1	x	—	—	—	—	—	—	√	√	√	√
P27	P27	Input	0	1	x	—	—	—	—	—	—	—	√	√	√
		Output	0	0	0/1										
	ANI7	Analog input	1	1	x	—	—	—	—	—	—	—	√	√	√

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (6/15)

Pin Name	Function Used		PIOR	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P30	P30	Input	—	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0/1	SCK11/SCL11 = 1	RTC1HZ = 0										
	TSCAP	I/O	—	1	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	INTP3	Input	—	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
	RTC1HZ	Output	—	0	0	0	x	x	—	—	—	—	—	—	—	√	√	√
	SCK11	Input	—	0	1	x	x	x	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0	1	x	RTC1HZ = 0	√	√	√	√	√	√	√	√	√	√
	SCL11	Output	—	0	0	1	x	RTC1HZ = 0	√	√	√	√	√	√	√	√	√	√
P31	P31	Input	—	0	1	x	—	x	—	—	√	√	√	√	√	√	√	√
		Output	—	0	0	0/1	—	TO03 = 0 PLCBUZ0 = 0 ^{Note 1} (PCLBUZ0) = 0 ^{Note 2}										
	TS01	I/O	—	1	1	x	x	x	—	—	√	√	√	√	√	√	√	√
	TI03	Input	PIOR0 = 0	0	1	x	—	x	—	—	√	√	√	√	√	√	√	√
	TO03	Output	PIOR0 = 0	0	0	0	—	PLCBUZ0 = 0 ^{Note 1} (PCLBUZ0) = 0 ^{Note 2}	—	—	√	√	√	√	√	√	√	√
	INTP4	Input	—	0	1	x	—	x	—	—	√	√	√	√	√	√	√	√
	PCLBUZ0	Output	—	0	0	0	—	TO03 = 0	—	—	√	√	√	√	√	√	√	—
	(PCLBUZ0)	Output	PIOR3 = 1	0	0	0	—	TO03 = 0	—	—	—	—	—	—	—	—	—	√

Note 1. This setting is only applicable in the 24- to 44-pin products.

Note 2. This setting is only applicable in the 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (7/15)

Pin Name	Function Used		PIOR	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O				SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P40	P40	Input	—	1	x	—	—	√	√	√	√	√	√	√	√	√	√
		Output	—	0	0/1	—	—										
P41	P41	Input	—	1	x	—	x	—	—	—	—	—	—	—	—	√	√
		Output	—	0	0/1	—	TO07 = 0										
	TI07	Input	PIOR0 = 0	1	x	—	x	—	—	—	—	—	—	—	—	√	√
	TO07	Output	PIOR0 = 0	0	0	—	x	—	—	—	—	—	—	—	—	√	√

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (8/15)

Pin Name	Function Used		POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P50	P50	Input	x	0	1	x	x	—	—	—	√	√	√	√	√	√	√	√
		Output	0	0	0	0/1	SDA11 = 1	—	—	—	—	—	—	—	—	—	—	—
		N-ch open drain output	1	0	0	0/1	—	—	—	—	—	—	—	—	—	—	—	—
	TS00	I/O	x	1	1	x	x	—	—	—	√	√	√	√	√	√	√	√
	INTP1	Input	x	0	1	x	x	—	—	—	√	√	√	√	√	√	√	√
	SI11	Input	x	0	1	x	x	—	—	—	√	√	√	√	√	√	√	√
	SDA11	I/O	1	0	0	1	x	—	—	—	√	√	√	√	√	√	√	√
P51	P51	Input	—	0	1	x	x	—	—	—	—	√	√	√	√	√	√	√
		Output	—	0	0	0/1	SO1 = 1	—	—	—	—	—	—	—	—	—	—	—
	TS28	I/O	—	1	1	x	x	—	—	—	—	√	√	√	√	√	√	√
	INTP2	Input	—	0	1	x	x	—	—	—	—	√	√	√	√	√	√	√
	SO11	Output	—	0	0	1	x	—	—	—	—	√	√	√	√	√	√	√

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (9/15)

Pin Name	Function Used		PIORx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O				SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P60	P60	Input	—	1	x	—	x	—	—	√	√	√	√	√	√	√	√
		N-ch open drain output (withstand voltage of 6 V)	—	0	0/1	—	SCLA0 = 0	—	—	—	—	—	—	—	—	—	—
	SCLA0	I/O	PIOR2 = 0	0	0	—	x	—	—	√	√	√	√	√	√	√	√
P61	P61	Input	—	1	x	—	x	—	—	√	√	√	√	√	√	√	√
		N-ch open drain output (withstand voltage of 6 V)	—	0	0/1	—	SDAA0 = 0	—	—	—	—	—	—	—	—	—	—
	SDAA0	I/O	PIOR2 = 0	0	0	—	x	—	—	√	√	√	√	√	√	√	√
P62	P62	Input	—	1	x	—	x	—	—	—	—	√	√	√	√	√	√
		N-ch open drain output (withstand voltage of 6 V)	—	0	0/1	—	SCLA1 = 0 ^{Note}	—	—	—	—	—	—	—	—	—	—
P63	P63	Input	—	1	x	—	x	—	—	—	—	—	—	—	—	√	√

Note This setting is only applicable in the 44- and 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (10/15)

Pin Name	Function Used		POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P70	P70	Input	—	0	1	x	x	—	—	—	—	—	—	√	√	√	√	√
		Output	—	0	0	0/1	SCK21/SCL21 = 1 Note 1	—	—	—	—	—	—	—	—	—	—	—
	TS02	I/O	—	1	1	x	x	—	—	—	—	—	—	√	√	√	√	√
	KR0	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	√	√	√
	SCK21	Input	—	0	1	x	x	—	—	—	—	—	—	—	√	√	√	√
		Output	—	0	0	1	x	—	—	—	—	—	—	—	√	√	√	√
	SCL21	Output	—	0	0	1	x	—	—	—	—	—	—	—	√	√	√	√
P71	P71	Input	—	0	1	x	x	—	—	—	—	—	—	—	√	√	√	√
		Output	0	0	0	0/1	SDA21 = 1	—	—	—	—	—	—	—	—	—	—	—
		N-ch open drain output	1	0	0	0/1		—	—	—	—	—	—	—	—	—	—	—
	TS03	I/O	—	1	1	x	x	—	—	—	—	—	—	—	√	√	√	√
	KR1	Input	x	0	1	x	x	—	—	—	—	—	—	—	—	√	√	√
	SI21	Input	x	0	1	x	x	—	—	—	—	—	—	—	√	√	√	√
	SDA21	I/O	1	0	0	1	x	—	—	—	—	—	—	—	√	√	√	√
	RxDA0	Input	—	0	1	x	x	—	—	—	—	—	—	—	√	√	√	√
P72	P72	Input	—	0	1	x	x	—	—	—	—	—	—	—	√	√	√	√
		Output	—	0	0	0/1	SO21 = 1 TxDA0 = 1	—	—	—	—	—	—	—	—	—	—	—
		N-ch open drain output	1	0	0	0/1		—	—	—	—	—	—	—	—	—	—	—
	TS04	I/O	—	1	1	x	x	—	—	—	—	—	—	—	√	√	√	√
	KR2	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	√	√	√
	SO21	Output	—	0	0	1	TxDA0 = 1	—	—	—	—	—	—	—	√	√	√	√
	TxDA0	Output	x	0	0	1	SO21 = 1	—	—	—	—	—	—	—	√	√	√	√
P73	P73	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	√	√	√
		Output	—	0	0	0/1	SO01 = 1 ^{Note 2}	—	—	—	—	—	—	—	—	—	—	—
	TS05	I/O	—	1	1	x	x	—	—	—	—	—	—	—	—	√	√	√
	KR3	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	√	√	√
	SO01	Output	—	0	0	1	x	—	—	—	—	—	—	—	—	—	—	—

Note 1. This setting is only applicable in the 36- to 48-pin products.

Note 2. This setting is only applicable in the 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (11/15)

Pin Name	Function Used		POMxx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P74	P74	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
		Output	0	0	0	0/1	SDA01 = 1 ^{Note}	—	—	—	—	—	—	—	—	—	—	√
		N-ch open drain output	1	0	0	0/1		—	—	—	—	—	—	—	—	—	—	√
	TS06	I/O	—	1	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	KR4	Input	x	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	INTP8	Input	x	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	SI01	Input	x	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
P75	P75	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
		Output	—	0	0	0/1	SCK01/SCL01 = 1	—	—	—	—	—	—	—	—	—	—	√
	TS07	I/O	—	1	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	KR5	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	INTP9	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
	SCK01	Input	—	0	1	x	x	—	—	—	—	—	—	—	—	—	—	√
		Output	—	0	0	1	x	—	—	—	—	—	—	—	—	—	—	√
	SCL01	Output	—	0	0	1	x	—	—	—	—	—	—	—	—	—	—	√

Note This setting is only applicable in the 36- to 48-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (12/15)

Pin Name	Function Used		POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P120	P120	Input	x	0	1	x	—	—	—	—	—	—	√	√	√	√	√	√
		Output	0	0	0	0/1	—	—	—	—	—	—	—	—	—	—	—	—
		N-ch open drain output	1	0	0	0/1		—	—	—	—	—	—	—	—	—	—	—
	ANI19	Analog input	x	1	1	x	x	—	—	—	—	—	√	√	√	√	√	√

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (13/15)

Pin Name	Function Used		CMC		PMxx	Pxx	16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O	EXCLK, OSCSEL, EXCLKS, OSCSELS	XTSEL												
P121	P121	Input	00xx/10xx/11xx	0	1	x	√	√	√	√	√	√	√	√	√	√
			xx00/xx10/xx11	1Note												
		Output	00xx/10xx/11xx	0	0	0/1										
			xx00/xx10/xx11	1Note												
	VBAT	Input	00xx/10xx/11xx	0	0	1	—	—	—	—	—	—	—	√	√	√
	X1	—	01xx	0	1	x	√	√	√	√	√	√	√	—	—	—
	XT1	—	xx01	1	1	x	√	√	√	√	√	√	√	√	√	√
P122	P122	Input	00xx/10xx	0	1	x	√	√	√	√	√	√	√	√	√	√
			xx00/xx10	1Note												
		Output	00xx/10xx	0	0	0/1										
			xx00/xx10	1Note												
	X2	—	01xx	0	1	x	√	√	√	√	√	√	√	√	√	
	XT2	—	xx01	1	1	x	√	√	√	√	√	√	√	—	—	—
	EXCLK	Input	11xx	0	1	x	√	√	√	√	√	√	√	√	√	√
	EXCLKS	Input	xx11	1	1	x	√	√	√	√	√	√	√	√	√	√
P123	P123	Input	xx00/xx10/xx11	0	—	x	—	—	—	—	—	—	—	√	√	√
	XT1	—	xx01	0	—	x	—	—	—	—	—	—	—	√	√	√
P124	P124	Input	xx00/xx10	0	—	x	—	—	—	—	—	—	—	√	√	√
	XT2	—	xx01	0	—	x	—	—	—	—	—	—	—	√	√	√
	EXCLKS	Input	xx11	0	—	x	—	—	—	—	—	—	—	√	√	√

Note This setting is only applicable in the 16- to 36-pin products.

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (14/15)

Pin Name	Function Used		PMCTxx	PMxx	Pxx	16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O													
P130	P130	Output	0	—	0/1	—	—	—	√	—	—	—	—	—	√
	TS19	I/O	1	1	x	—	—	—	√	—	—	—	—	—	√
P137	P137	Input	0	—	x	√	√	√	√	√	√	√	√	√	√
	INTPO	Input	0	—	x	√	√	√	√	√	√	√	√	√	√

Table 4 - 5 Examples of Register and Output Latch Settings for Alternate Functions (15/15)

Pin Name	Function Used		PIORx	PMCAXx	PMCTxx	PMxx	Pxx	Alternate Function Output		16-pin	20-pin	24-pin	25-pin	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin
	Function Name	I/O						SAU and UARTA (excluding clock output from the UARTA)	Other than SAU and UARTA (including clock output from the UARTA)										
P140	P140	Input	—	—	0	1	x	—	x	—	—	—	—	—	—	—	—	—	√
		Output	—	—	0	0	0/1	—	PCLBUZ0 = 0	—	—	—	—	—	—	—	—	—	√
	TS08	I/O	—	—	1	1	0	—	x	—	—	—	—	—	—	—	—	—	√
	PCLBUZ0	Output	PIOR3 = 0	—	0	0	0	—	x	—	—	—	—	—	—	—	—	—	√
	INTP6	Input	—	—	0	1	x	—	x	—	—	—	—	—	—	—	—	—	√
P146	P146	Input	—	—	0	1	x	—	—	—	—	—	—	—	—	—	—	√	√
		Output	—	—	0	0	0/1	—	—	—	—	—	—	—	—	—	—	√	√
	TS09	I/O	—	—	1	1	x	—	—	—	—	—	—	—	—	—	—	√	√
P147	P147	Input	—	0	0	1	x	—	—	—	√	√	√	√	√	√	√	√	√
		Output	—	0	0	0	0/1	—	—	—	—	—	—	—	—	—	—	—	—
	ANI18	Analog input	—	1	0	1	x	—	—	—	√	√	√	√	√	√	√	√	√
	TS10	I/O	—	1	1	1	x	—	—	—	√	√	√	√	√	√	√	√	√

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch before switching a port from input mode to output mode.

<Example> When P10 is an output port, P11 to P17 are input ports (all pins are at the high level), and the output latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 becomes FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin state, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78/G22.

<1> The Pn register is read in 8-bit units.

<2> The targeted one bit is manipulated.

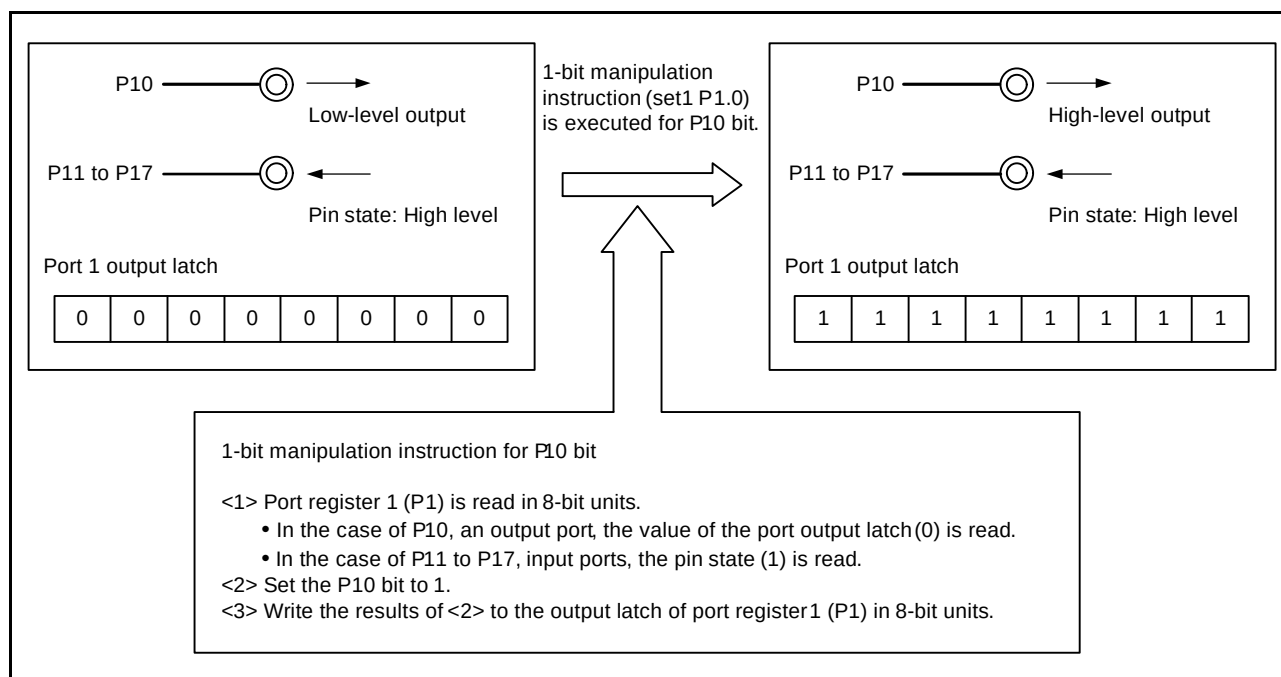
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the states of P11 to P17, which are input ports, are read. If the states of P11 to P17 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4 - 12 One-Bit Manipulation Instruction (P10)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to the initial state of the pin so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate function output, see **4.5 Register Settings When Using Alternate Function**.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

Section 5 Operation State Control

The operating voltage, operating timing, and operating current of the internal circuit are optimized using flash operation modes. Select an appropriate flash operation mode in accord with the operating voltage range and clock frequencies of the MCU.

The flash operation mode set by the option byte is selected for operation immediately after a reset is released.

The mode can be changed by setting of the respective register.

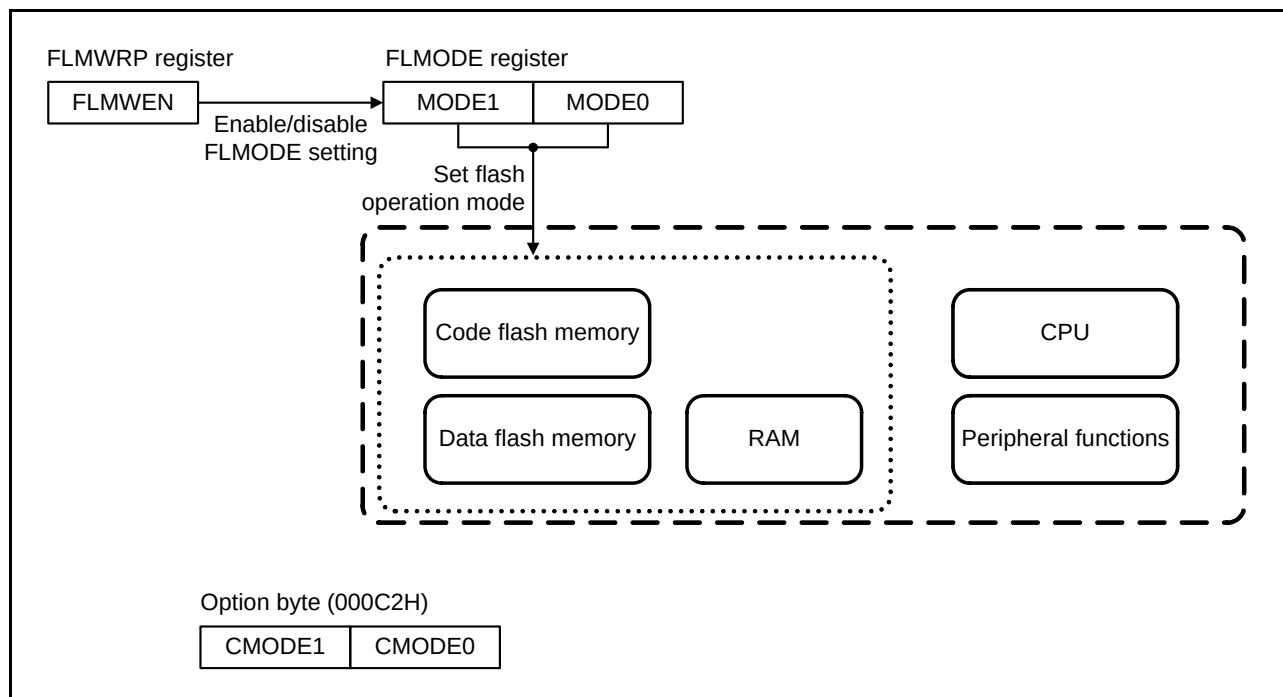
5.1 Configuration of Operation State Control

Operation state control is supported by the following hardware blocks.

Table 5 - 1 Configuration of Operation State Control

Item	Configuration
Option byte	Address of the user option byte: 000C2H
Control registers	- Flash operating mode select register (FLMODE) - Flash operating mode protect register (FLMWRP)

Figure 5 - 1 Block Diagram of Operation State Control



There are the following four flash operation modes.

- HS (high-speed main) mode
- LS (low-speed main) mode
- LP (low-power main) mode
- SUB mode

The MCU can be operated efficiently by setting these flash operation modes according to MCU operating conditions.

Table 5 - 2 lists the features of each flash operation mode.

Table 5 - 2 Features of Each Flash Operation Mode

Flash Operation Mode	Recommended Operating Range		Description
HS (high-speed main) mode	1.6 V to 1.8 V	1 MHz to 4 MHz (Rewriting of the flash memory is not possible.)	High-speed CPU operation (at 32 MHz (max.)) is possible in this mode. Suitable when CPU processing capacity is required.
	1.8 V to 5.5 V	1 MHz to 32 MHz	
LS (low-speed main) mode	1.6 V to 1.8 V	1 MHz to 4 MHz (Rewriting of the flash memory is not possible.)	The operating current and CPU operation processing (at 24 MHz (max.)) are well-balanced in this mode.
	1.8 V to 5.5 V	1 MHz to 24 MHz	
LP (low-power main) mode	1.6 V to 5.5 V	1 MHz to 2 MHz (Rewriting of the flash memory is not possible.)	The CPU operates at 1 MHz to 2 MHz in this mode. Low operating current is realized at 1 MHz to 2 MHz.
SUB mode	1.6 V to 5.5 V	32.768 kHz (Rewriting of the flash memory is not possible.)	CPU operation is driven by the subsystem clock. Note This mode enables low-power operation.

Note The subsystem clock can be derived from the subsystem clock X (fsx) or the low-speed on-chip oscillator (f_{IL}).

5.2 Registers for Controlling the Operation State Control

The following registers are used to control the operation state control.

- Flash operating mode select register (FLMODE)
- Flash operating mode protect register (FLMWRP)

5.2.1 Flash operating mode select register (FLMODE)

The FLMODE is an 8-bit register used to control flash operation modes.

This register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the value of this register cannot be changed when FLMWEN in the flash operation mode protect register (FLMWRP) is 0.

The settings of the MODE1 and MODE0 bits are updated to those of CMODE1 and CMODE0 in the option byte at address 000C2H following a reset.

Figure 5 - 2 Format of Flash Operating Mode Select Register (FLMODE)

Address: F00AAH
After reset: 40H/80H/C0HNote
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FLMODE	MODE1	MODE0	0	0	0	0	0	0

MODE1	MODE0	Selection of flash operation mode
0	0	Setting prohibited
0	1	LP (low-power main) mode (Selectable when $1\text{ MHz} \leq f_{\text{CLK}} \leq 2\text{ MHz}$ in LS mode)
1	0	LS (low-speed main) mode (Selectable when $1\text{ MHz} \leq f_{\text{CLK}} \leq 24\text{ MHz}$ or in LP mode)
1	1	HS (high-speed main) mode (Selectable when in LS mode)

Note The initial value of the FLMODE register is set to the value of the MODE1 and MODE0 bits updated with the set value of the CMODE1 and CMODE0 bits in the option byte (address: 000C2H).

- Caution 1.** The value of the FLMODE register can be changed when the FLMWEN bit in the flash operation mode protect register (FLMWRP) is 1. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.
- Caution 2.** Operation is in SUB mode when the setting of the CSS bit in the system clock control register (CKC) is 1 (operation of the CPU and peripheral functions is driven by the subsystem clock) regardless of the setting of the MODE1 and MODE0 bits.
- Caution 3.** Do not change the value of the MODE1 and MODE0 bits using the DTC or SMS.
- Caution 4.** When changing the flash operation mode, make sure that operation is possible within the voltage range and operating frequency range in the changed flash operation mode before changing the mode.

(Cautions continue on the next page.)

Caution 5. When the flash operation mode is changed by the MODE1 and MODE0 bits, the CPU enters a wait state for the following time until the mode changes. Interrupt requests are held pending during this wait period.

Flash Operation Mode Change	Change Time
LS (low-speed main) mode → HS (high-speed main) mode	225 cycles ^{Note}
LP (low-power main) mode → LS (low-speed main) mode	10 cycles ^{Note}
LS (low-speed main) mode → LP (low-power main) mode	10 cycles ^{Note}
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles ^{Note}

Note The cycle of the CPU/peripheral hardware clock (fCLK)

- Caution 6.** When rewriting the FLMODE register, insert one or more clock cycles after rewriting the FLMODE register and before writing to this register. Do not write to the FLMODE register successively.
- Caution 7.** Do not change the FLMODE register while rewriting the flash memory.
- Caution 8.** Writing new values to the FLMODE register should be done while the snooze mode sequencer is stopped, that is, while the setting of the SMSSTART bit is 0, or the settings of the SMSSTAT bit and the SMSTRGWAIT bit are 0 and 1, respectively.
- Caution 9.** Before changing the flash operating mode, set the DFLEN bit in the data flash control register (DFLCTL) to 1 to enable access to the data flash memory.
- Caution 10.** Before rewriting the contents of the code flash memory or data flash memory area by self-programming, be sure to place the MCU in HS (high-speed main) or LS (low-speed main) mode.

5.2.2 Flash operating mode protect register (FLMWRP)

The FLMWRP is an 8-bit register used to control access to the flash operation mode select register.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 5 - 3 Format of Flash Operating Mode Protect Register (FLMWRP)

Address: F00ABH

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
FLMWRP	0	0	0	0	0	0	0	FLMWEN
FLMWEN	Control of flash operation mode select register (FLMODE)							
0	Rewriting the FLMODE register is disabled.							
1	Rewriting the FLMODE register is enabled.							

5.3 Initial Setting of Flash Operation Modes

The option byte (000C2H) is used to set the initial state of flash operation mode and the high-speed on-chip oscillator after a reset is released.

Set an appropriate flash operation mode according to the V_{DD} voltage and the high-speed on-chip oscillator frequency at a reset release.

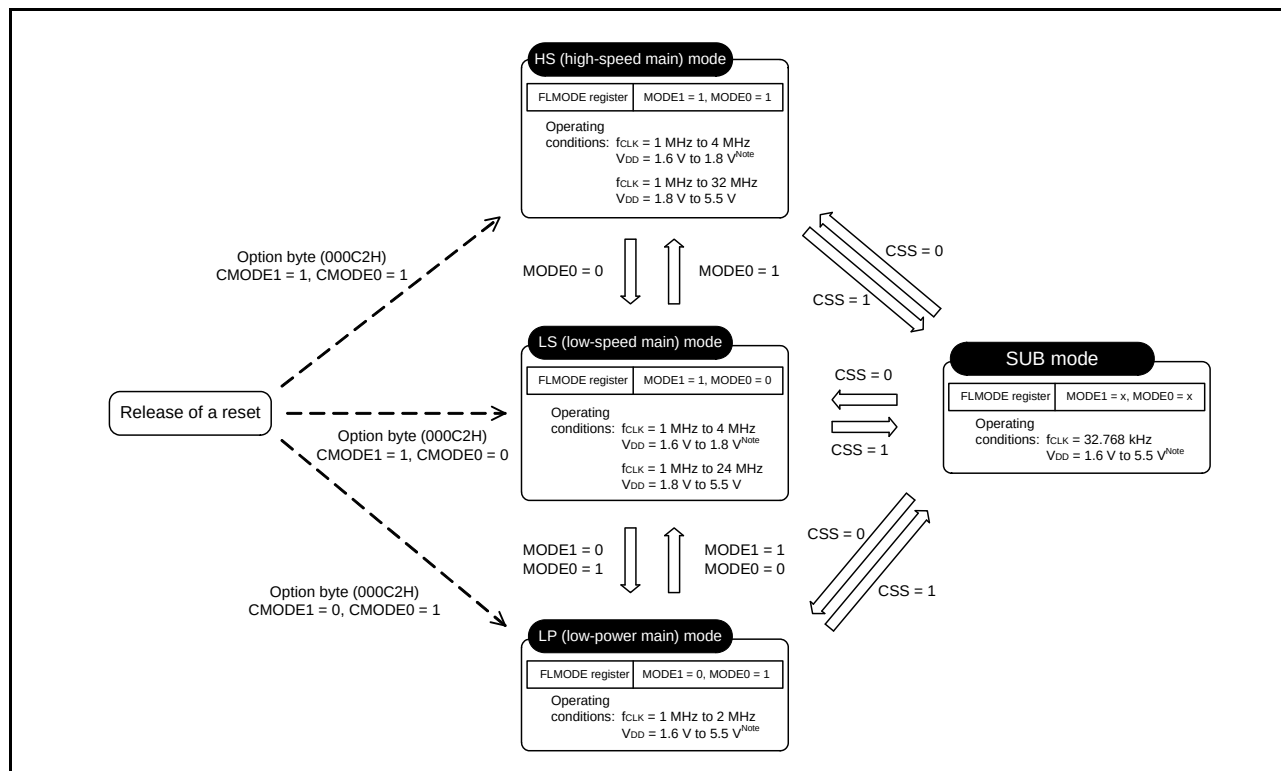
When a reset is released, the value of CMODE1 and CMODE0 is updated in MODE1 and MODE0 in the flash operation mode select register (FLMODE) and the value of FRQSEL3 to FRQSEL0 is updated in the high-speed on-chip oscillator frequency select register (HOCODIV). For details on the option byte (000C2H), see **Section 29 Option Bytes**.

5.4 Transitions between Flash Operation Modes

Setting of CMODE1 and CMODE0 in the option byte (000C2H) determines the initial flash operation mode immediately after a reset is released. The initial state can be selected from among the HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode.

The value of CMODE1 and CMODE0 is updated in the MODE1 and MODE0 bits in the flash operation mode select register (FLMODE). After that, the flash operation mode can be changed by changing the value of the FLMODE register during CPU operation. Setting the CSS bit in the CKC register enables automatically placing the MCU in SUB mode.

Figure 5 - 4 State Transitions between Flash Operation Modes



Note Rewriting of the flash memory is not possible.

Caution When a reset is applied while the MCU operates, operation always starts in the flash operation mode set by the option byte after a reset release. Therefore, make sure that operation does not start outside the operating voltage range when a reset is released by setting the LVD detection voltage to at least the operating voltage range of the flash operation mode set in the option byte.

5.5 Details of Flash Operation Modes

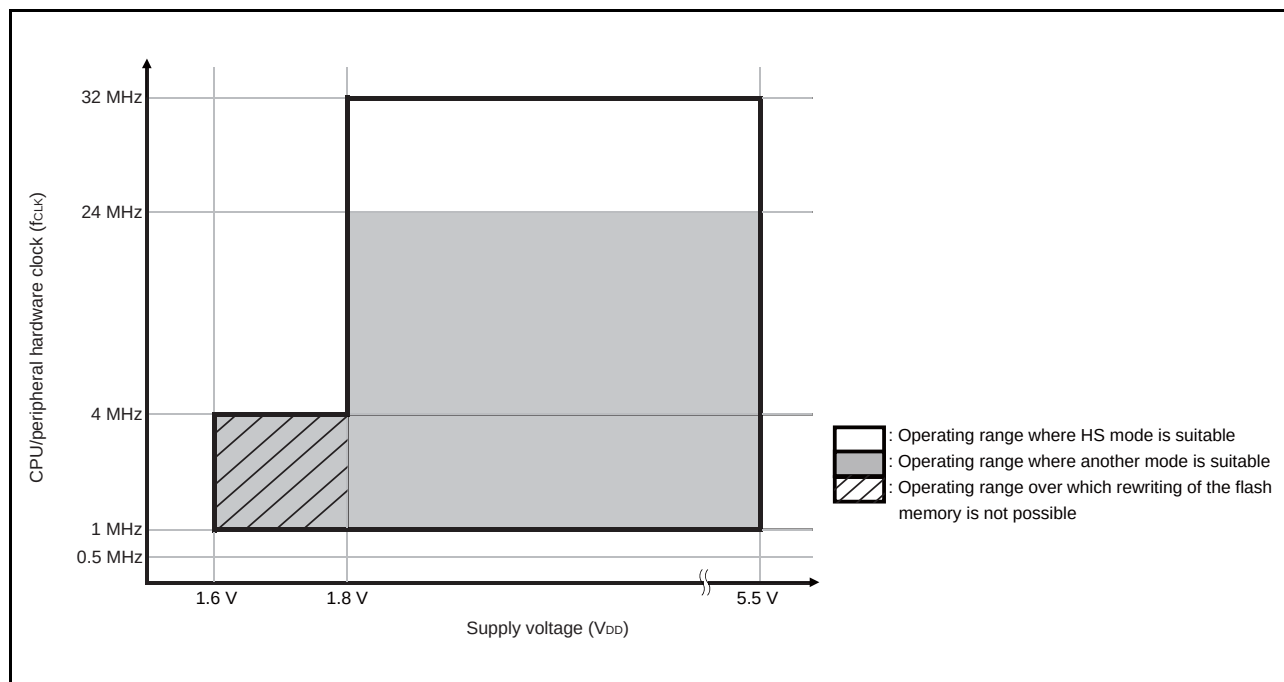
5.5.1 Details of HS (high-speed main) mode

HS (high-speed main) mode is suitable for applications that require CPU high-speed processing.

HS mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode.

The suitable operating range in HS mode is when the supply voltage is $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ and the operating frequency is $24\text{ MHz} < f_{CLK} \leq 32\text{ MHz}$. When 24 MHz or lower is used for operation, another mode can be used as the suitable flash operation mode.

Figure 5 - 5 Operating Range in HS Mode



5.5.2 Details of LS (low-speed main) mode

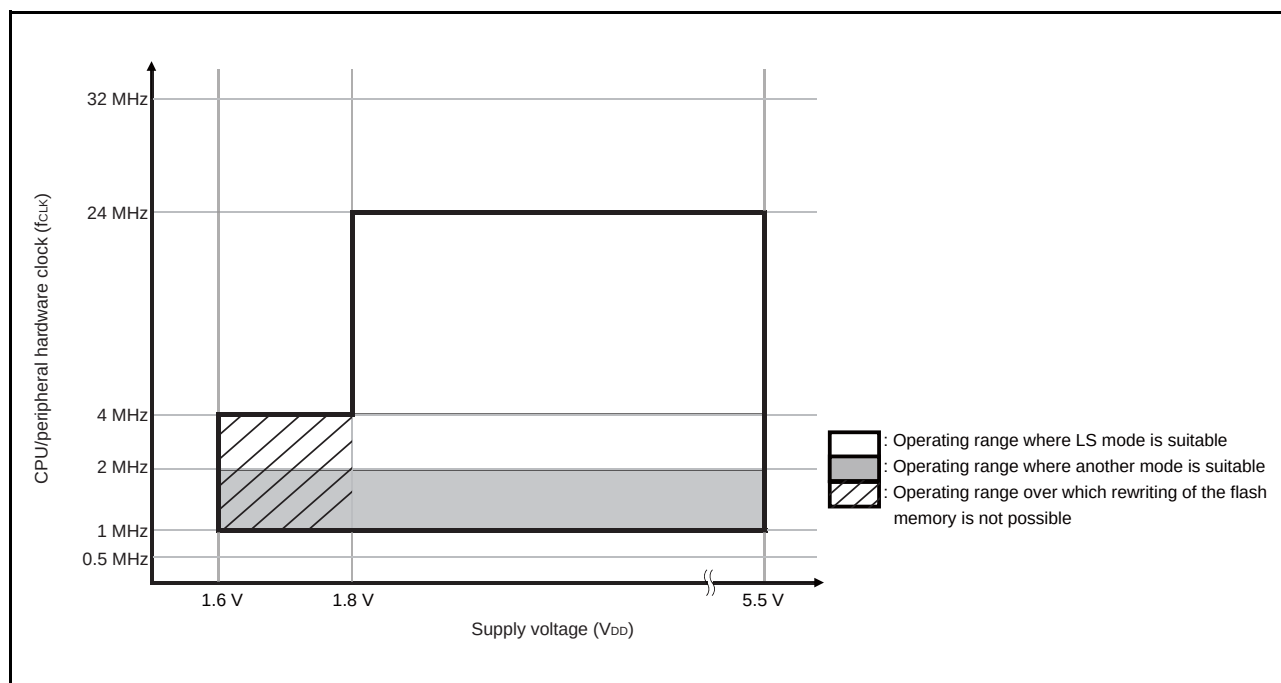
LS (low-speed main) mode supports both CPU processing capacity and operating voltage performance, suitable for applications that require low-power consumption at 2 to 24 MHz.

LS mode can be operated immediately after a reset release. Also, this mode can be entered from HS (high-speed main) mode, or LP (low-power main) mode. When entering from HS mode to LS mode, make sure that the operating frequency is $1 \text{ MHz} \leq f_{\text{CLK}} \leq 24 \text{ MHz}$.

The suitable operating range in LS mode is when the supply voltage is $1.8 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ and the operating frequency is $2 \text{ MHz} < f_{\text{CLK}} \leq 24 \text{ MHz}$, and when the supply voltage is $1.6 \text{ V} \leq V_{\text{DD}} < 1.8 \text{ V}$ and the operating frequency is $2 \text{ MHz} < f_{\text{CLK}} \leq 4 \text{ MHz}$ ^{Note}.

Note Rewriting of the flash memory is not possible.

Figure 5 - 6 Operating Range in LS Mode



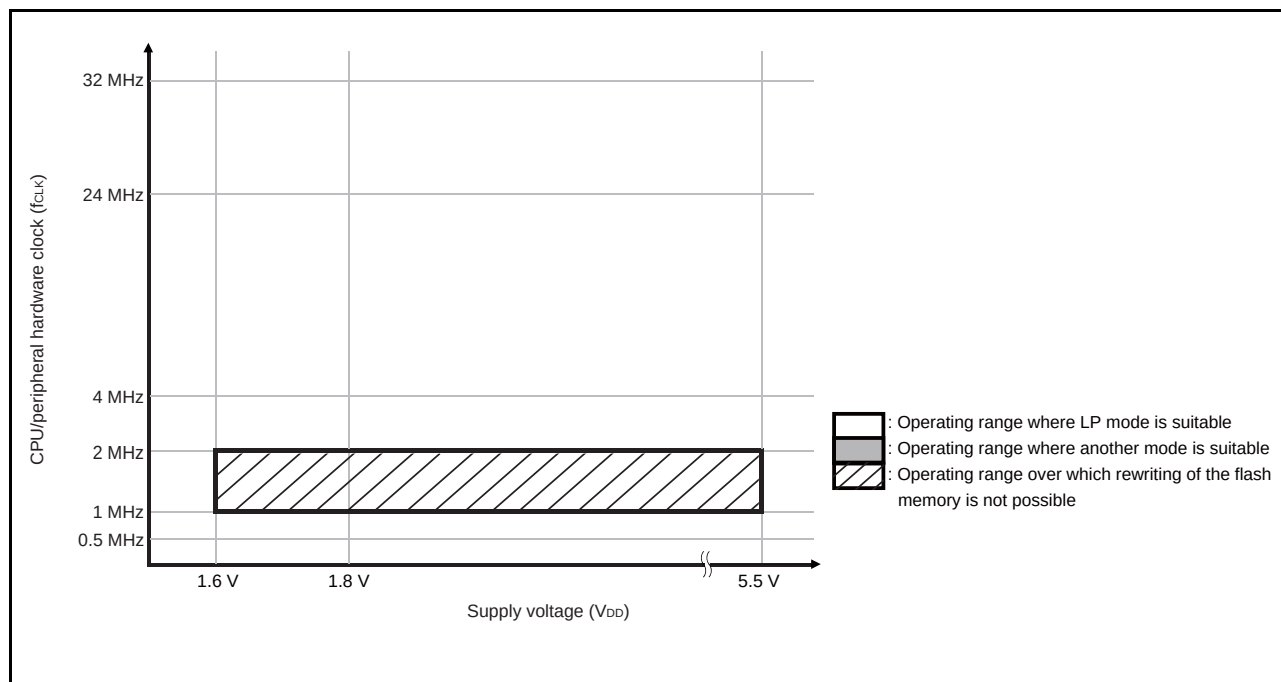
5.5.3 Details of LP (low-power main) mode

In LP (low-power main) mode, the CPU operates with a low voltage and at a frequency from 1 MHz to 2 MHz.

The CPU can operate in LP mode immediately after a reset is released. LP mode can be entered from LS (low-speed main) mode. When entering from LS mode to LP mode, make sure the operating frequency is in the range of $1 \text{ MHz} \leq f_{\text{CLK}} \leq 2 \text{ MHz}$.

The suitable operating range in LP mode is when the supply voltage is $1.6 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ and the operating frequency is $1 \text{ MHz} \leq f_{\text{CLK}} \leq 2 \text{ MHz}$. When rewriting of flash memory is required, place the CPU in LS (low-speed main) mode.

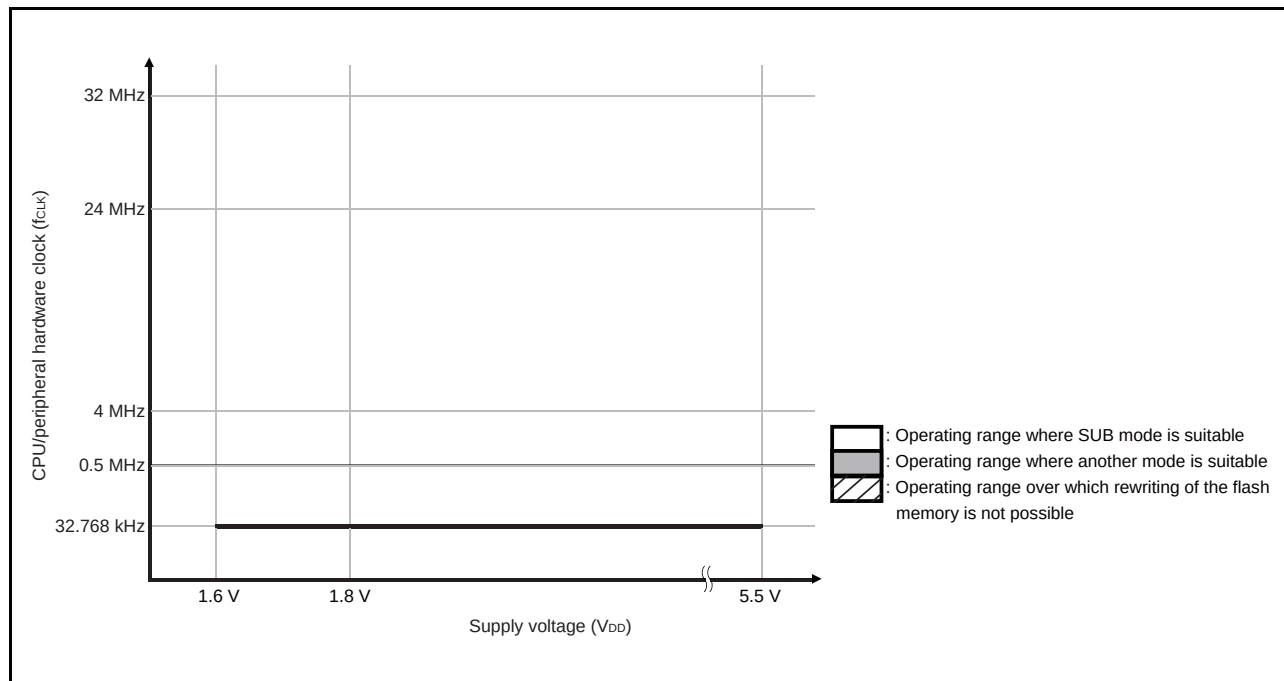
Figure 5 - 7 Operating Range in LP Mode



5.5.4 Details on SUB mode

In SUB mode, the CPU operates at a frequency of 32.768 kHz. Transition to SUB mode is possible from HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode. Setting the CSS bit in the system clock control register (CKC) places the CPU in SUB mode. Rewriting of the flash memory is not possible in SUB mode. When rewriting of flash memory is required, place the CPU in HS (high-speed main) mode or LS (low-speed main) mode.

Figure 5 - 8 Operating Range in SUB Mode



Section 6 Clock Generator

6.1 Functions of Clock Generator

The clock generator generates clocks to be supplied to the CPU and peripheral hardware.

The following kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of $f_x = 1$ to 20 MHz by connecting a resonator to the X1 pin and X2 pin. Oscillation can be stopped by executing a STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency of oscillation can be selected from among $f_{IH} = 32, 24, 16, 12, 8, 6, 4, 3, 2$, or 1 MHz (typ.) by using an option byte (000C2H). After release from the reset state, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing a STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

The frequency specified by using the option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 6 - 11 Format of High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)**.

The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)									
	1	2	3	4	6	8	12	16	24	32
$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	√	√	√	√	√	√
$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	—	—	—	—	—	—	—	—

<3> Middle-speed on-chip oscillator

The frequency of oscillation can be selected from among $f_{IM} = 4, 2$, or 1 MHz (typ.) by using the MOCODIV bits (bits 0, 1 of the MOCODIV register). Oscillation can be stopped by executing a STOP instruction or clearing of the MIOEN bit (bit 1 of the CSC register).

An external main system clock ($f_{EX} = 1$ to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. External main system clock input can be disabled by executing a STOP instruction or setting of the MSTOP bit.

As the main system clock, a high-speed system clock (X1 clock or external main system clock) or a main on-chip oscillator clock (high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock) can be selected by setting of the MCM0 and MCM1 bits (bits 4 and 0 of the system clock control register (CKC)).

Note that the usable frequency range of the CPU/peripheral hardware clock depends on the flash operation mode and VDD power supply voltage settings.

When the main system clock is to be used as the CPU/peripheral hardware clock, select the frequency of the main system clock to suit the flash operation mode specified in the CMODE0 and CMODE1 bits of an option byte (000C2H) or the flash operating mode select register (FLMODE). For details on the option byte (000C2H), see **Section 29 Option Bytes**.

(2) Subsystem clock

<1> XT1 oscillator

This circuit oscillates a clock of $f_{XT} = 32.768$ kHz by connecting a 32.768-kHz resonator to XT1 pin and XT2 pin. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock ($f_{EXS} = 32.768$ kHz) can also be supplied from the EXCLKS/XT2/P124 pin. External subsystem clock input can be disabled by the setting of the XTSTOP bit.

<2> Low-speed on-chip oscillator

This circuit oscillates a clock of $f_{IL} = 32.768$ kHz (typ.).

The low-speed on-chip oscillator operates when either of the following conditions is met.

- The value of one or more of the following bits is 1: bit 4 (WDTON) of an option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), and bit 0 (SELLOSC) of the subsystem clock select register (CKSEL)

- f_{IL} is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

However, if a HALT or STOP instruction is executed when WDTON = 1, WUTMMCK0 = 0, SELLOSC = 0 and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, this oscillator stops oscillating.

Remark	f_X :	X1 clock oscillation frequency
	f_{IH} :	High-speed on-chip oscillator clock frequency (32 MHz max.)
	f_{IM} :	Middle-speed on-chip oscillator clock frequency (4 MHz max.)
	f_{EX} :	External main system clock frequency
	f_{XT} :	XT1 clock oscillation frequency
	f_{EXS} :	External subsystem clock frequency
	f_{IL} :	Low-speed on-chip oscillator clock frequency

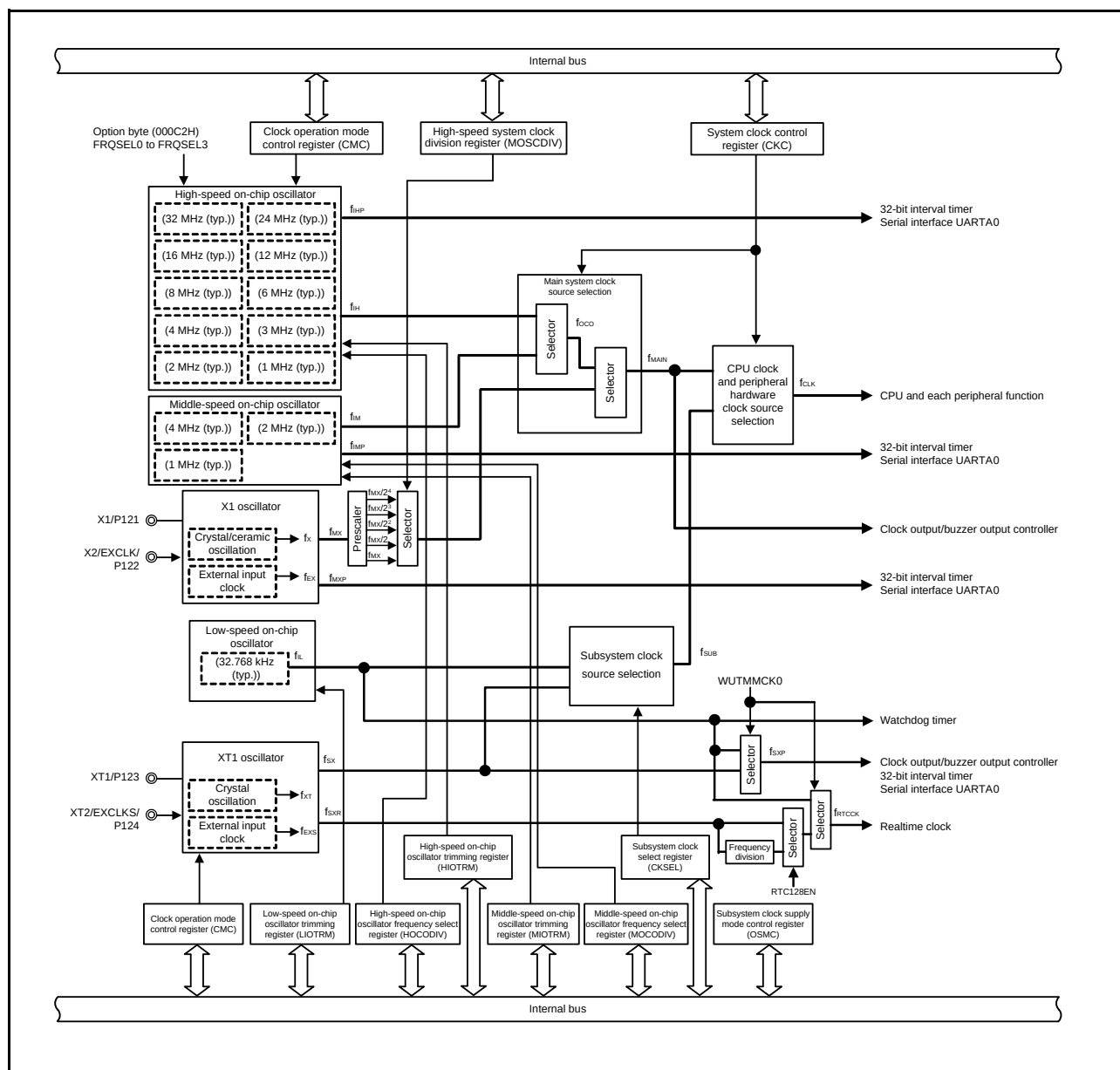
6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 6 - 1 Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1 (PER0, PER1) Subsystem clock supply mode control register (OSMC) Subsystem clock select register (CKSEL) High-speed on-chip oscillator frequency select register (HOCODIV) Middle-speed on-chip oscillator frequency select register (MOCODIV) High-speed system clock division register (MOSCDIV) High-speed on-chip oscillator trimming register (HIOTRM) Middle-speed on-chip oscillator trimming register (MIOTRM) Low-speed on-chip oscillator trimming register (LIOTRM) Standby mode release setting register (WKUPMD)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Middle-speed on-chip oscillator Low-speed on-chip oscillator

Figure 6 - 1 Block Diagram of Clock Generator



(Remark continues on the next page.)

Remark	fx:	X1 clock oscillation frequency
	fEX:	External main system clock frequency
	fIH:	High-speed on-chip oscillator clock frequency (32 MHz max.)
	fIHP:	High-speed on-chip oscillator peripheral clock frequency (32 MHz when FRQSEL3 = 1, 24 MHz when FRQSEL3 = 0)
	fIM:	Middle-speed on-chip oscillator clock frequency (4 MHz max.)
	fIMP:	Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
	fMX:	High-speed system clock frequency
	fMXP:	High-speed peripheral clock frequency
	fMAIN:	Main system clock frequency
	fXT:	XT1 clock oscillation frequency
	fEXS:	External subsystem clock frequency
	fSX:	Subsystem clock X frequency
	fSXR:	Subsystem clock XR frequency
	fRTCK:	Operating clock for the realtime clock controller
	fSXP:	Low-speed peripheral clock frequency
	fSUB:	Subsystem clock frequency
	fCLK:	CPU/peripheral hardware clock frequency
	fIL:	Low-speed on-chip oscillator clock frequency
	fOCO:	Main on-chip oscillator clock frequency (fIH or fIM)

6.3 Registers for Controlling the Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- Subsystem clock select register (CKSEL)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Middle-speed on-chip oscillator frequency select register (MOCODIV)
- High-speed system clock division register (MOSCDIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- Middle-speed on-chip oscillator trimming register (MIOTRM)
- Low-speed on-chip oscillator trimming register (LIOTRM)
- Standby mode release setting register (WKUPMD)

Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted in a product to their initial values.

6.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 pins, and to select the gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. This register can be read by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 6 - 2 Format of Clock Operation Mode Control Register (CMC) (1/2)

Address: FFFA0H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS Note 1	OSCSELS Note 1	XTSEL Notes 1, 2	AMPHS1 Note 1	AMPHS0 Note 1	AMPH

Products with 16 to 36 pins

XTSEL Notes 1, 2	EXCLK	OSCSEL	EXCLKS Note 1	OSCSELS Note 1	System clock pin operation mode	X1/P121/ XT1 pin	X2/EXCLK/ P122/XT2/ EXCLKS pin
0	0	0	0	0	Port mode	Port	Port
0	0	1	0	0	X1 oscillation mode	Crystal/ceramic resonator connection	
0	1	0	0	0	Port mode	Port	Port
0	1	1	0	0	External clock input mode	Port	External clock EXCLK input
1	0	0	0	0	Port mode	Port	Port
1	0	0	0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	0	1	0	Port mode	Port	Port
1	0	0	1	1	External clock input mode	Port	External clock EXCLKS input
Other than above					Setting prohibited		

Products with 40 to 48 pins

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 pin
0	0	Port mode	Port	Port
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Port mode	Port	Port
1	1	External clock input mode	Port	External clock input

Figure 6 - 2 Format of Clock Operation Mode Control Register (CMC) (2/2)

EXCLKS Note 1	OSCSLS Note 1	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	Input port
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	Input port
1	1	External clock input mode	Input port	External clock input

AMPHS1 Note 1	AMPHS0 Note 1	Selection of the oscillation mode of the XT1 oscillator
0	0	Low power consumption oscillation 1 (default) ^{Note 3}
0	1	Normal oscillation
1	0	Low power consumption oscillation 2 ^{Note 3}
1	1	Low power consumption oscillation 3 ^{Note 3}

AMPH	Control of the X1 clock oscillation frequency
0	$1 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_x \leq 20 \text{ MHz}$

Note 1. The EXCLKS, OSCSLS, XTSEL, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the values when a reset caused by another source occurs.

Note 2. The XTSEL bit can be written only in the products with 16 to 36 pins. Be sure to set this bit to 0 in the products with 40 to 48 pins.

Note 3. The gain and operating current of the XT1 clock oscillator decrease in the following order: low power consumption oscillation 1 > low power consumption oscillation 2 > low power consumption oscillation 3.

(Cautions and Remark are listed on the next page.)

- Caution 1.** The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. Even if you intend to use the CMC register with its initial value (00H), be sure to write 00H to the register after release from the reset state as a precaution against malfunctions (since returning the value to 00H after erroneously having written a value other than 00H to it is not possible).
- Caution 2.** After release from the reset state, set the CMC register before X1 or XT1 oscillation is started by setting the clock operation status control register (CSC).
- Caution 3.** Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.
- Caution 4.** Make the settings of the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after release from the reset state (before f_{CLK} is switched to f_{MX} or f_{SUB}).
- Caution 5.** Count the oscillation stabilization time of f_{XT} by software.
- Caution 6.** The XT1 oscillator is a circuit with low amplification in order to achieve low power consumption. Note the following points when designing the circuit.
- Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.
 - When using low power consumption oscillation 2 (AMPHS1, AMPHS0 = 1, 0) or low power consumption oscillation 3 (AMPHS1, AMPHS0 = 1, 1) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 6.7 Resonator and Oscillator Constants, before using it in either of these modes.
 - Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS1, AMPHS0 = 1, 0) or low power consumption oscillation 3 (AMPHS1, AMPHS0 = 1, 1) is selected.
 - Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.
 - Place a ground pattern that has the same voltage as V_{SS} as much as possible near the XT1 oscillator.
 - Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.
 - The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.
 - When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.
- Caution 7.** When selecting the XT1 oscillation mode for the P121/X1/XT1 and P122/X2/EXCLK/XT2/EXCLKS pins while the value of the XTSEL bit is 1 in a product with 16 to 36 pins, make sure that V_{DD} is no less than 2.4 V.
- Remark** fx: X1 clock oscillation frequency

6.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 6 - 3 Format of System Clock Control Register (CKC)

Address: FFFA4H
After reset: 00H
R/W: R/W>Note 1

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
CLS	State of the CPU/peripheral hardware clock (fCLK)							
0	Main system clock (fMAIN)							
1	Subsystem clock (fSUB)							
CSSNote 2	Selection of the CPU/peripheral hardware clock (fCLK)							
0	Main system clock (fMAIN)							
1	Subsystem clock (fSUB)							
MCS	State of the main system clock (fMAIN)							
0	Main on-chip oscillator clock (foco)							
1	High-speed system clock (fmx)							
MCM0 Note 2	Main system clock (fMAIN) operation control							
0	Selects the main on-chip oscillator clock (foco) as the main system clock (fMAIN)							
1	Selects the high-speed system clock (fmx) as the main system clock (fMAIN)							
MCS1	State of the main on-chip oscillator clock (foco)							
0	High-speed on-chip oscillator clock							
1	Middle-speed on-chip oscillator clock							
MCM1 Note 2	Main on-chip oscillator clock (foco) operation control							
0	High-speed on-chip oscillator clock							
1	Middle-speed on-chip oscillator clock							

Note 1. Bits 7, 5, and 1 are read-only.

Note 2. Changing the value of the MCM0 and MCM1 bits is prohibited while the CSS bit is set to 1.

(Cautions and Remark are listed on the next page.)

- Caution 1.** Be sure to set bits 3 and 2 of the CKC register to 0.
- Caution 2.** The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the realtime clock, clock output/buzzer output, 32-bit interval timer, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.
- Caution 3.** If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and serial interface IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the sections describing each item of peripheral hardware as well as Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$.
- Caution 4.** When switching the CPU clock between the main system clock and subsystem clock, accessing the data flash memory is prohibited until the CLS bit changes after a clock is selected by the CSS bit.
- Caution 5.** When switching the CPU clock between the main system clock and sub system clock, executing the HALT or STOP instruction is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

Remark f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)
 f_{MX}: High-speed system clock frequency
 f_{MAIN}: Main system clock frequency
 f_{SUB}: Subsystem clock frequency
 f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

6.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and subsystem clock (except the low-speed on-chip oscillator clock).

The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is C0H.

Figure 6 - 4 Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H
After reset: C0H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
CSC	MSTOP	XTSTOP Note	0	0	0	0	MIOEN	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Port mode
0	The X1 oscillator runs.	An external clock signal on the EXCLK pin is effective.	I/O port
1	The X1 oscillator is stopped.	An external clock signal on the EXCLK pin is ineffective.	

XTSTOP Note	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Port mode
0	The XT1 oscillator runs.	An external clock signal on the EXCLKS pin is effective.	Input port
1	The XT1 oscillator is stopped.	An external clock signal on the EXCLKS pin is ineffective.	

MIOEN	Middle-speed on-chip oscillator clock operation control
0	The middle-speed on-chip oscillator is stopped.
1	The middle-speed on-chip oscillator runs.

HIOSTOP	High-speed on-chip oscillator clock operation control
0	The high-speed on-chip oscillator runs.
1	The high-speed on-chip oscillator is stopped.

Note The XTSTOP bit is only initialized after a reset by a power-on reset; it retains its value when a reset caused by another source occurs.

(Cautions are listed on the next page.)

- Caution 1.** After release from the reset state, set the clock operation mode control register (CMC) before setting the CSC register.
- Caution 2.** Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after release from the reset state. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.
- Caution 3.** When starting X1 oscillation by using the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).
- Caution 4.** When starting XT1 oscillation by using the XTSTOP bit, include code to wait for oscillation of the subsystem clock to become stable.
- Caution 5.** Do not stop the clock selected for the CPU/peripheral hardware clock (fCLK) with the CSC register.
- Caution 6.** When stopping the clock, confirm the conditions before clock oscillation is stopped. For details on how to stop the clock, see Table 6 - 8 Conditions Before the Clock Oscillation is Stopped and Flag Settings.

6.3.4 Oscillation stabilization time counter status register (OSTC)

This register indicates the counter value by the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following cases:

- If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 00H following a reset, STOP instruction, or the MSTOP bit (bit 7 of the clock operation status control register (CSC)) being set to 1.

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 6 - 5 Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

MOS T8	MOS T9	MOS T10	MOS T11	MOS T13	MOS T15	MOS T17	MOS T18	State of the oscillation stabilization time		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	Less than $2^8/fx$	Less than 25.6 μs	Less than 12.8 μs
1	0	0	0	0	0	0	0	$2^8/fx$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/fx$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/fx$ min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/fx$ min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	$2^{13}/fx$ min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	$2^{15}/fx$ min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	$2^{17}/fx$ min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/fx$ min.	26.2 ms min.	13.1 ms min.

Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

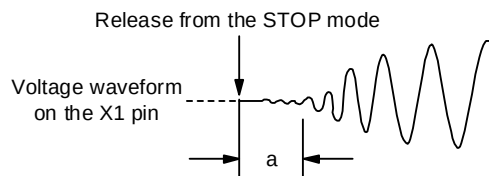
Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in the oscillation stabilization time select register (OSTS).

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register.

- If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating

Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

Caution 3. The X1 clock oscillation stabilization time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

6.3.5 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization time.

When the X1 clock is made to oscillate by clearing the MSTOP bit to start operation of the X1 oscillator, actual operation is automatically delayed for the time set in the OSTS register.

Use the oscillation stabilization time counter status register (OSTC) to confirm that the specified oscillation stabilization time has elapsed when the CPU clock is switched from the main on-chip oscillator clock or the subsystem clock to the X1 clock or entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating. The OSTC register can be used to check the counter value when counting has reached the time set beforehand in the OSTS register.

The OSTS register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 07H.

Figure 6 - 6 Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H
After reset: 07H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of the oscillation stabilization time		
				fx = 10 MHz	fx = 20 MHz
0	0	0	$2^8/f_x$	25.6 μ s	12.8 μ s
0	0	1	$2^9/f_x$	51.2 μ s	25.6 μ s
0	1	0	$2^{10}/f_x$	102 μ s	51.2 μ s
0	1	1	$2^{11}/f_x$	204 μ s	102 μ s
1	0	0	$2^{13}/f_x$	819 μ s	409 μ s
1	0	1	$2^{15}/f_x$	3.27 ms	1.63 ms
1	1	0	$2^{17}/f_x$	13.1 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.2 ms	13.1 ms

Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register.

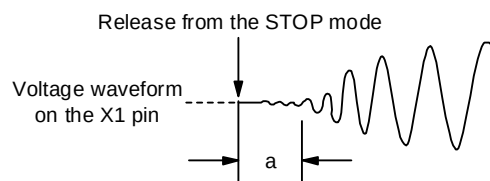
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register after the oscillation starts.

- If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.

Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

(Caution 3 and Remark continue on the next page.)

Caution 3. The X1 clock oscillation stabilization time does not include the time until clock oscillation starts (“a” below).



Remark f_x : X1 clock oscillation frequency

6.3.6 Peripheral enable registers 0, 1 (PER0, PER1)

These registers are used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

To use any of the on-chip peripheral modules listed below, the clock supplies to which are controlled by these registers, set the corresponding bit to 1 before making the initial settings of the on-chip peripheral module.

- Realtime clock
- Serial interface IICA0
- A/D converter
- Serial array unit n
- Timer array unit 0
- SNOOZE mode sequencer
- 32-bit interval timer
- DTC
- Serial interface UARTA0
- Capacitive sensing unit

Remark n = 0, 1

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of each of these registers following a reset is 00H.

Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN
	RTCWEN	Control of access to the realtime clock						
	0	The SFRs used by the realtime clock cannot be written.						
	1	The SFRs used by the realtime clock can be read and written.						
	ADCEN	Control of supply of an input clock to the A/D converter						
	0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written. • When an SFR used by the A/D converter is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.						
	IICA0EN	Control of supply of an input clock to the IICA0 serial interface						
	0	Stops supply of an input clock. • The SFRs used by the IICA0 serial interface cannot be written. • When an SFR used by the IICA0 serial interface is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by the IICA0 serial interface can be read and written.						
	SAU1EN	Control of supply of an input clock to serial array unit 1						
	0	Stops supply of an input clock. • The SFRs used by serial array unit 1 cannot be written. • When an SFR used by serial array unit 1 is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by serial array unit 1 can be read and written.						
	SAU0EN	Control of supply of an input clock to serial array unit 0						
	0	Stops supply of an input clock. • The SFRs used by serial array unit 0 cannot be written. • When an SFR used by serial array unit 0 is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by serial array unit 0 can be read and written.						

Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0) (2/2)

TAU0EN	Control of supply of an input clock to timer array unit 0
0	Stops supply of an input clock. • The SFRs used by timer array unit 0 cannot be written. • When an SFR used by timer array unit 0 is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by timer array unit 0 can be read and written.

Caution 1. Be sure to clear the following bits to 0.

16- and 20-pin products: Bits 1, 3, 4, and 6

24- and 25-pin products: Bits 1, 3, and 6

30-, 32-, 36-, 40-, and 48-pin products: Bits 1 and 6

Caution 2. Do not change the value of a bit of the PER0 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>
PER1	0	SMSSEN	0	TML32EN	DTCEN	UTAEN	0	CTSSEN

SMSSEN	Control of supply of an input clock to the SNOOZE mode sequencer
0	Stops supply of an input clock. - The SFRs used by the SNOOZE mode sequencer cannot be written. - When an SFR used by the SNOOZE mode sequencer is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the SNOOZE mode sequencer can be read and written.

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. - The SFRs used by the 32-bit interval timer cannot be written. - When an SFR used by the 32-bit interval timer is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the 32-bit interval timer can be read and written.

DTCEN	Control of supply of an input clock to the DTC
0	Stops supply of an input clock. The DTC cannot operate.
1	Enables supply of an input clock. The DTC can operate.

UTAEN	Control of supply of an input clock to the UARTAn serial interface (n = 0)
0	Stops supply of an input clock. - The SFRs used by the UARTAn serial interface cannot be written. - When an SFR used by the UARTAn serial interface is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the UARTAn serial interface can be read and written.

CTSSEN	Control of supply of an input clock to the capacitive sensing unit
0	Stops supply of an input clock. - The SFRs used by the capacitive sensing unit cannot be written. - When an SFR used by the capacitive sensing unit is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the capacitive sensing unit can be read and written.

(Cautions are listed on the next page.)

Caution 1. Be sure to clear the following bits to 0.

16-, 20-, 24-, 25-, 30-, and 32-pin products: Bits 1, 2, 5, and 7

36-, 40-, and 48-pin products: Bits 1, 5, and 7

Caution 2. Do not change the value of a bit of the PER1 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

6.3.7 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions except the realtime clock is stopped in STOP mode or in HALT mode while the CPU is operating with the subsystem clock.

The OSMC register can also be used to select the operating clock for the realtime clock, clock output/buzzer output controller, 32-bit interval timer, and serial interface UARTA0.

The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is undefined.

Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined
R/W: R/W^{Note 1}

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK 0	x	x	0	HIPREC
RTCLPC Note 4	Setting in STOP mode or in HALT mode while the CPU is operating with subsystem clock X.							
0	Enables supply of subsystem clock X to peripheral functions (See Table 20 - 1 to Table 20 - 4 for peripheral functions whose operations are enabled.)							
1	Stops supply of the subsystem clock to peripheral functions other than the realtime clock.							
WUTMMC K0	Selection of the operating clock for the realtime clock, 32-bit interval timer, serial interface UARTA0, and clock output/buzzer output controller							
0	Subsystem clock X							
1	Low-speed on-chip oscillator clock ^{Notes 2, 3}							
HIPREC	State of the high-speed on-chip oscillator clock ^{Notes 5, 6}							
0	The high-speed on-chip oscillator clock is being started at high speed and waiting for the precision of its oscillation to become stable is in progress. ^{Note 7}							
1	The high-speed on-chip oscillator clock is operating with high precision.							

Note 1. Be sure to set bits 6, 5, and 1 to 0. Bits 3, 2, and 0 are read-only. Writing to these bits is ignored.

Note 2. Do not set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.

Note 3. Switching between the subsystem clock X and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the realtime clock, 32-bit interval timer, serial interface UARTA0, and clock output/buzzer output function are stopped.

Note 4. When the subsystem clock X is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock (fs_{UB}) is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock (fs_{UB}) is not stopped.

Note 5. Undefined while the high-speed on-chip oscillator is stopped.

Note 6. For frequency accuracy of the high-speed on-chip oscillator clock, see **Section 34 Electrical Characteristics** for T_A = -40 to +105°C.

(**Note 7**, **Caution**, and **Remark** continue on the next page.)

Note 7. When the RL78/G22 is released from the STOP mode while the setting for starting the high-speed on-chip oscillator at high speed is in place (WKUPMD.FWKUP = 1), the high-speed on-chip oscillator clock runs at low precision while it is starting up. After the oscillation accuracy stabilization time, the value of the HIPREC bit changes automatically to 1.

The table below shows the frequency of the high-speed on-chip oscillator when FRQSEL3 = 0 and HIPREC = 0.

Setting of FRQSEL2 or HOCODIV2	Setting of FRQSEL1 or HOCODIV1	Setting of FRQSEL0 or HOCODIV0	Frequency of the High-Speed On-Chip Oscillator
0	0	0	16 MHz
0	0	1	8 MHz
0	1	0	4 MHz
0	1	1	2 MHz
1	0	0	Setting prohibited
1	0	1	Setting prohibited

Caution Do not execute a STOP instruction when HIPREC = 0.

Remark x: Undefined

6.3.8 Subsystem clock select register (CKSEL)

The CKSEL register is used to select the subsystem clock X or low-speed on-chip oscillator clock as the subsystem clock.

The CKSEL register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 6 - 10 Format of Subsystem Clock Select Register (CKSEL)

Address: FFFA7H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
CKSEL	0	0	0	0	0	0	0	SELLOSC

SELLOSC	Selection of the subsystem clock X or low-speed on-chip oscillator clock
0	Subsystem clock X
1	Low-speed on-chip oscillator clock ^{Note}

Note Do not set SELLOSC to 1 when the subsystem clock X (fsx) or XR (fsxR) is operating.

Caution When changing the value of the SELLOSC bit, be sure to set the CSS bit to 0 (selecting f_{MAIN}) and confirm that the value of the CLS bit is 0 before doing so.

6.3.9 High-speed on-chip oscillator frequency select register (HOCODIV)

The HOCODIV register is used to change the frequency of the high-speed on-chip oscillator set in an option byte (000C2H). Note that the selectable frequencies depend on the value of the FRQSEL3 bit of the option byte (000C2H).

The HOCODIV register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is that set in FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 6 - 11 Format of High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H
After reset: The value set in FRQSEL2 to FRQSEL0 of an option byte (000C2H)
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of the high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f _{IH} = 24 MHz	f _{IH} = 32 MHz
0	0	1	f _{IH} = 12 MHz	f _{IH} = 16 MHz
0	1	0	f _{IH} = 6 MHz	f _{IH} = 8 MHz
0	1	1	f _{IH} = 3 MHz	f _{IH} = 4 MHz
1	0	0	Setting prohibited	f _{IH} = 2 MHz
1	0	1	Setting prohibited	f _{IH} = 1 MHz
Other than above			Setting prohibited	

Caution 1. Set the HOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change. For details about the FLMODE register, see 5.2.1 Flash operating mode select register (FLMODE).

Caution 2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 3. After use of the HOCODIV register to change the frequency, actually switching of the frequency only proceeds after the following transition times have elapsed.

- The CPU/peripheral hardware clock continues to operate at the frequency before the change for up to three cycles.
- Up to three cycles of waiting are required for the CPU/peripheral hardware clock to be at the post-change frequency.

6.3.10 Middle-speed on-chip oscillator frequency select register (MOCODIV)

The MOCODIV register is used to select the frequency of the middle-speed on-chip oscillator.

The MOCODIV register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 6 - 12 Format of Middle-Speed On-Chip Oscillator Frequency Select Register (MOCODIV)

Address: F00F2H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOCODIV	0	0	0	0	0	0	MOCODIV1	MOCODIV0

MOCODIV1	MOCODIV0	Selection of the middle-speed on-chip oscillator clock frequency
0	0	4 MHz
0	1	2 MHz
1	0	1 MHz
Other than above		Setting prohibited

Caution Set the MOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

6.3.11 High-speed system clock division register (MOSCDIV)

This register is used to select the division ratio of the high-speed system clock.

The MOSCDIV register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 6 - 13 Format of High-Speed System Clock Division Register (MOSCDIV)

Address: F0214H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOSCDIV	0	0	0	0	0	MOSCDIV2	MOSCDIV1	MOSCDIV0

MOSCDIV2	MOSCDIV1	MOSCDIV0	Selected division ratio for the high-speed system clock	f _{MX} = 20 MHz
0	0	0	f _{MX}	20 MHz
0	0	1	f _{MX} /2	10 MHz
0	1	0	f _{MX} /4	5 MHz
0	1	1	f _{MX} /8	2.5 MHz
1	0	0	f _{MX} /16	1.25 MHz
Other than above			Setting prohibited	

Caution Set the MOSCDIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

6.3.12 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

The accuracy of the high-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways.

The HIOTRM register can be set by an 8-bit memory manipulation instruction.

Caution The frequency will vary if the temperature and VDD pin voltage change after accuracy adjustment. When the temperature and VDD voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 6 - 14 Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H
After reset: **Note**
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
• • •						
1	1	1	1	1	0	
1	1	1	1	1	1	

Note The value of this register following a reset is that adjusted at shipment.

Remark 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

Remark 2. For the usage example of the HIOTRM register, see the **Application Note for RL78 MCU Series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN2833)**.

6.3.13 Middle-speed on-chip oscillator trimming register (MIOTRM)

This register is used to adjust the accuracy of the middle-speed on-chip oscillator.

The accuracy of the middle-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways.

The MIOTRM register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 90H.

Caution The frequency will vary if the temperature and VDD pin voltage change after accuracy adjustment. When the temperature and VDD voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 6 - 15 Format of Middle-Speed On-Chip Oscillator Trimming Register (MIOTRM)

Address: F0212H
After reset: 90H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIOTRM	MIOTRM7	MIOTRM6	MIOTRM5	MIOTRM4	MIOTRM3	MIOTRM2	MIOTRM1	MIOTRM0
	MIOTRM7	MIOTRM6	MIOTRM5	MIOTRM4	MIOTRM3	MIOTRM2	MIOTRM1	MIOTRM0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1
	1	0	0	0	1	1	1	1
	1	0	0	1	0	0	0	0
	1	0	0	1	0	0	0	1
	1	1	1	1	1	1	1	0
	1	1	1	1	1	1	1	1

Remark For details about the accuracy adjustment resolution of the middle-speed on-chip oscillator clock, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

6.3.14 Low-speed on-chip oscillator trimming register (LIOTRM)

This register is used to adjust the accuracy of the low-speed on-chip oscillator.

The accuracy of the low-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways.

The LIOTRM register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 80H.

Caution The frequency will vary if the temperature and VDD pin voltage change after accuracy adjustment. When the temperature and VDD voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 6 - 16 Format of Low-Speed On-Chip Oscillator Trimming Register (LIOTRM)

Address: F0213H
After reset: 80H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LIOTRM	LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0
	LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1
	0	1	1	1	1	1	1	1
	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	1
	1	1	1	1	1	1	1	0
	1	1	1	1	1	1	1	1

Remark For details about the accuracy adjustment resolution of the low-speed on-chip oscillator clock, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

6.3.15 Standby mode release setting register (WKUPMD)

This register is used to set the operation when the standby mode is released.
The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of this register following a reset is 00H.

Figure 6 - 17 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode ^{Notes 1, 2}
0	Starting of the high-speed on-chip oscillator is at normal speed. ^{Note 3}
1	Starting of the high-speed on-chip oscillator is at high speed. ^{Note 3}

- Note 1.** This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.
- Note 2.** This register is initialized when the RL78/G22 is released from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.
- Note 3.** For the activation time, see **Section 20 Standby Function**.
The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

6.4 System Clock Oscillator

6.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins. An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

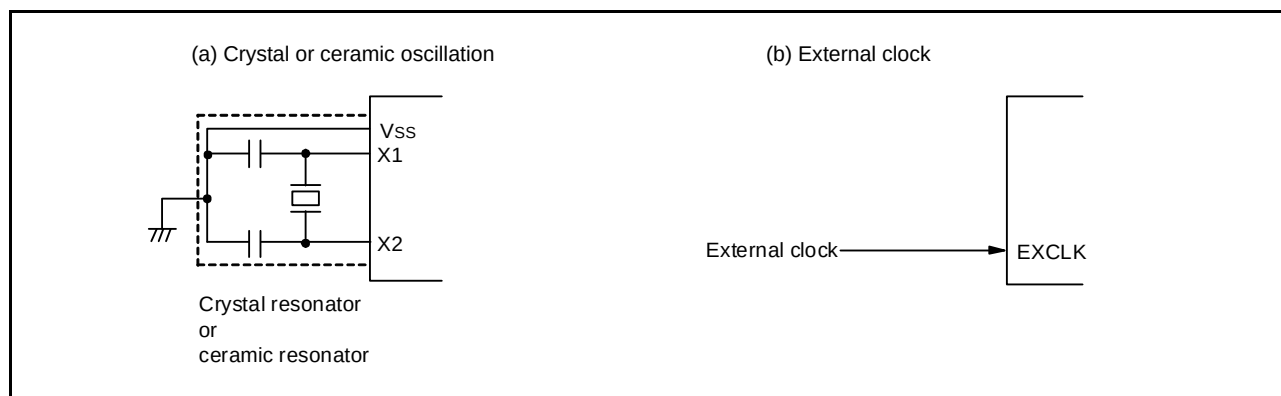
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

When the X1 oscillator is not used, set the port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as port pins, either, see **Table 2 - 3 Connections of Unused Pins**.

Figure 6 - 18 shows **Examples of External Circuits for the X1 Oscillator**.

Figure 6 - 18 Examples of External Circuits for the X1 Oscillator



(Caution is listed on the next page.)

6.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (typ.)) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

An external clock can also be input. In this case, input the clock signal to the EXCLKS pin. To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

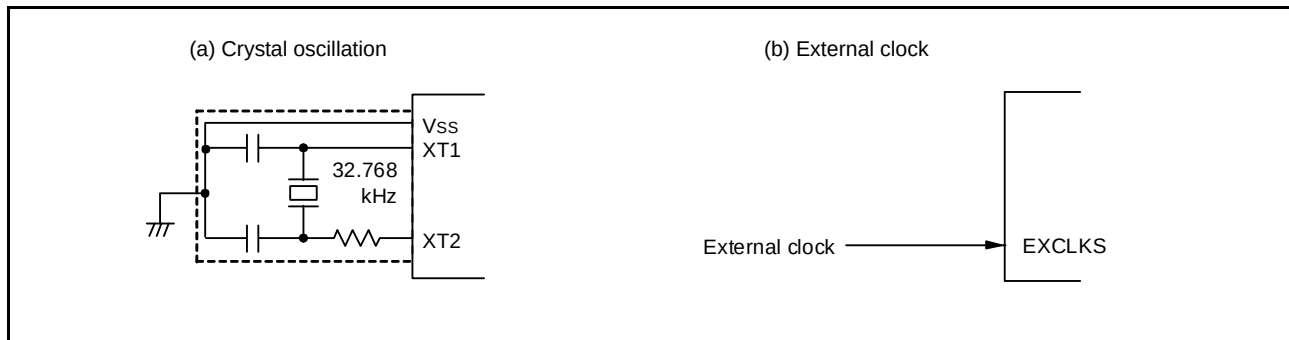
- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When XT1 oscillator is not used, and the pins are not used as input port pins, either, see **Table 2 - 3 Connections of Unused Pins**.

Figure 6 - 19 shows Examples of External Circuits for the XT1 Oscillator.

Figure 6 - 19 Examples of External Circuits for the XT1 Oscillator



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed in broken lines in the Figure 6 - 18 and Figure 6 - 19 to avoid an adverse effect from wiring capacitance.

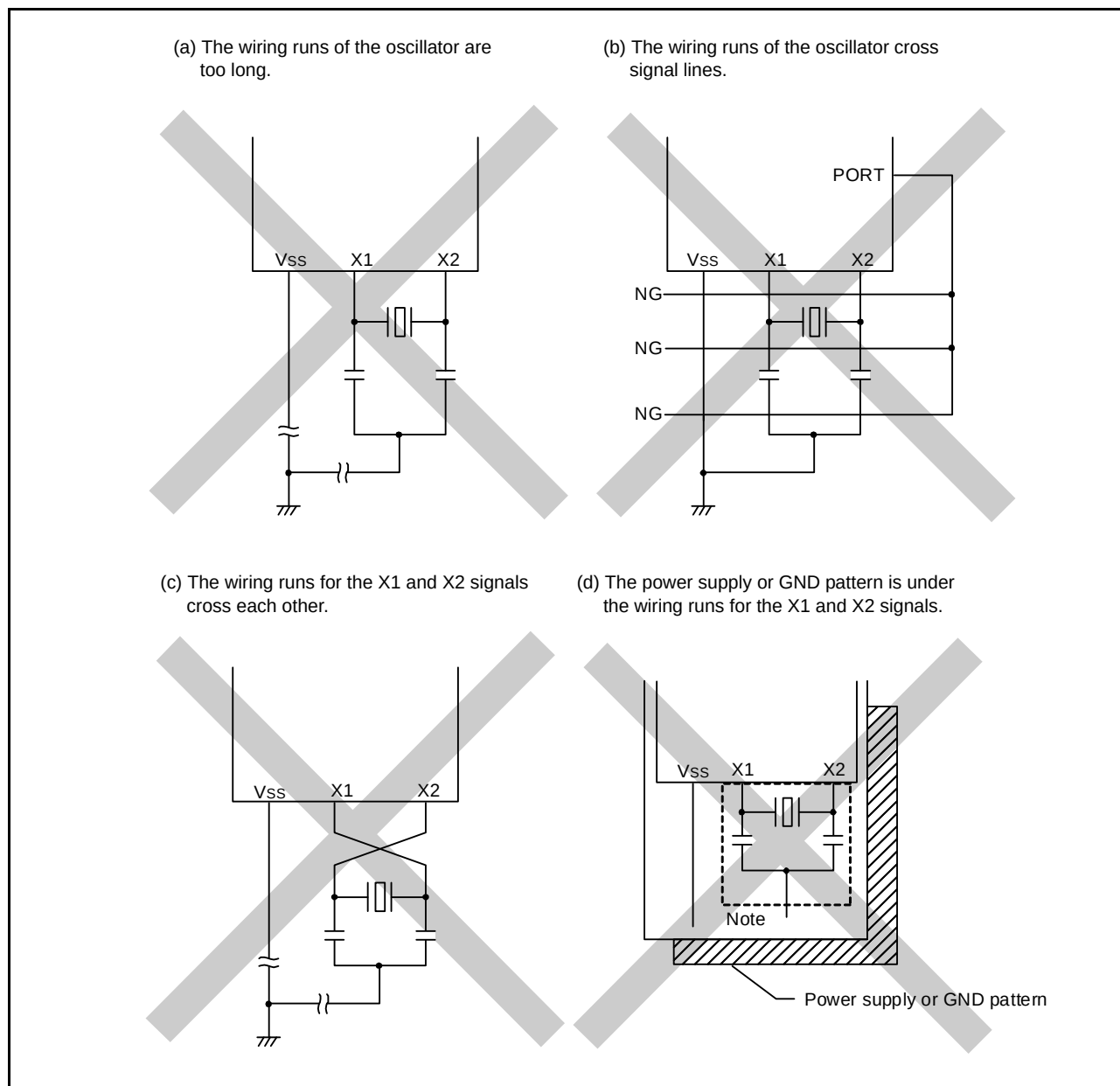
- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.
- Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.
- Do not bring a signal line out from the oscillator.

The XT1 oscillator is a circuit with low amplification in order to achieve low power consumption. Note the following points when designing the circuit.

- Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.
- When using low power consumption oscillation 2 (AMPHS1, AMPHS0 = 1, 0) or low power consumption oscillation 3 (AMPHS1, AMPHS0 = 1, 1) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 6.7 Resonator and Oscillator Constants, before using it in either of these modes.
- Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS1, AMPHS0 = 1, 0) or low power consumption oscillation 3 (AMPHS1, AMPHS0 = 1, 1) is selected.
- Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.
- Place a ground pattern that has the same voltage as Vss as much as possible near the XT1 oscillator.
- Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.
- The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.
- When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.

Figure 6 - 20 shows examples of incorrect resonator connection.

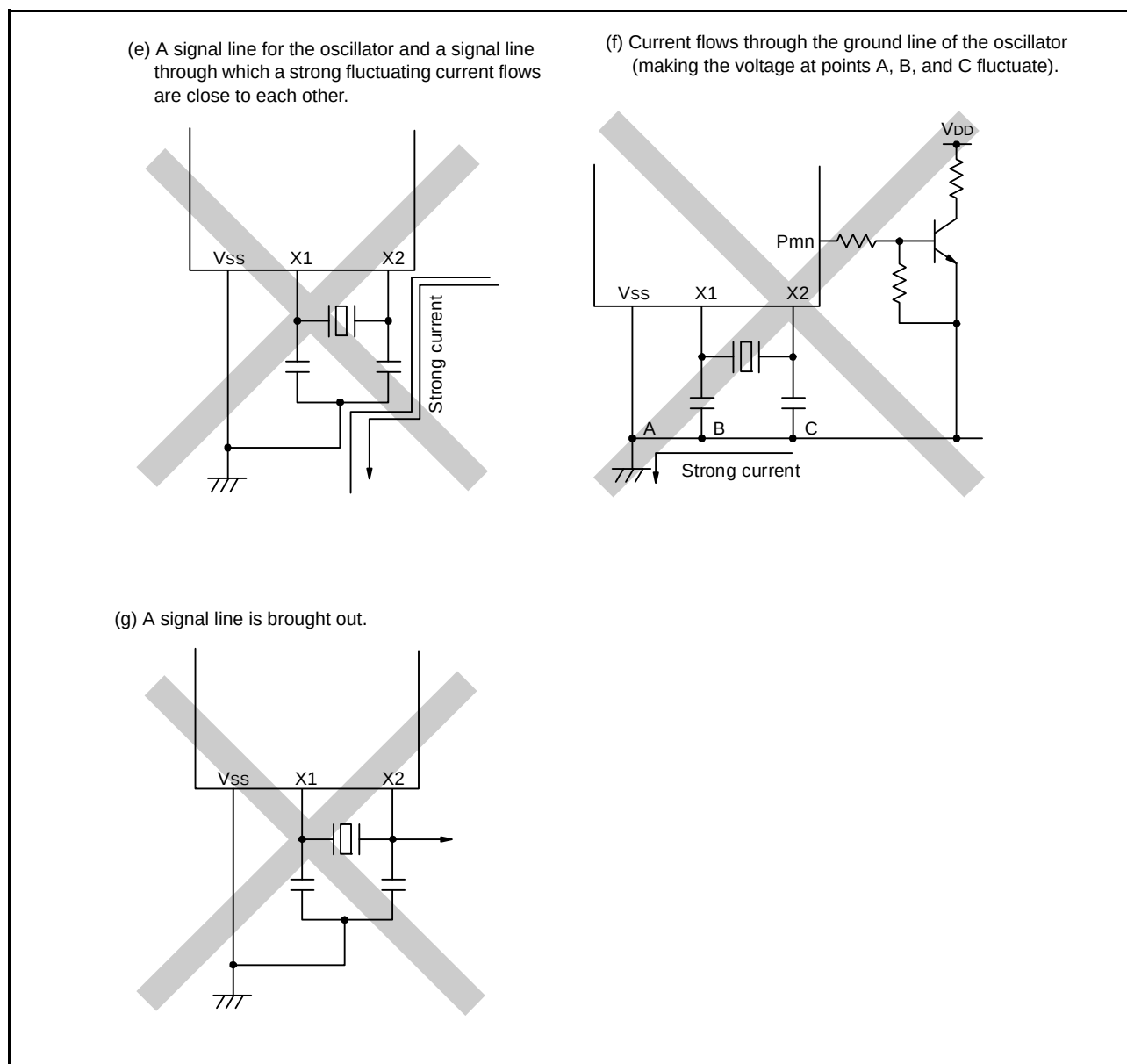
Figure 6 - 20 Examples of Incorrect Resonator Connection (1/2)



Note Do not place a power supply or GND pattern under the wiring section (section enclosed in broken lines in the figure) of the X1 and X2 pins and the resonator in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

Figure 6 - 20 Examples of Incorrect Resonator Connection (2/2)



Caution When the wiring runs for the X2 and X1 signals are in parallel, crosstalk noise from the X2 signal line may be imposed on the X1 signal, resulting in malfunctions.

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

6.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G22. The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using an option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after release from the reset state.

6.4.4 Middle-speed on-chip oscillator

The middle-speed on-chip oscillator is incorporated in the RL78/G22. Oscillation can be controlled by bit 1 (MIOEN) of the clock operation status control register (CSC).

6.4.5 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G22.

The low-speed on-chip oscillator operates when any of the following conditions is met.

- The watchdog timer is operating.
- The value of one or both of bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC) and bit 0 (SELLOSC) of the subsystem clock select register (CKSEL) is 1.
- fil is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and both WUTMMCK0 and SELLOSC are set to 0.

6.5 Operations of the Clock Generator

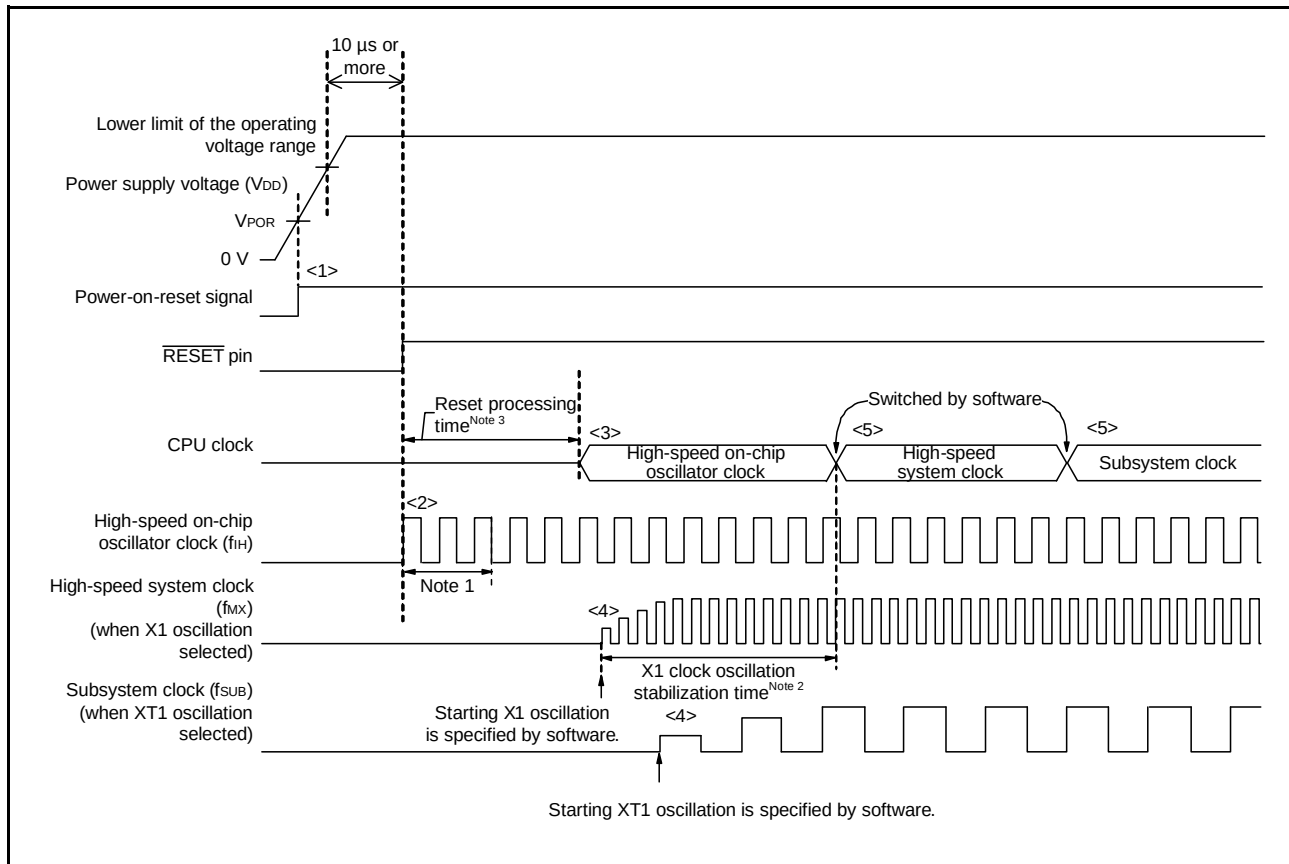
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 1**).

- Main system clocks (f_{MAIN})
 - High-speed system clocks (f_{MX})
 - X1 clock (f_X)
 - External main system clock (f_{EX})
 - High-speed on-chip oscillator clock (f_{IH})
 - Middle-speed on-chip oscillator clock (f_{IM})
- Subsystem clocks (f_{SUB})
 - XT1 clock (f_{XT})
 - External subsystem clock (f_{EXS})
 - Low-speed on-chip oscillator clock (f_{IL})
- CPU/peripheral hardware clock (f_{CLK})
- Subsystem clock X (f_{SX})
- Peripheral clocks
 - High-speed on-chip oscillator peripheral clock (f_{IHP})
 - Middle-speed on-chip oscillator peripheral clock (f_{IMP})
 - High-speed peripheral clock (f_{MXP})
 - Low-speed peripheral clock (f_{SXP})
 - Subsystem clock XR (f_{SXR})

The CPU starts operation when the high-speed on-chip oscillator starts outputting after release from the reset state in the RL78/G22.

Clock Generator Operation When Power Supply Voltage is Turned On is shown in **Figure 6 - 21**.

Figure 6 - 21 Clock Generator Operation When Power Supply Voltage is Turned On



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit.

Note that the reset state is maintained after a reset by the voltage detection circuit or an external reset until the voltage reaches the range of operating voltage described in **34.4 AC Characteristics** (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation with the high-speed on-chip oscillator clock after waiting for the voltage to become stable and a reset processing have been performed after release from the reset state.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **6.6.2 Example of setting the X1 oscillator clock** and **6.6.3 Example of setting the XT1 oscillator clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to become stable, and then make the setting for switching via software (see **6.6.2 Example of setting the X1 oscillator clock** and **6.6.3 Example of setting the XT1 oscillator clock**).

Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

Note 3. For the reset processing time, see **Section 22 Power-on-Reset Circuit (POR)**.

Caution Waiting for the oscillation stabilization time is not required when external clock input through the EXCLK pin is used.

6.6 Controlling Clocks

6.6.1 Example of setting the high-speed on-chip oscillator

After release from the reset state, the CPU/peripheral hardware clock (fCLK) always starts operating with the high-speed on-chip oscillator clock as the source. The frequency of the high-speed on-chip oscillator can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using FRQSEL0 to FRQSEL3 of an option byte (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)	7	6	5	4	3	2	1	0
	CMODE1 0/1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of the high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f _{IH} = 24 MHz	f _{IH} = 32 MHz
0	0	1	f _{IH} = 12 MHz	f _{IH} = 16 MHz
0	1	0	f _{IH} = 6 MHz	f _{IH} = 8 MHz
0	1	1	f _{IH} = 3 MHz	f _{IH} = 4 MHz
1	0	0	Setting prohibited	f _{IH} = 2 MHz
1	0	1	Setting prohibited	f _{IH} = 1 MHz
Other than above			Setting prohibited	

6.6.2 Example of setting the X1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (fCLK) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the X1 oscillator clock, set and start the oscillator by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC), and wait for the oscillation to become stable by using the oscillation stabilization time counter status register (OSTC). After the oscillation becomes stable, set the X1 oscillator clock as the source of fCLK by using the system clock control register (CKC).

[Register settings] Set the registers in the order of <1> to <5> below.

<1> Set the OSCSEL bit of the CMC register to 1 to make the X1 oscillator operate. Also set the AMPH bit to 1 if fx is more than 10 MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	EXCLKS 0	OSCELS 0	XTSEL 0	AMPHS1 0	AMPHS0 0	AMPH 0/1

<2> Use the OSTS register to select the oscillation stabilization time for the X1 oscillator after release from the STOP mode.

Example: Set the register as shown below if waiting is to be for at least 102 μ s with a 10-MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2 0	OSTS1 1	OSTS0 0

<3> Clear the MSTOP bit of the CSC register to 0 to start X1 oscillation.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	XTSTOP 1	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to become stable.

Example: Wait until counting has reached the following value if waiting is to be for at least 102 μ s with a 10-MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	0	0	MCS1 0	MCM1 0

Caution Set the HOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change. For details about the FLMODE register, see 5.2.1 Flash operating mode select register (FLMODE).

6.6.3 Example of setting the XT1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (fCLK) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the XT1 oscillator clock, set and start the oscillator by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), and set the XT1 oscillator clock as the source of fCLK by using the system clock control register (CKC).

[Register settings] Set the registers in the order of <1> to <5> below.

<1> To select only running the realtime clock by the subsystem clock (for ultra-low current) in STOP mode or in HALT mode while the CPU is also operating with the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 0	x	x	0	HIPREC 0

<2> Set the OSCSELS bit of the CMC register to 1 to make the XT1 oscillator operate. Also set the XTSEL bit to 1 in a product with 16 to 36 pins.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	XTSEL 0/1	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear the XTSTOP bit of the CSC register to 0 to start XT1 oscillation.

	7	6	5	4	3	2	1	0
CSC	MSTOP 1	XTSTOP 0	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Include code to wait for oscillation of the subsystem clock to become stable by using a timer or in other ways.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 1	MCS 0	MCM0 0	0	0	MCS1 0	MCM1 0

Remark x: Undefined

6.6.4 State transitions of the CPU clock

Figure 6 - 22 shows the state transitions of the CPU clock in this product.

Figure 6 - 22 State Transitions of the CPU Clock

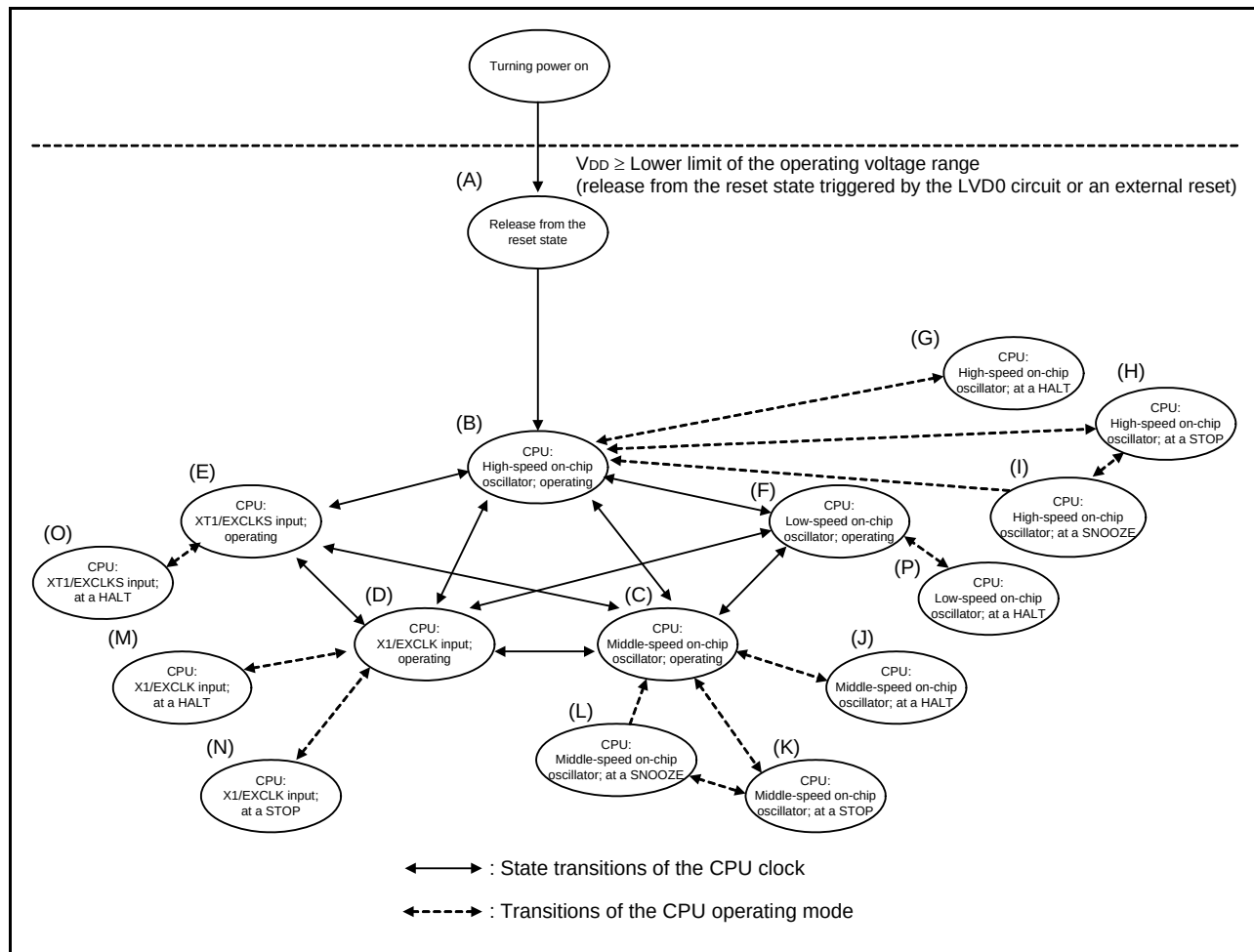


Table 6 - 2 (1/4) to Table 6 - 2 (4/4) show examples of transitions of the CPU clock and SFR settings.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (1/4)

- (1) Transition to state (B) where the CPU is operating with the high-speed on-chip oscillator clock after release from the reset state (A)

Scope of state transitions: (A) → (B)

Clock after Transition	SFR Setting
High-speed on-chip oscillator	SFR setting not required (default state after release from the reset state)

- (2) Transitions to state (B) where the CPU is operating with the high-speed on-chip oscillator clock

Scope of state transitions: (C) → (B), (D) → (B), (E) → (B), (F) → (B)

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	HIOSTOP		CSS	MCM0	MCM1
High-speed on-chip oscillator	0	5 μs	0	0	0

Unnecessary if the high-speed on-chip oscillator clock is already running

- (3) Transitions to state (C) where the CPU is operating with the middle-speed on-chip oscillator clock

Scope of state transitions: (B) → (C), (D) → (C), (E) → (C), (F) → (C)

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	MIOEN		CSS	MCM0	MCM1
Middle-speed on-chip oscillator	1	1 μs	0	0	1

Unnecessary if the middle-speed on-chip oscillator clock is already running

Remark (A) to (P) in **Table 6 - 2** correspond to (A) to (P) in **Figure 6 - 22**.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (2/4)

(4) Transitions to state (D) where the CPU is operating with the high-speed system clock

Scope of state transitions: (B) → (D), (C) → (D), (E) → (D)^{Note 1}, (F) → (D)

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CMC Register ^{Note 2}			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
X1 clock: $1 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$	0	1	0	Note 3	0	Must be checked	0	1
X1 clock: $10 \text{ MHz} < f_x \leq 20 \text{ MHz}$	0	1	1	Note 3	0	Must be checked	0	1
External main clock	1	1	x	Note 3	0	Need not be checked	0	1

Unnecessary if these bits are already set

Unnecessary if the high-speed system clock is already running

Note 1. Products with 16 to 36 pins do not support this transition.

Note 2. The clock operation mode control register (CMC) can be changed only once after release from the reset state. This register setting is not necessary if it has already been set.
In the products with 16 to 36 pins, set XTSEL to 0.

Note 3. Set the oscillation stabilization time as follows.

- Desired oscillation stabilization time counted by the oscillation stabilization time counter status register (OSTC) $\leq$ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set a clock after the supply voltage has reached the operating voltage range for the clock to be set (see Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$).

Remark (A) to (P) in Table 6 - 2 correspond to (A) to (P) in Figure 6 - 22.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (3/4)

(5) Transitions to state (E) where the CPU is operating with the subsystem clock

Scope of state transitions: (B) → (E), (C) → (E), (D) → (E)^{Note 1}

(Sequence of setting the SFRs)

SFR Bit to be Set Clock after Transition	CMC Register ^{Note 2}				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
XT1 clock	0	1	0/1	0/1	0	Necessary	1
External subsystem clock	1	1	x	x	x	Necessary	1

Unnecessary if these bits are already set

Unnecessary if the subsystem
clock is already running

Note 1. Products with 16 to 36 pins do not support this transition.

Note 2. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after release from the reset state.
In the products with 16 to 36 pins, set XTSEL to 1.

(6) Transitions to state (F) where the CPU is operating with the low-speed on-chip oscillator clock

Scope of state transitions: (B) → (F), (C) → (F), (D) → (F)

(Sequence of setting the SFRs)

SFR Bit to be Set Clock after Transition	CKSEL	Oscillation accuracy stabilization time	CKC Register
	SELLOSC		CSS
Low-speed on-chip oscillator	1	80 μs	1

Unnecessary if the low-
speed on-chip oscillator
clock is already running

Remark 1. x: Don't care.

Remark 2. (A) to (P) in Table 6 - 2 correspond to (A) to (P) in Figure 6 - 22.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (4/4)

- (7) Transitions from the CPU operating mode (B), (C), (D), (E), or (F) to the HALT mode (G), (J), (M), (O), or (P)
Scope of state transitions: (B) → (G), (C) → (J), (D) → (M), (E) → (O), (F) → (P)

Mode after Transition	Description
HALT mode	Executing a HALT instruction

- (8) Transitions from the CPU operating mode (B), (C), or (D) to the STOP mode (H), (K), or (N)
Scope of state transitions: (B) → (H), (C) → (K), (D) → (N)

(Setting sequence) →

Mode after Transition	Description			
STOP mode	Stopping peripheral functions that cannot operate in STOP mode	Setting the OSTS register	Confirming the value of the HIPREC bit is 1.	Executing a STOP instruction

When STOP or SNOOZE mode is released

- FWKUP = 0 if the high-speed on-chip oscillator is started at normal speed.
- FWKUP = 1 if the high-speed on-chip oscillator is started at high speed.

Only necessary if the CPU is shifting from the state of operating with the high-speed on-chip oscillator to the STOP mode

Only necessary if the CPU is shifting from the state of operating with the high-speed system clock to the STOP mode

Only necessary if the high-speed on-chip oscillator is started at high speed

- (9) Transitions between a STOP mode (H) or (K) and SNOOZE mode (I) or (L)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **Section 20 Standby Function** and descriptions of the SNOOZE mode functions of the on-chip peripheral modules.

Remark (A) to (P) in **Table 6 - 2** correspond to (A) to (P) in **Figure 6 - 22**.

6.6.5 Conditions before changing the CPU clock and processing after changing the CPU clock

The conditions before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 6 - 3 Changing the CPU Clock (1/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
High-speed on-chip oscillator clock	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0 ^{Note 1} • The oscillation stabilization time has elapsed.	
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0 ^{Note 1}	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1 ^{Note 1} • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1 ^{Note 1}	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (2/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
Middle-speed on-chip oscillator clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating. • HIOSTOP = 0	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0 ^{Note 1} • The oscillation stabilization time has elapsed.	
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0 ^{Note 1}	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1 ^{Note 1} • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1 ^{Note 1}	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (3/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
X1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	External main system clock	Transition is not possible.	—
	XT1 clock ^{Note 2}	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock ^{Note 2}	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. • SELLOSC = 1	
External main system clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	X1 clock	Transition is not possible.	—
	XT1 clock ^{Note 2}	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock ^{Note 2}	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (4/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock ^{Note 2}	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock ^{Note 2}	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock	Transition is not possible.	

Table 6 - 3 Changing the CPU Clock (5/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock ^{Note 2}	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock ^{Note 2}	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock	Transition is not possible.	

Table 6 - 3 Changing the CPU Clock (6/6)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
Low-speed on-chip oscillator clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	—
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0 ^{Note 1} • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock	External clock input from the EXCLK pin is enabled and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0 ^{Note 1} • MCS = 1	
	XT1 clock	Transition is not possible.	—
	External subsystem clock	Transition is not possible.	

Note 1. Writing to the XTSEL bit is only possible in the products with 16 to 36 pins.

Note 2. Switching to this clock is only possible in the products with 40 to 48 pins.

6.6.6 Time required for switchover of the CPU clock and main system clock

By setting bits 6, 4, 0 (CSS, MCM0, MCM1) of the system clock control register (CKC), the CPU clock can be switched between the main system clock and subsystem clock, the main system clock can be switched between the on-chip oscillator clock and high-speed system clock, and the on-chip oscillator clock can be switched between the high-speed on-chip oscillator clock and middle-speed on-chip oscillator clock.

In actual operation, the clock is not switched immediately after writing to the CKC register; the CPU continues to operate with the prior clock for several clock cycles after writing proceeds (see **Table 6 - 4** to **Table 6 - 7**).

Whether the source of the CPU clock is the main system clock or subsystem clock can be ascertained from bit 7 (CLS) of the CKC register. Whether the source of the main system clock is the high-speed system clock or main on-chip oscillator clock can be ascertained from bit 5 (MCS) of the CKC register. Whether the source of the main on-chip oscillator clock is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock can be ascertained from bit 1 (MCS1) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 6 - 4 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
foco	←→	fmx	See Table 6 - 5 .
fih	←→	fim	See Table 6 - 6 .
fMAIN	←→	fSUB	See Table 6 - 7 .

Table 6 - 5 Maximum Number of Clock Cycles Required for foco ↔ fmx

Set Value before Switchover		Set Value after Switchover	
MCM0		MCM0	
		0 (fMAIN = foco)	1 (fMAIN = fmx)
0 (fMAIN = foco)	fmx ≥ foco		2 cycles
	fmx < foco		2 foco/fmx cycles
1 (fMAIN = fmx)	fmx ≥ foco	2 fmx/foco cycles	
	fmx < foco	2 cycles	

Table 6 - 6 Maximum Number of Clock Cycles Required for fih ↔ fim

Set Value before Switchover		Set Value after Switchover	
MCM1		MCM1	
		0 (foco = fih)	1 (foco = fim)
0 (foco = fih)	fim ≥ fih		2 cycles
	fim < fih		2 fih/fim cycles
1 (foco = fim)	fim ≥ fih	2 fim/fih cycles	
	fim < fih	2 cycles	

Table 6 - 7 Maximum Number of Clock Cycles Required for fMAIN ↔ fSUB

Set Value before Switchover	Set Value after Switchover	
CSS	CSS	
	0 (fCLK = fMAIN)	1 (fCLK = fSUB)
0 (fCLK = fMAIN)		1 + 2 fMAIN/fSUB cycles
1 (fCLK = fSUB)	3 cycles	

Remark 1. The number of clock cycles listed in **Table 6 - 5**, **Table 6 - 6**, and **Table 6 - 7** is the number of cycles of the CPU clock before switchover.

Remark 2. Calculate the number of clock cycles in **Table 6 - 5**, **Table 6 - 6**, and **Table 6 - 7** by rounding up the number after the decimal position.

Example: When switching the main system clock from the high-speed on-chip oscillator clock (8 MHz selected) to the high-speed system clock (in this case, f_H = 8 MHz, f_M = 10 MHz)

$$1 + f_H/f_M = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2 \text{ cycles}$$

6.6.7 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

When stopping the clock, confirm the conditions before clock oscillation is stopped.

Table 6 - 8 Conditions Before the Clock Oscillation is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation is Stopped (External Clock Input is Disabled)	Flag Settings of the SFRs
High-speed on-chip oscillator clock	MCS1 = 1, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
Middle-speed on-chip oscillator clock	MCS1 = 0, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the middle-speed on-chip oscillator clock.)	MIOEN = 0
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating with a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating with a clock other than the subsystem clock.)	XTSTOP = 1
External subsystem clock		
Low-speed on-chip oscillator clock ^{Note}	CLS = 0 (The CPU is operating with a clock other than the low-speed on-chip oscillator clock.)	SELLOSC = 0 WUTMMCK0 = 0

Note The low-speed on-chip oscillator clock is not stopped while the WDT is operating or when f_{IL} is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

6.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants (reference values), see the target product page on the Renesas Web site.

Caution 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board.

Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

Caution 2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 6 - 23 Examples of External Circuits



Section 7 Timer Array Unit (TAU)

The number of units or channels of the timer array unit differs, depending on the product.

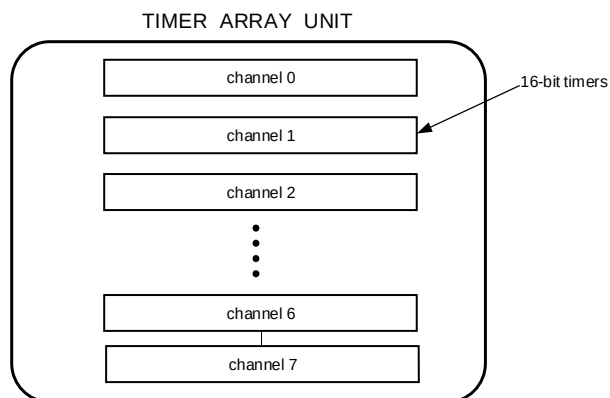
Units	Channels	16-pin	20-pin	24- and 25-pin	30-, 32-, 36-, 40-, 44-, and 48-pin
Unit 0	Channel 0	—	√	√	√
	Channel 1	—	√	√	√
	Channel 2	√	√	√	√
	Channel 3	—	—	√	√
	Channel 4	—	—	—	√
	Channel 5	—	—	—	√
	Channel 6	—	—	—	√
	Channel 7	—	—	—	√

Caution 1. The presence or absence of timer I/O pins depends on the product. See Table 7 - 2 Timer I/O Pins Provided in Each Product for details.

Caution 2. Most of the following descriptions in this section use the 48-pin products as an example.

The timer array unit has eight 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more channels can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
- Interval timer (→ refer to 7.8.1) - Square wave output (→ refer to 7.8.1) - External event counter (→ refer to 7.8.2) - DividerNote (→ refer to 7.8.3) - Input pulse interval measurement (→ refer to 7.8.4) - Measurement of high-/low-level width of input signal (→ refer to 7.8.5) - Delay counter (→ refer to 7.8.6)	- One-shot pulse output (→ refer to 7.9.1) - PWM output (→ refer to 7.9.2) - Multiple PWM output (→ refer to 7.9.3)

Note Only channel 0 of unit 0 supports this.

It is possible to use the 16-bit timer of channels 1 and 3 of unit 0 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Channel 7 of unit 0 can be used to realize LIN-bus communication operating in combination with UART2 of the serial array unit.

7.1 Functions of Timer Array Unit

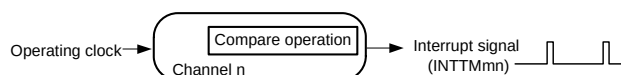
Timer array unit has the following functions.

7.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

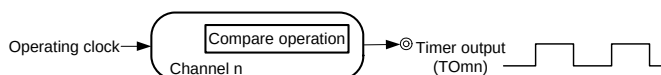
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



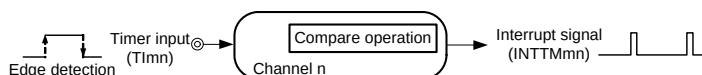
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOMn).



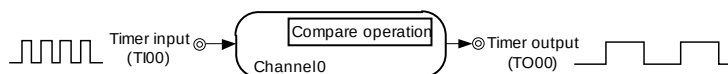
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TIMn) has reached a specific value.



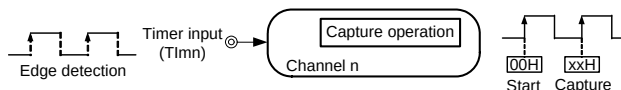
(4) Divider function (only channel 0 of unit 0)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



(5) Input pulse interval measurement

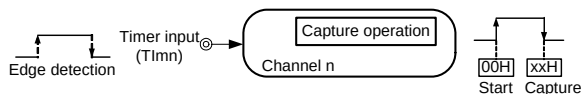
Counting is started by the valid edge of a pulse signal input to a timer input pin (TIMn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(Remarks are listed on the next page.)

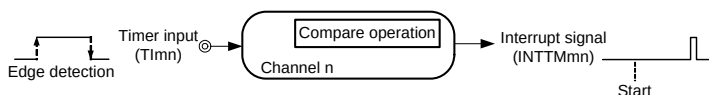
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

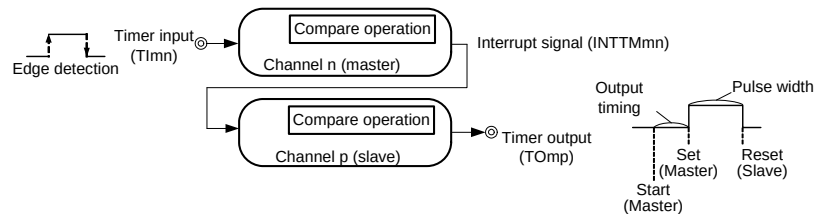
Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

7.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

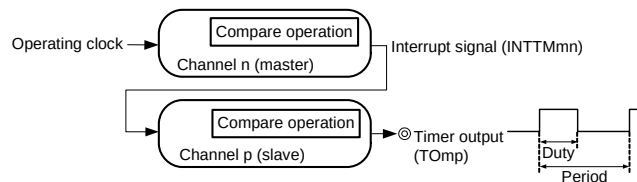
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



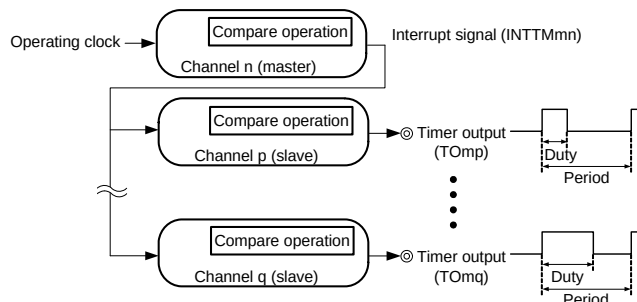
(2) PWM (Pulse Width Modulation) output

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.



Caution For details about the rules of simultaneous channel operation function, see 7.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7),
p, q: Slave channel number (n < p < q ≤ 7)

7.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution There are several rules for using 8-bit timer operation function.
For details, see 7.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

7.1.4 LIN-bus supporting function (channel 7 of unit 0 only)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

(2) Detection of break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.

(3) Measurement of pulse width of sync field

After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD2) of UART2 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

Remark For details about setting up the operations used to implement the LIN-bus, see 7.3.15 Input switch control register (ISC) and 7.8.5 Operation for input signal high-/low-level width measurement.

7.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 7 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07 ^{Note 1} , RxD2 pin (for LIN-bus)
Timer output	TO00 to TO07 pins ^{Note 1} , output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer input select register 1 (TIS1) • Timer output enable register m (TOEm) • Timer output register m (TOM) • Timer output level register m (TOLm) • Timer output mode register m (TOMm) <Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable register 1 (NFEN1) • Port mode control A registers (PMCAxx)^{Note 2} • Port mode control T registers (PMCTxx)^{Note 2} • Port mode registers (PMxx)^{Note 2} • Port registers (Pxx)^{Note 2}

Note 1. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

Note 2. The port mode control A registers (PMCAxx), port mode control T registers (PMCTxx), port mode registers (PMxx), and port registers (Pxx) to be set differ depending on the product. For details, see **4.5.4 Examples of register settings for port and alternate functions**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

Table 7 - 2 Timer I/O Pins Provided in Each Product

Timer array unit channels		I/O Pins of Each Product			
		30-, 32-, 36-, 40-, 44-, and 48-pin	24- and 25-pin	20-pin	16-pin
Unit 0	Channel 0	TI00/TO00			x
	Channel 1	TI01/TO01			x
	Channel 2	TI02/TO02			
	Channel 3	TI03/TO03		x	x
	Channel 4	TI04/TO04	x	x	x
	Channel 5	TI05/TO05	x	x	x
	Channel 6	TI06/TO06	x	x	x
	Channel 7	TI07/TO07	x	x	x

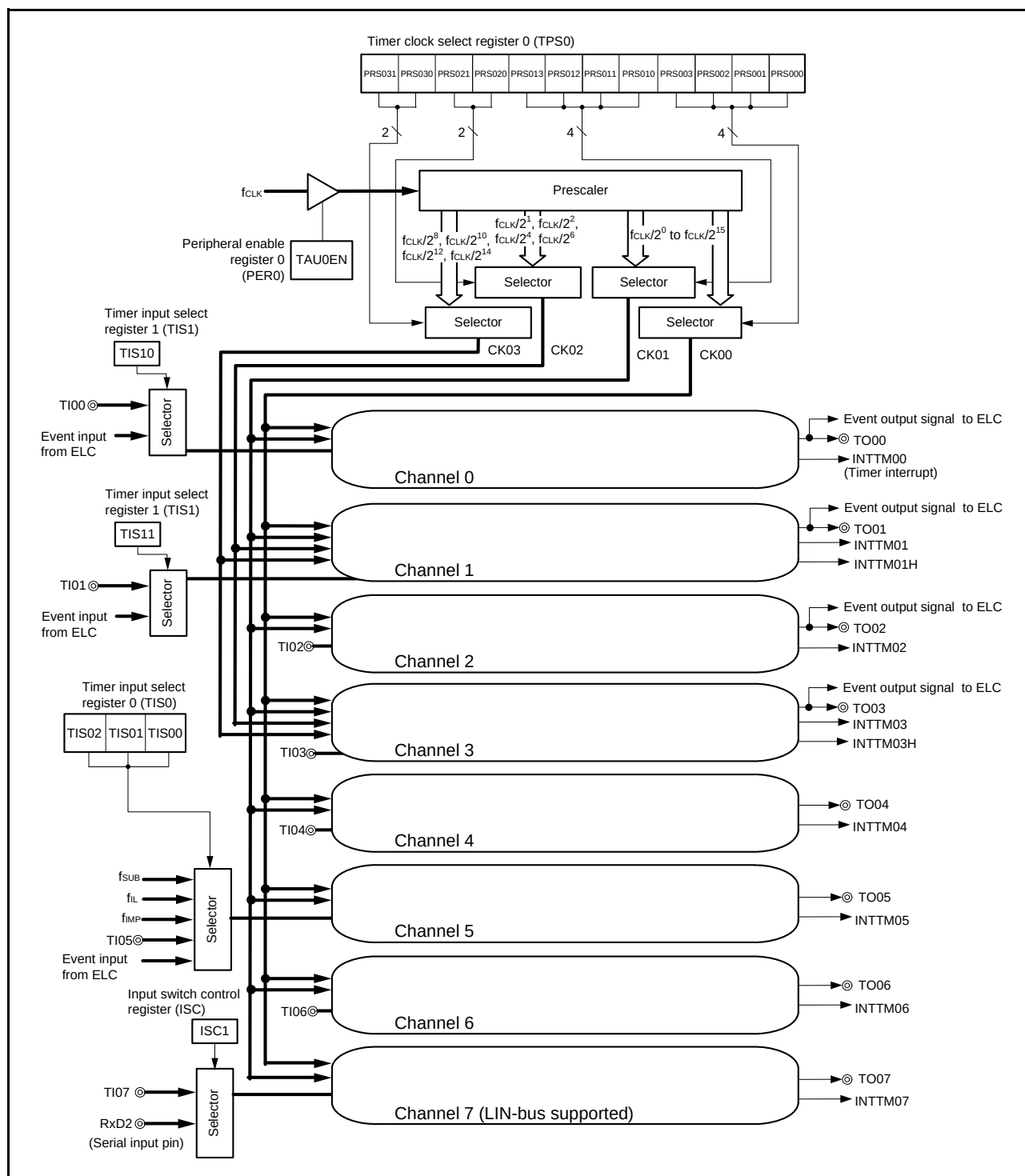
Remark 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Remark 2. x: The channel is not available.

Remark 3. Pins in the parentheses indicate an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to 1.

Figure 7 - 1 shows the block diagram of the timer array unit.

Figure 7 - 1 Entire Configuration of Timer Array Unit 0 (Example: 48-Pin Products)



Remark f_{SUB}: Subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency

Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

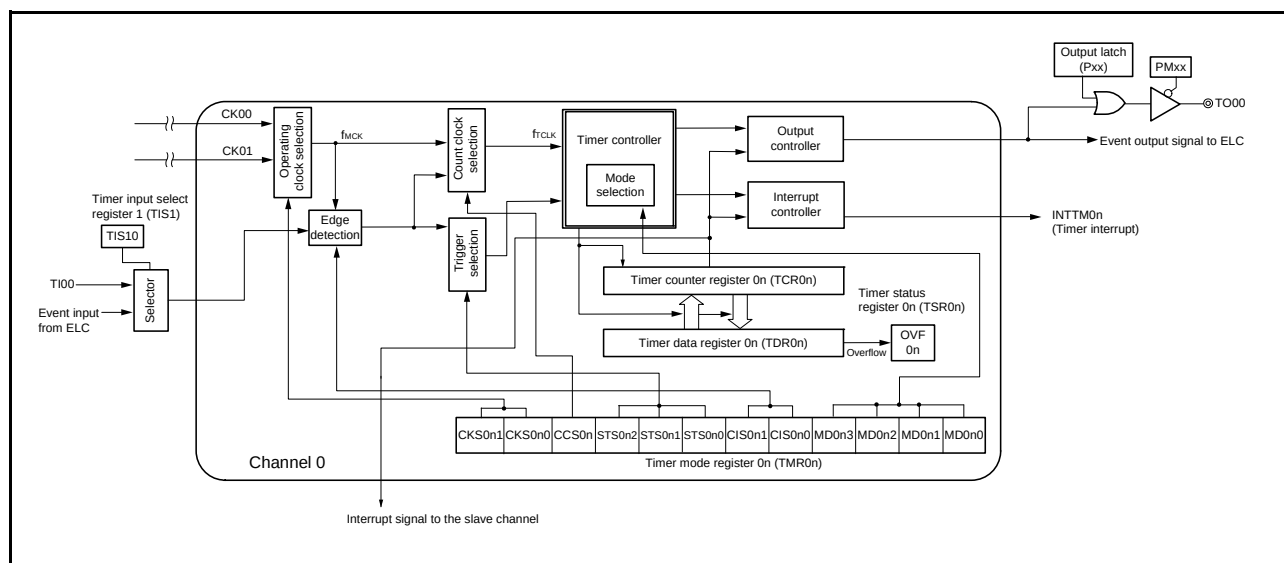


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

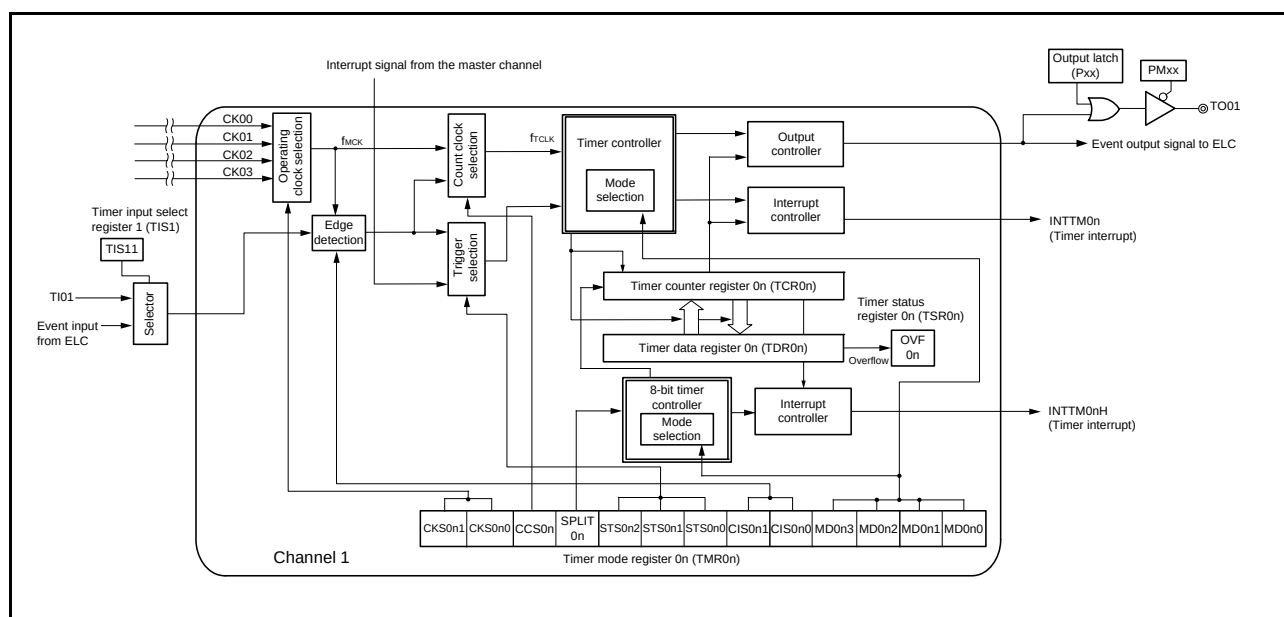
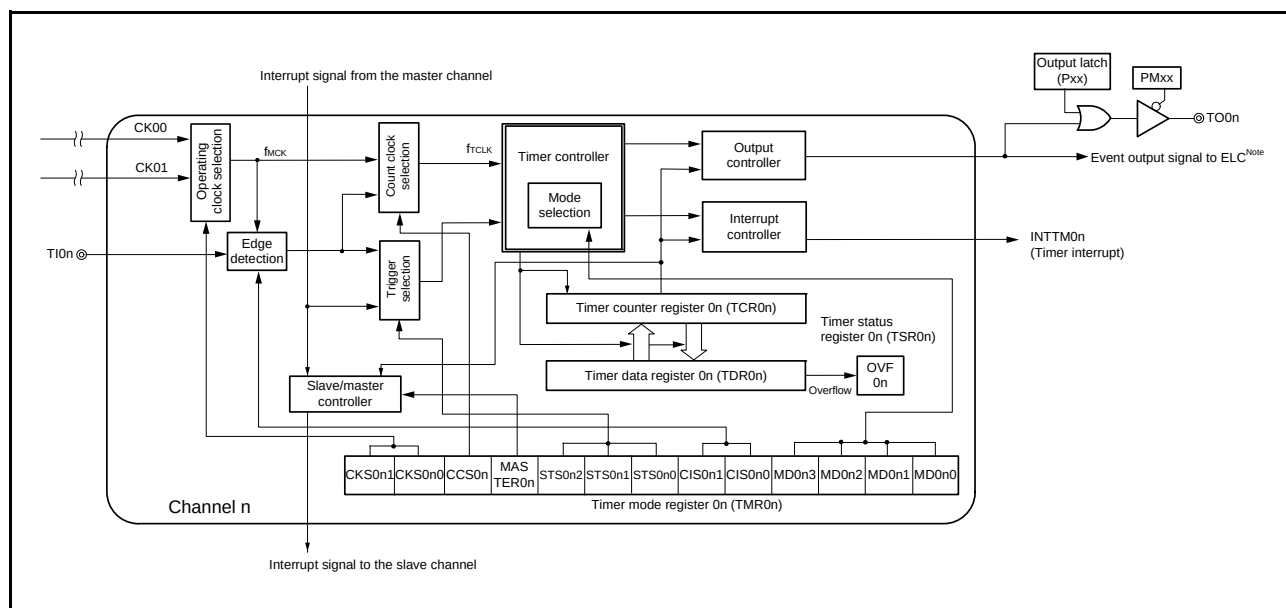


Figure 7 - 4 Internal Block Diagram of Channels 2, 4, and 6 of Timer Array Unit 0



Note Channel 2 only supports this signal.

Remark $n = 2, 4, 6$

Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0

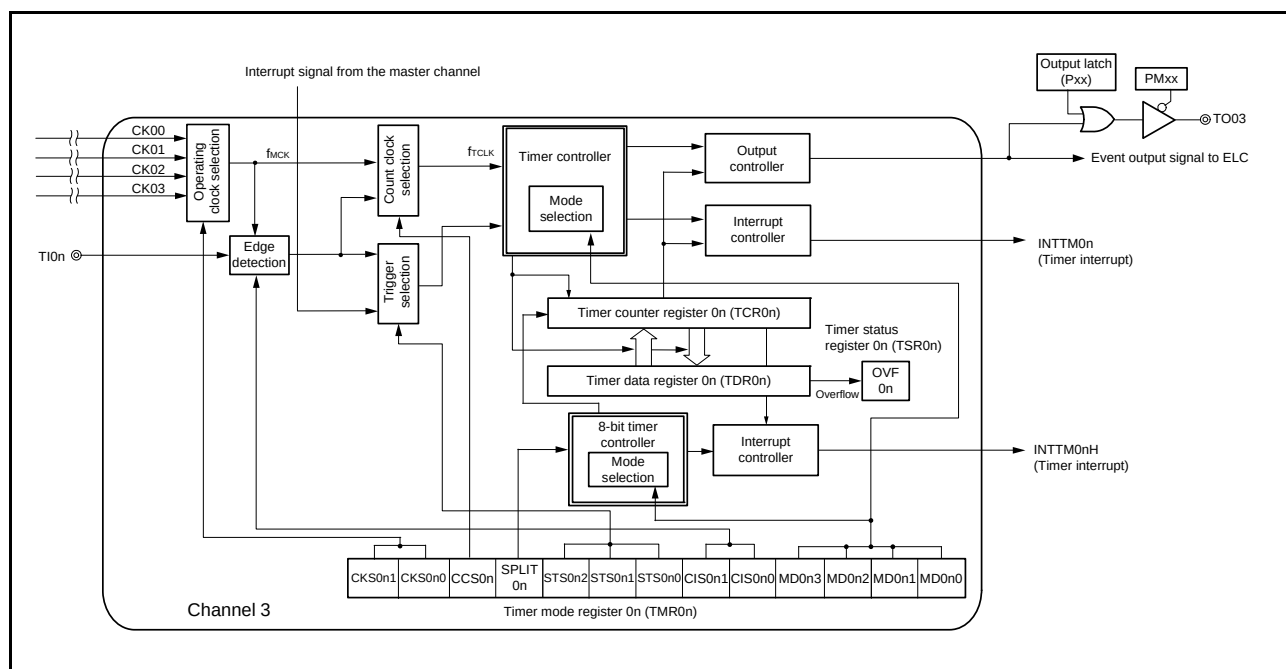


Figure 7 - 6 Internal Block Diagram of Channel 5 of Timer Array Unit 0

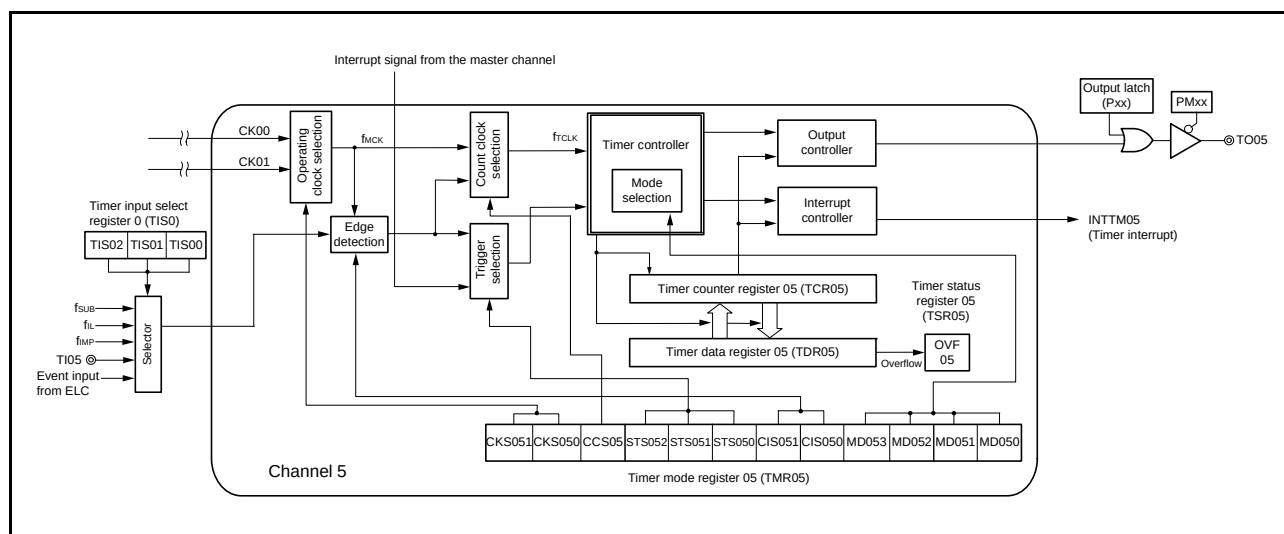
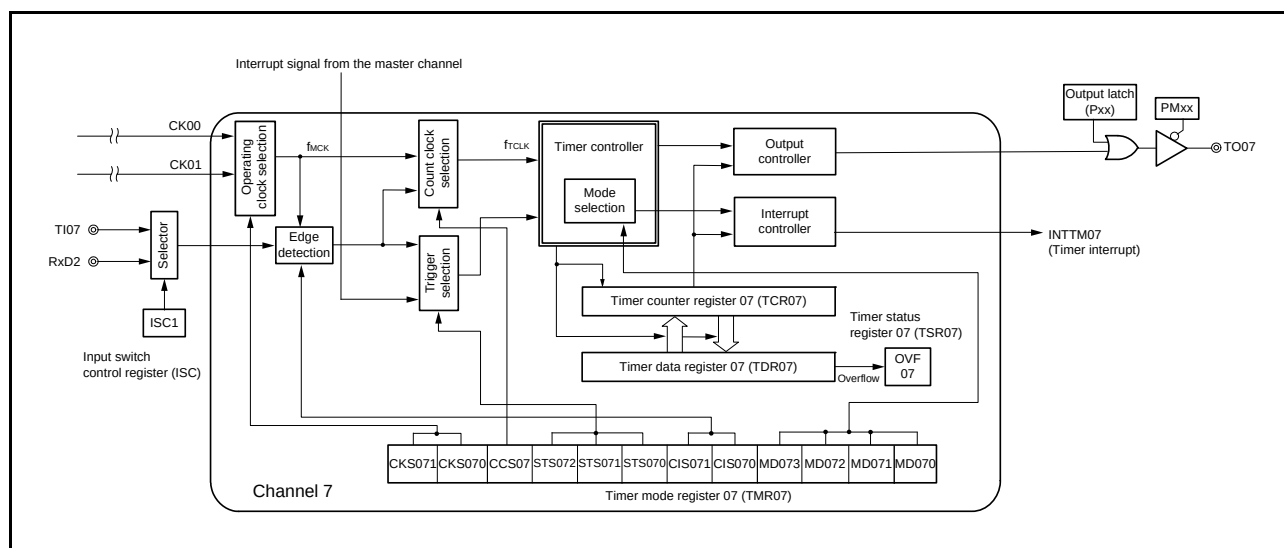


Figure 7 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit 0



7.2.1 Timer counter register mn (TCRmn)

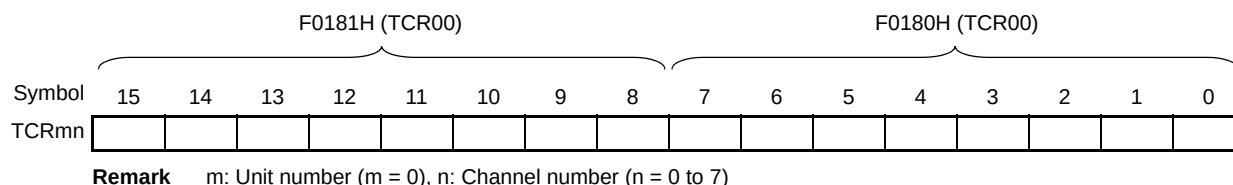
The TCRmn register is a 16-bit read-only register used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **7.3.4 Timer mode register mn (TMRmn)**).

Figure 7 - 8 Format of Timer Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07)
After reset: FFFFH
R/W: R



The count value can be read by reading timer counter register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmRES bit of peripheral reset control register 0 (PRR0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The value read from the TCRmn register varies depending on the change to the operation mode and operating state as shown in the table below.

Table 7 - 3 Timer Counter Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Value Read from the Timer Counter Register mn (TCRmn) ^{Note}			
		Value when the operation mode is changed after releasing reset	Value when count operation is temporarily stopped (TTmn = 1)	Value when the operation mode is changed after count operation was temporarily stopped (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
Capture mode	Count-up	0000H	Value when counting is stopped	Undefined	—
Event counter mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
One-count mode	Countdown	FFFFH	Value when counting is stopped	Undefined	FFFFH
Capture & one-count mode	Count-up	0000H	Value when counting is stopped	Undefined	Captured value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLITm1, SPLITm3 bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

The value of this register following a reset is 0000H.

Figure 7 - 9 Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07)
After reset: 0000H
R/W: R/W

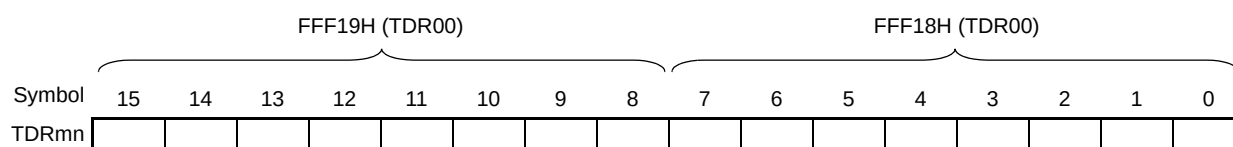
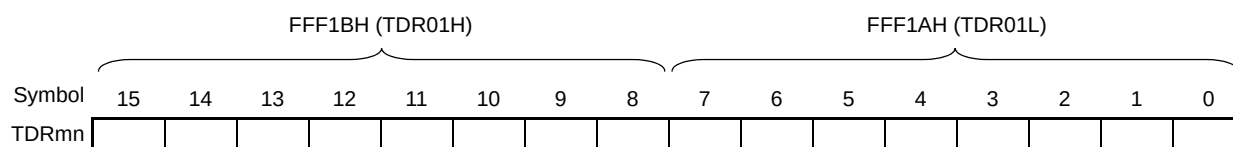


Figure 7 - 10 Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03)
After reset: 0000H
R/W: R/W



(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer counter register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input. A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3 Registers for Controlling the Timer Array Unit

The following registers are used to control the timer array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSM)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer input select register 1 (TIS1)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode registers (PMxx)
- Port registers (Pxx)

Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If timer array unit 0 is to be used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 7 - 11 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN
	TAU0EN	Control of supply of an input clock to timer array unit 0						
	0	Stops supply of an input clock. • The SFRs used by timer array unit 0 cannot be written. • When an SFR used by timer array unit 0 is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by timer array unit 0 can be read and written.						

Caution 1. When setting the timer array unit, start by setting the following registers while the setting of the TAU0EN bit is 1.

- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)

If the setting of the TAU0EN bit is 0, each register which controls the timer array unit is restored to 00H and writing to any of those registers is ignored. Note, however, writing to the following registers is valid.

- Timer input select registers 0, 1 (TIS0, TIS1)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode control A register 0 (PMCA0)
- Port mode control T registers 0, 3 (PMCT0, PMCT3)
- Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)
- Port registers 0, 1, 3, 4 (P0, P1, P3, P4)

Caution 2. The functions that are mounted depend on the product. For details on the PER0 register, see the description in Section 6 Clock Generator.

7.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

Set bit 0 (TAU0RES) of this register to 1 to place timer array unit 0 in the reset state.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 7 - 12 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H

After reset: 00H

R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

TAU0RES	Control resetting of timer array unit 0
0	Timer array unit 0 is released from the reset state.
1	Timer array unit 0 is in the reset state. • The SFRs for use with timer array unit 0 are initialized.

Caution The functions that are mounted depend on the product. For details on the PRR0 register, see the description in Section 21 Reset Function.

7.3.3 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register used to select two types or four types of operating clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register.

Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten ($n = 0$ to 7):

All channels for which CKm0 is selected as the operating clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten ($n = 0$ to 7):

All channels for which CKm1 is selected as the operating clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten ($n = 1, 3$):

All channels for which CKm2 is selected as the operating clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).

If the PRSm30 and PRSm31 bits can be rewritten ($n = 1, 3$):

All channels for which CKm3 is selected as the operating clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

The value of this register following a reset is 0000H.

Address: F01B6H, F01B7H (TPS0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20
	7	6	5	4	3	2	1	0
	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operating clock (CKmk) ^{Note} (k = 0, 1)					
				fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz	
0	0	0	0	fCLK	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	fCLK/2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	fCLK/2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	fCLK/2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	fCLK/2 ⁵	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	fCLK/2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	fCLK/2 ⁷	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	fCLK/2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	fCLK/2 ⁹	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	fCLK/2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	fCLK/2 ¹¹	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	fCLK/2 ¹²	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	fCLK/2 ¹³	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	fCLK/2 ¹⁴	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	fCLK/2 ¹⁵	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock (fMCK) or the valid edge of the signal input from the TImn pin is selected.

Caution 1. Be sure to clear bits 15, 14, 11, 10 to 0.

Caution 2. If fCLK (undivided) is selected as the operating clock (CKm_k) and TDR_nm is set to 0000H (n = 0, m = 0 to 7), interrupt requests output from the timer array unit cannot be used.

Remark 1. fCLK: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPSM register which becomes high level for one period of fCLK from its rising edge (m = 0). For details, see **7.5.1 Count clock (fCLK)**.

Figure 7 - 13 Format of Timer Clock Select Register m (TPSm) (2/2)

PRSm21	PRSm20		Selection of operating clock (CKm2) ^{Note}				
			fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz
0	0	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	1	fCLK/2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
1	0	fCLK/2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
1	1	fCLK/2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz

PRSm31	PRSm30		Selection of operating clock (CKm3) ^{Note}				
			fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz
0	0	fCLK/2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
0	1	fCLK/2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	fCLK/2 ¹²	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	fCLK/2 ¹⁴	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock (fMCK) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, 10 to 0.

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operating clock, the interval times shown in **Table 7 - 4** can be achieved by using the interval timer function.

Table 7 - 4 Interval Times Available for Operating Clock CKSm2 or CKSm3

Clock		Interval time ^{Note} (fCLK = 32 MHz)			
		10 μ s	100 μ s	1 ms	10 ms
CKm2	fCLK/2	√	—	—	—
	fCLK/2 ²	√	—	—	—
	fCLK/2 ⁴	√	√	—	—
	fCLK/2 ⁶	√	√	—	—
CKm3	fCLK/2 ⁸	—	√	√	—
	fCLK/2 ¹⁰	—	√	√	—
	fCLK/2 ¹²	—	—	√	√
	fCLK/2 ¹⁴	—	—	√	√

Note The margin is within 5%.

Remark 1. fCLK: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of fCLK/2ⁱ selected with the TPSPm register, see **7.5.1 Count clock (fCLK)**.

7.3.4 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operating clock (fMCK), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEMn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEMn = 1). For details, see **7.8 Independent Channel Operation Function of Timer Array Unit** and **7.9 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMRmn register can be set by a 16-bit memory manipulation instruction.

The value of this register following a reset is 0000H.

Caution The bit function assigned to bit 11 of the TMRmn register depends on the channel.
TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0Note 1	STSmn2	STSmn1	STSmn0

7	6	5	4	3	2	1	0
CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operating clock (fmck) of channel n
0	0	Operating clock CKm0 set by timer clock select register m (TPSm)
0	1	Operating clock CKm2 set by timer clock select register m (TPSm)
1	0	Operating clock CKm1 set by timer clock select register m (TPSm)
1	1	Operating clock CKm3 set by timer clock select register m (TPSm)
Operating clock (fmck) is used by the edge detector. A count clock (ftCLK) and a sampling clock are generated depending on the setting of the CCSmn bit. The operating clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of count clock (ftCLK) of channel n
0	Operating clock (fmck) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TIMn pin • In the case of unit 0: In channels 0 to 4, valid edge of input signal selected by TIS1n In channel 5, valid edge of input signal selected by TIS0 In channel 7, valid edge of input signal selected by ISC
Count clock (ftCLK) is used for the counter, output controller, and interrupt controller.	

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (2/3)

(Bit 11 of TMRmn (n = 2, 4, 6))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2, 4, 6 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0, 5, 7 are fixed to 0 (Regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

CISmn1	CISmn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (3/3)

MD mn3	MD mn2	MD mn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter/One-shot pulse output/PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low- level width of input signal	Counting up
Other than above			Setting prohibited		
The operation in each mode varies depending on the MDmn0 bit. See the table below for details.					

Operation mode set by the MDmn3 to MDmn1 bits (see the table above)	MDmn0	Setting of starting counting and interrupt
- Interval timer mode (0, 0, 0) - Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
One-count mode^{Note 2} (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation ^{Note 3} . At that time, interrupt is not generated.
Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated.
Other than above		Setting prohibited

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Note 2. In one-count mode, interrupt output (INTTmn) when starting a count operation and Tmn output are not controlled.

Note 3. If the start trigger (Tsmn = 1) is issued during operation, the counter is initialized, and recounting is started (does not occur the interrupt request).

Caution 1. Be sure to clear bits 13, 5, and 4 to 0.

Caution 2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for fCLK is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (fmck) or the valid edge of the signal input from the TImn pin is selected as the count clock (fCLK).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.5 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow state of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 7 - 5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The 8 lower-order bits of a TSRmn register can be handled as TSRmnL, which can be read by an 8-bit memory manipulation instruction.

The value of this register following a reset is 0000H.

Figure 7 - 15 Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
TSRmn	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	OVF
OVF	Counter overflow state of channel n							
0	Overflow does not occur.							
1	Overflow occurs.							
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.								

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Table 7 - 5 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	— (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

7.3.6 Timer channel enable status register m (TE_m)

The TE_m register is used to enable or stop the timer operation of each channel.

Each bit of the TE_m register corresponds to each bit of the timer channel start register m (TS_m) and the timer channel stop register m (TT_m). When a bit of the TS_m register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TT_m register is set to 1, the corresponding bit of this register is cleared to 0.

The TE_m register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TE_m register can be set with a 1-bit or 8-bit memory manipulation instruction with TE_mL.

The value of this register following a reset is 0000H.

Figure 7 - 16 Format of Timer Channel Enable Status Register m (TE_m)

Address: F01B0H, F01B1H (TE₀)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
TE _m	0	0	0	0	TEH _m 3	0	TEH _m 1	0
	7	6	5	4	3	2	1	0
	TE _m 7	TE _m 6	TE _m 5	TE _m 4	TE _m 3	TE _m 2	TE _m 1	TE _m 0
TEH _m 3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode							
0	Operation is stopped.							
1	Operation is enabled.							
TEH _m 1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode							
0	Operation is stopped.							
1	Operation is enabled.							
TE _m _n	Indication of operation enabled or stopped state of channel n							
0	Operation is stopped.							
1	Operation is enabled.							
This bit displays whether operation of the lower 8-bit timer for TE _m 1 and TE _m 3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.								

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.7 Timer channel start register m (TSM)

The TSM register is a trigger register that is used to initialize timer counter register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEM) is set to 1.

The TSMn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEMn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSM register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSM register can be set with a 1-bit or 8-bit memory manipulation instruction with TSM_L.

The value of this register following a reset is 0000H.

Figure 7 - 17 Format of Timer Channel Start Register m (TSM)

Address: F01B2H, F01B3H (TS0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TSM	0	0	0	0	TSHm3	0	TSHm1	0
	7	6	5	4	3	2	1	0
	TSM7	TSM6	TSM5	TSM4	TSM3	TSM2	TSM1	TSM0
TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode							
0	No trigger operation							
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see Table 7 - 6 in 7.5.2 Timing of the start of counting).							
TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode							
0	No trigger operation							
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see Table 7 - 6 in 7.5.2 Timing of the start of counting).							
TSMn	Operation enable (start) trigger of channel n							
0	No trigger operation							
1	The TEMn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see Table 7 - 6 in 7.5.2 Timing of the start of counting). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSM1 and TSM3 when channel 1 or 3 is in the 8-bit timer mode.							

Caution 1. Be sure to clear bits 15 to 12, 10, and 8 to 0.

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSMn (TSHm1, TSHm3) bit is set to 1.

When the TImn pin noise filter is enabled (TNFENnm = 1): Four cycles of the operating clock (fMCK)

When the TImn pin noise filter is disabled (TNFENnm = 0): Two cycles of the operating clock (fMCK)

Remark 1. When the TSM register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.8 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TEHm1, TEHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

The value of this register following a reset is 0000H.

Figure 7 - 18 Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H (TT0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TTm	0	0	0	0	TTHm3	0	TTHm1	0
	7	6	5	4	3	2	1	0
	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0
TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode							
0	No trigger operation							
1	TEHm3 bit is cleared to 0 and the count operation is stopped.							
TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode							
0	No trigger operation							
1	TEHm1 bit is cleared to 0 and the count operation is stopped.							
TTmn	Operation stop trigger of channel n							
0	No trigger operation							
1	TEmn bit is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.							

Caution Be sure to clear bits 15 to 12, 10, and 8 of the TTm register to 0.

Remark 1. When the TTm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.9 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 5 of unit 0 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 7 - 19 Format of Timer Input Select Register 0 (TIS0)

Address: F0074H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELC
0	1	1	Middle-speed on-chip oscillator peripheral clock (fMCP)
1	0	0	Low-speed on-chip oscillator clock (fIL)
1	0	1	Subsystem clock (fSUB)
Other than above			Setting prohibited

Caution Make sure that both the high-level and low-level widths of timer input to be selected are no less than $1/f_{MCK} + 10 \text{ ns}$.
Therefore, when selecting fSUB as fCLK (the CSS bit of the CKC register = 1), the TIS02 bit cannot be set to 1.

7.3.10 Timer input select register 1 (TIS1)

The TIS1 register is used to select channels 0 and 1 of unit 0 timer input.

The TIS1 register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 7 - 20 Format of Timer Input Select Register 1 (TIS1)

Address: F0075H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	0	0	0	0	TIS11	TIS10

TIS1n	Selection of timer input used with channel n
0	Timer input pin (TI0n)
1	Event input signal from ELC

Remark n = 0, 1

7.3.11 Timer output enable register m (TOEm)

The TOEm register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOMn bit of timer output register m (TOM) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOMn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

The value of this register following a reset is 0000H.

Figure 7 - 21 Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH (TOE0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOEm	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Enabling/disabling timer output for channel n
0	Disables timer output. The corresponding TOMn bit does not reflect timer operation with this setting, so the output level of a TOMn bit is fixed to the level written to the TOM register. Writing to the TOMn bit is enabled and the level set in the TOMn bit is output from the TOMn pin.
1	Enables timer output. The corresponding TOMn bit reflects timer operation with this setting, so the output waveform is generated. Writing to the TOMn bit is ignored.

Caution Be sure to clear bits 15 to 8 to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.12 Timer output register m (TOM)

The TOM register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOMn) of each channel.

The TOMn bit on this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P00/TI00, P01/TO00, P16/TI01/TO01, P17/TI02/TO02, P31/TI03/TO03, P13/TI04/TO04, P12/TI05/TO05, P11/TI06/TO06, or P41/TI07/TO07 pin as a port function pin, set the corresponding TOMn bit to 0.

The TOM register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOM register can be set with an 8-bit memory manipulation instruction with TOML.

The value of this register following a reset is 0000H.

Figure 7 - 22 Format of Timer Output Register m (TOM)

Address: F01B8H, F01B9H (TO0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOM	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TOM7	TOM6	TOM5	TOM4	TOM3	TOM2	TOM1	TOM0

TOMn	Timer output of channel n
0	Timer output value is 0.
1	Timer output value is 1.

Caution Be sure to clear bits 15 to 8 to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.13 Timer output level register m (TOLm)

The TOLm register controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

The value of this register following a reset is 0000H.

Figure 7 - 23 Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH (TOL0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOLm	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0
TOLmn	Control of timer output level of channel n							
0	Positive logic output (active-high)							
1	Negative logic output (active-low)							

Caution Be sure to clear bits 15 to 8, and 0 to 0.

Remark 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.3.14 Timer output mode register m (TOMm)

The TOMm register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.

The value of this register following a reset is 0000H.

Figure 7 - 24 Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH (TOM0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOMm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TOMm7	TOMm6	TOMm5	TOMm4	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggled output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear bits 15 to 8, and 0 to 0.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7
(For details of the relation between the master channel and slave channel, refer to **7.4.1 Basic rules of simultaneous channel operation function.**)

7.3.15 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 7 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD2) is selected as a timer input signal.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 7 - 25 Format of Input Switch Control Register (ISC)

Address: F0073H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0
ISC1								
Switching channel 7 input of timer array unit								
0								
Uses the input signal of the TI07 pin as a timer input (normal operation).								
1								
Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).								
ISC0								
Switching external interrupt (INTP0) input								
0								
Uses the input signal of the INTP0 pin as an external interrupt (normal operation).								
1								
Uses the input signal of the RxD2 pin as an external interrupt (wakeup signal detection).								

Remark When the LIN-bus communication function is used, select the input signal of the RxD2 pin by setting ISC1 to 1.

7.3.16 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel^{Note}.

The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Note For details, see 7.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1), 7.5.2 Timing of the start of counting, and 7.7 Timer Input (TImn) Control.

Figure 7 - 26 Format of Noise Filter Enable Register 1 (NFEN1) (1/2)

Address: F0071H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00
	TNFEN07	Enabling/disabling use of the noise filter for the TI07 pin ^{Note}						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						
	TNFEN06	Enabling/disabling use of the noise filter for the TI06 pin						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						
	TNFEN05	Enabling/disabling use of the noise filter for the TI05 pin						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						
	TNFEN04	Enabling/disabling use of the noise filter for the TI04 pin						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						
	TNFEN03	Enabling/disabling use of the noise filter for the TI03 pin						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						
	TNFEN02	Enabling/disabling use of the noise filter for the TI02 pin						
	0	Turns the noise filter off.						
	1	Turns the noise filter on.						

Figure 7 - 26 Format of Noise Filter Enable Register 1 (NFEN1) (2/2)

TNFEN01	Enabling/disabling use of the noise filter for the TI01 pin
0	Turns the noise filter off.
1	Turns the noise filter on.

TNFEN00	Enabling/disabling use of the noise filter for the TI00 pin
0	Turns the noise filter off.
1	Turns the noise filter on.

Note The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD2 pin can be selected.

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

7.3.17 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), port mode control A register (PMCAxx), and port mode control T register (PMCTxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.7 Port mode control A registers (PMCAxx)**, and **4.3.8 Port mode control T registers (PMCTxx)**.

The port mode register (PMxx), port register (Pxx), port mode control A register (PMCAxx), and port mode control T register (PMCTxx) to be set depend on the product. For details, see **4.5.4 Examples of register settings for port and alternate functions**.

When using the ports (such as P00/TI00 and P01/TO00) to be shared with the timer output pin for timer output, set the corresponding bits in the port mode control A register (PMCAxx), port mode control T register (PMCTxx), port mode register (PMxx), and port register (Pxx) to 0.

Example: When using P01/TO00 for timer output
Set the PMCT01 bit of port mode control T register 0 to 0.
Set the PM01 bit of port mode register 0 to 0.
Set the P01 bit of port register 0 to 0.

Remark The above statements apply to 36- to 48-pin products.

When using the ports (such as P00/TI00) to be shared with the timer input pin for timer input, set the corresponding bit in the port mode register (PMxx) to 1. And set the corresponding bits in the port mode control A register (PMCAxx) and port mode control T register (PMCTxx) to 0. At this time, the bit in the port register (Pxx) may be 0 or 1.

Example: When using P00/TI00 for timer input
Set the PMCT01 bit of port mode control T register 0 to 0.
Set the PM00 bit of port mode register 0 to 1.
Set the P00 bit of port register 0 to 0 or 1.

Remark The above statements apply to 36- to 48-pin products.

7.4 Basic Rules of Timer Array Unit

7.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

- (1) Only an even channel (channels 0, 2, 4, etc.) can be set as a master channel.
- (2) Any channel, except channel 0, can be set as a slave channel.
- (3) The slave channel must be lower than the master channel.

Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

- (4) Two or more slave channels can be set for one master channel.
- (5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.

Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

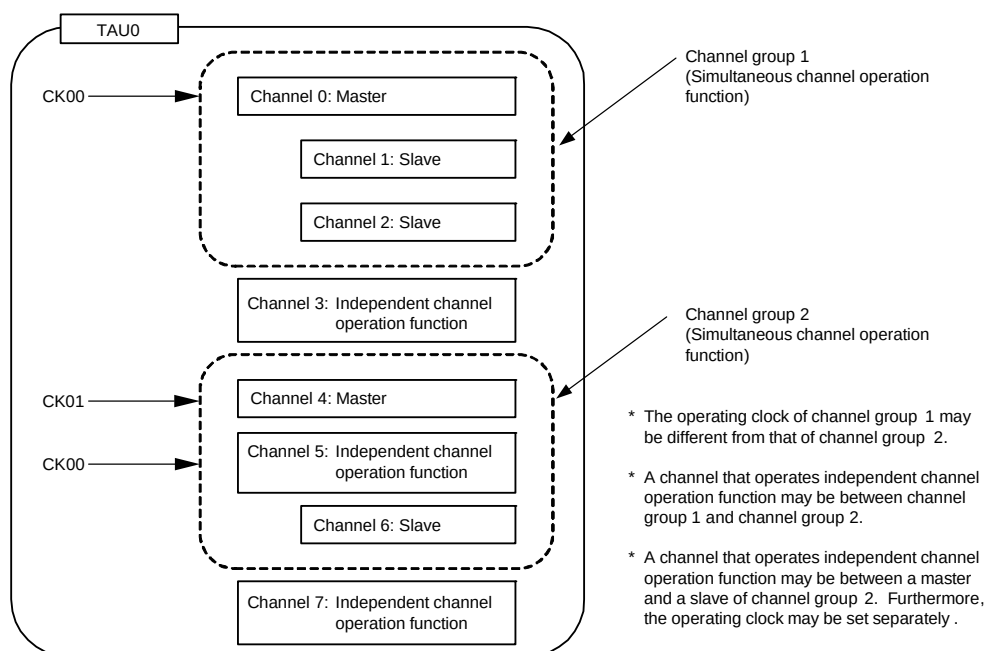
- (6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
- (7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
- (8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
- (9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
- (10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
- (11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
- (12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
- (13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
- (14) Timer mode register m0 (TMRm0) has no master bit (it is fixed to 0). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **7.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Example



7.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

- (1) The 8-bit timer operation function applies only to channels 1 and 3.
- (2) When using 8-bit timers, set the `SPLITmn` bit of timer mode register `mn` (`TMRmn`) to 1.
- (3) The higher 8 bits can be operated as the interval timer function.
- (4) At the start of operation, the higher 8 bits output `INTTMm1H/INTTMm3H` (an interrupt) (which is the same operation performed when `MDmn0` is set to 1).
- (5) The operating clock of the higher 8 bits is selected according to the `CKSmn1` and `CKSmn0` bits of the lower-bit `TMRmn` register.
- (6) For the higher 8 bits, the `TSHm1/TSHm3` bit is manipulated to start channel operation and the `TTHm1/TTHm3` bit is manipulated to stop channel operation. The channel state can be checked using the `TEHm1/TEHm3` bit.
- (7) The lower 8 bits operate according to the `TMRmn` register settings. The following three functions support operation of the lower 8 bits:
 - Interval timer function/square wave output function
 - External event counter function
 - Delay count function
- (8) For the lower 8 bits, the `TSm1/TSm3` bit is manipulated to start channel operation and the `TTm1/TTm3` bit is manipulated to stop channel operation. The channel state can be checked using the `TEm1/TEm3` bit.
- (9) During 16-bit operation, manipulating the `TSHm1`, `TSHm3`, `TTHm1`, and `TTHm3` bits is invalid. The `TSm1`, `TSm3`, `TTm1`, and `TTm3` bits are manipulated to operate channels 1 and 3. The `TEHm3` and `TEHm1` bits are not changed.
- (10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark `m`: Unit number (`m` = 0), `n`: Channel number (`n` = 1, 3)

7.5 Operations of Counters

7.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operating clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

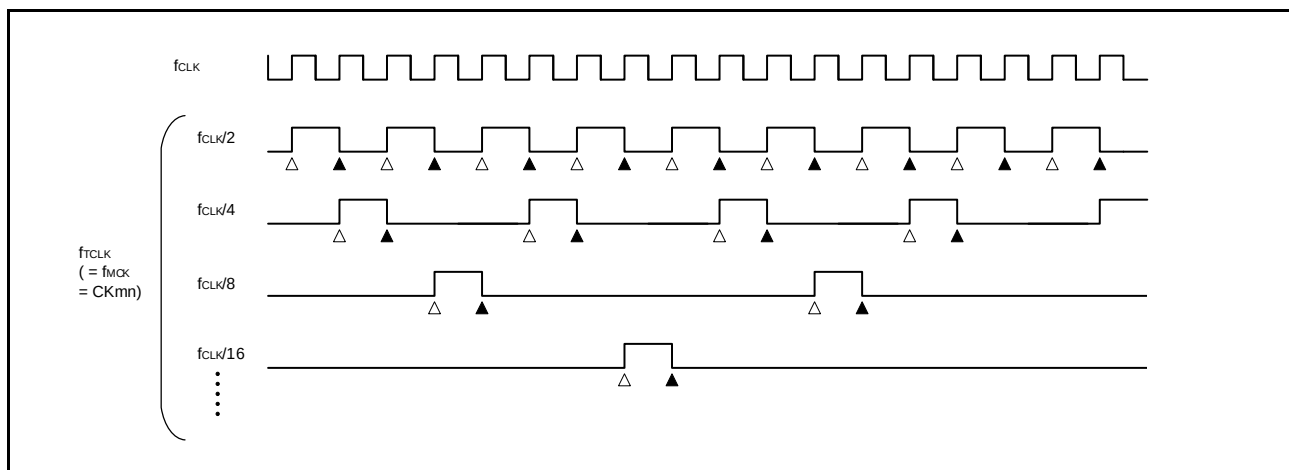
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operating clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (f_{TCLK}) is between f_{CLK} to f_{CLK}/2¹⁵ by setting of timer clock select register m (TPSm). When a divided f_{CLK} is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, fixed to high level

Counting of timer counter register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at rising edge of the count clock", as a matter of convenience.

Figure 7 - 27 Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCSmn = 0)



Remark 1. Δ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

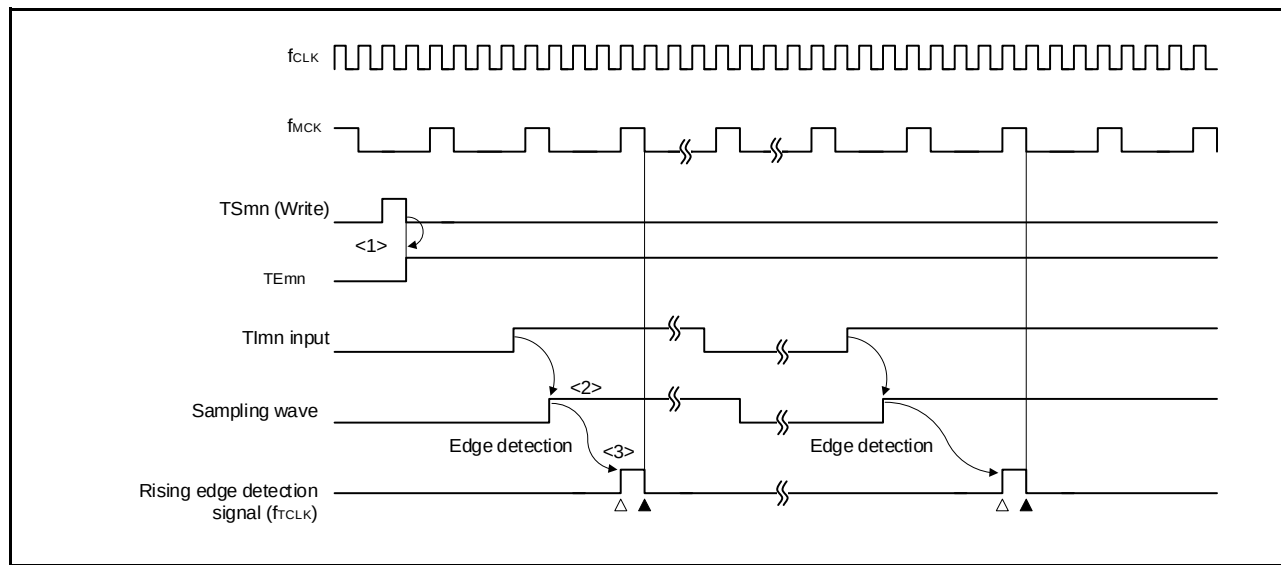
Remark 2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{CLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{CLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer counter register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at valid edge of input signal via the TImn pin", as a matter of convenience.

Figure 7 - 28 Timing of f_{CLK} and Count Clock (f_{CLK}) (When CCSmn = 1, Noise Filter Unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.

<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remark 1. Δ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock

f_{MCK}: Operating clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, the one-shot pulse output are the same.

7.5.2 Timing of the start of counting

Timer counter register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSM).

Operations from count operation enabled state to timer counter register mn (TCRmn) count start is shown in **Table 7 - 6**.

Table 7 - 6 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see 7.5.3 (1) Operation in interval timer mode).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation (see 7.5.3 (2) Operation in event counter mode).
• Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see 7.5.3 (3) Operation in capture mode (input pulse interval measurement)).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see 7.5.3 (4) Operation in one-count mode).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see 7.5.3 (5) Operation in capture & one-count mode (high-level width measurement)).

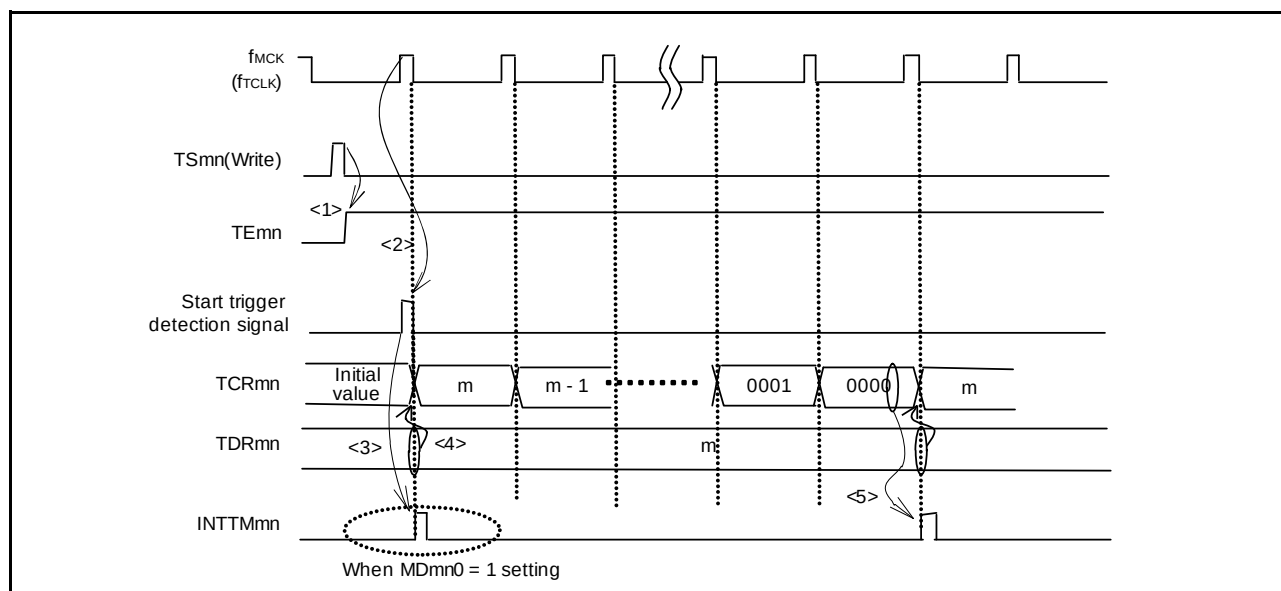
7.5.3 Operations of Counters

Here, the counter operation in each mode is explained.

(1) Operation in interval timer mode

- <1> Operation is enabled ($TE_{mn} = 1$) by writing 1 to the TS_{mn} bit. Timer counter register mn (TCR_{mn}) holds the initial value until count clock generation.
- <2> A start trigger is generated at the first count clock after operation is enabled.
- <3> When the MD_{mn0} bit is set to 1, $INTTM_{mn}$ is generated by the start trigger.
- <4> By the first count clock after the operation enable, the value of timer data register mn (TDR_{mn}) is loaded to the TCR_{mn} register and counting starts in the interval timer mode.
- <5> When the TCR_{mn} register counts down and its count value is 0000H, $INTTM_{mn}$ is generated and the value of timer data register mn (TDR_{mn}) is loaded to the TCR_{mn} register and counting keeps on.

Figure 7 - 29 Timing during Operation in Interval Timer Mode



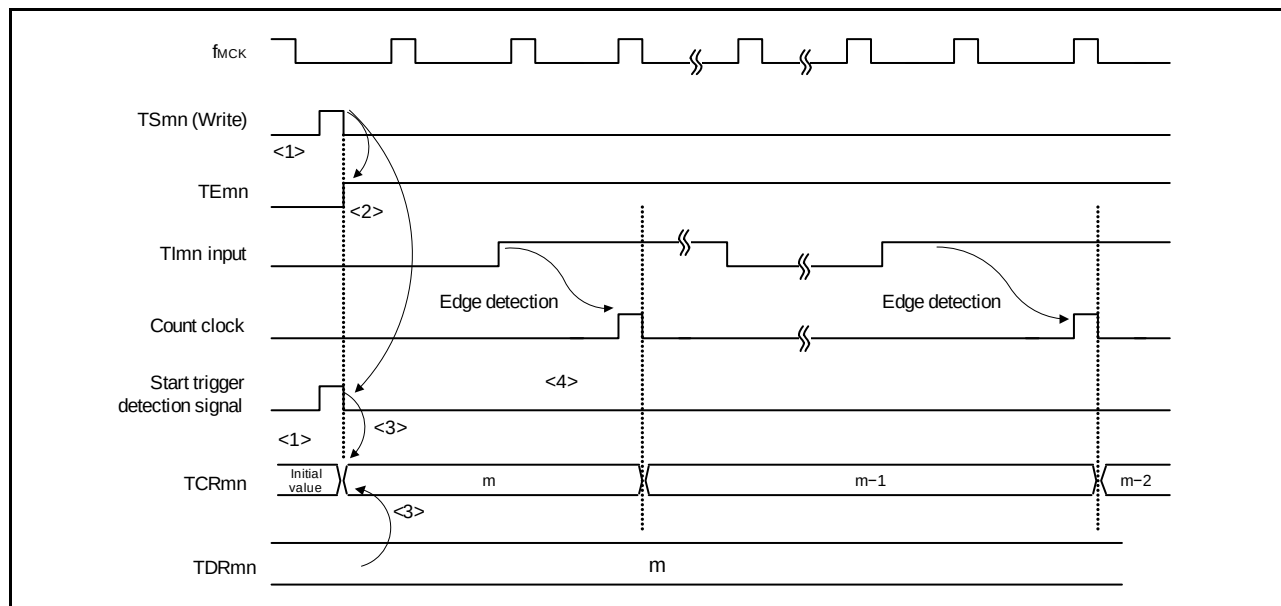
Caution In the first cycle operation of count clock after writing the TS_{mn} bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting $MD_{mn0} = 1$.

Remark f_{MCK} , the start trigger detection signal, and $INTTM_{mn}$ become active between one clock in synchronization with f_{CLK} .

(2) Operation in event counter mode

- <1> Timer counter register mn (TCRmn) holds its initial value while operation is stopped ($TE_{mn} = 0$).
- <2> Operation is enabled ($TE_{mn} = 1$) by writing 1 to the TS_{mn} bit.
- <3> As soon as 1 has been written to the TS_{mn} bit and 1 has been set to the TE_{mn} bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.
- <4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TI_{mn} input.

Figure 7 - 30 Timing during Operation in Event Counter Mode

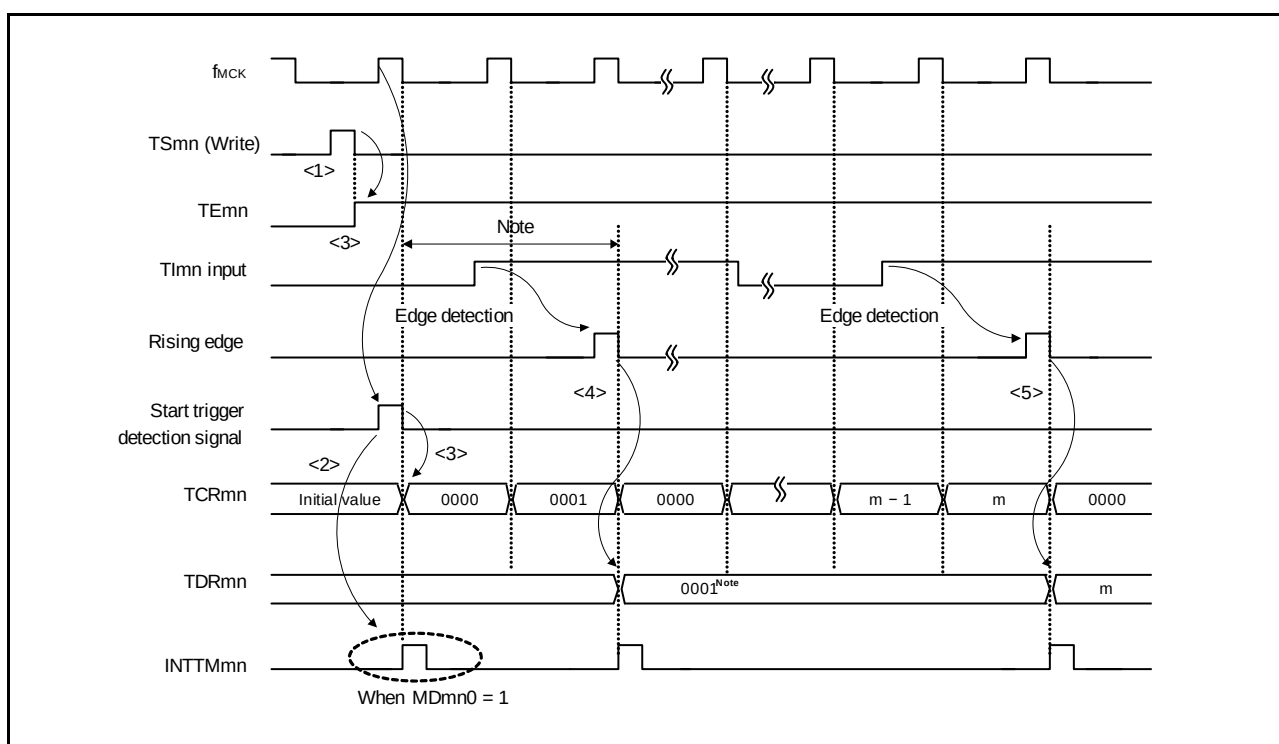


Remark Figure 7 - 30 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the count clock (f_{MCK}).

(3) Operation in capture mode (input pulse interval measurement)

- <1> Operation is enabled ($TE_{mn} = 1$) by writing 1 to the TS_{mn} bit.
- <2> Timer counter register mn (TCR_{mn}) holds the initial value until count clock generation.
- <3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR_{mn} register and counting starts in the capture mode. (When the MD_{mn0} bit is set to 1, $INTTM_{mn}$ is generated by the start trigger.)
- <4> On detection of the valid edge of the TI_{mn} input, the value of the TCR_{mn} register is captured to timer data register mn (TDR_{mn}) and $INTTM_{mn}$ is generated. However, this captured value is meaningless. The TCR_{mn} register keeps on counting from 0000H.
- <5> On next detection of the valid edge of the TI_{mn} input, the value of the TCR_{mn} register is captured to timer data register mn (TDR_{mn}) and $INTTM_{mn}$ is generated.

Figure 7 - 31 Timing during Operation in Capture Mode (Input Pulse Interval Measurement)



Note If a clock has been input to TI_{mn} (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

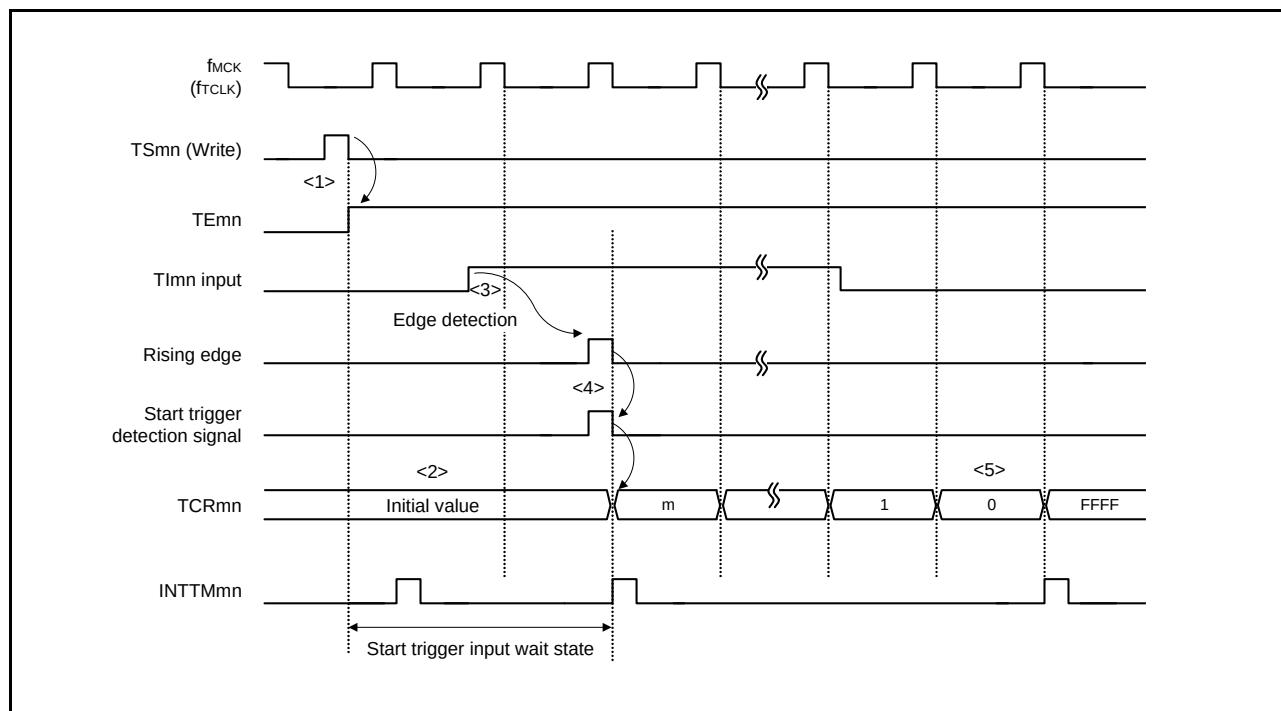
Caution In the first cycle operation of count clock after writing the TS_{mn} bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting $MD_{mn0} = 1$.

Remark Figure 7 - 31 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the count clock (f_{MCK}).

(4) Operation in one-count mode

- <1> Operation is enabled ($TE_{mn} = 1$) by writing 1 to the TS_{mn} bit.
- <2> Timer counter register mn (TCR_{mn}) holds the initial value until start trigger generation.
- <3> Rising edge of the TI_{mn} input is detected.
- <4> On start trigger detection, the value of timer data register mn (TDR_{mn}) is loaded to the TCR_{mn} register and count starts.
- <5> When the TCR_{mn} register counts down and its count value is 0000H, $INTT_{mn}$ is generated and the value of the TCR_{mn} register becomes FFFFH and counting stops.

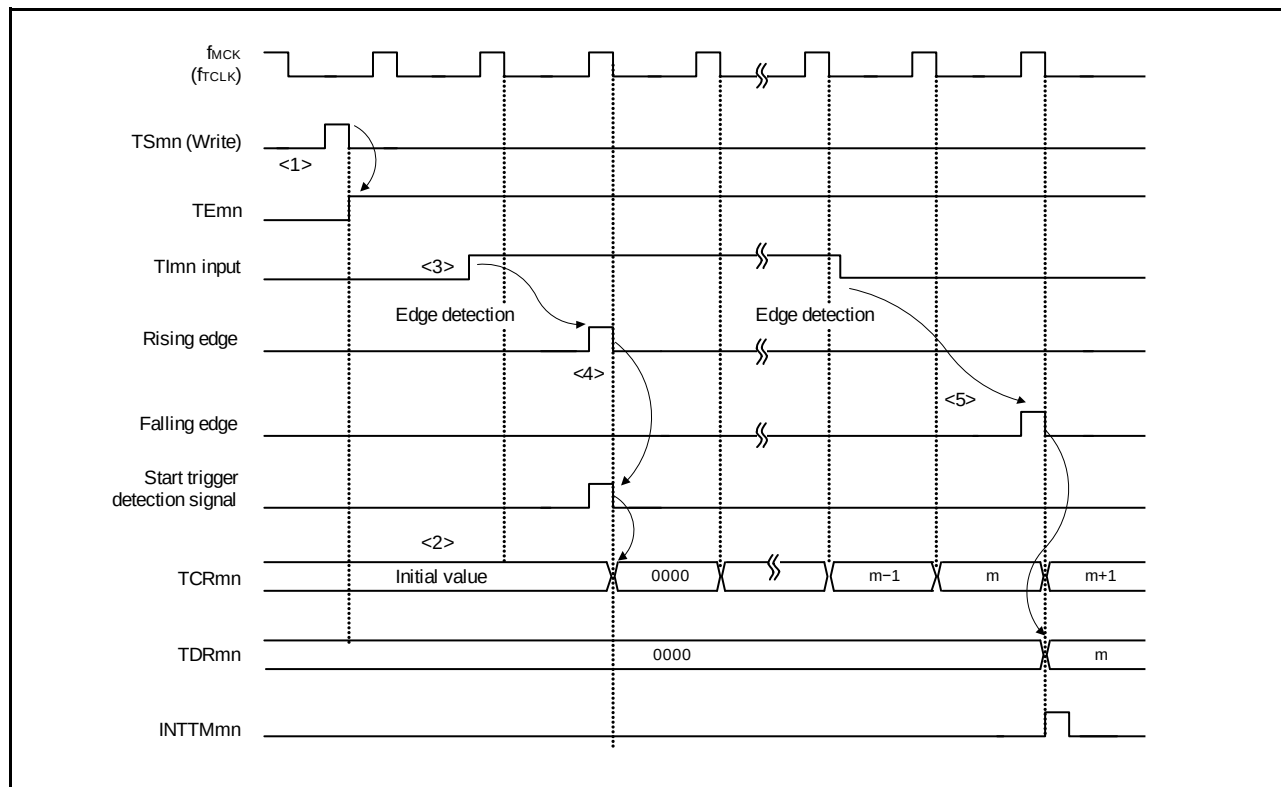
Figure 7 - 32 Timing during Operation in One-Count Mode



Remark Figure 7 - 32 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the count clock (f_{MCK}).

- (5) Operation in capture & one-count mode (high-level width measurement)
- <1> Operation is enabled ($TE_{mn} = 1$) by writing 1 to the TS_{mn} bit of timer channel start register m (TS_m).
 - <2> Timer counter register mn (TCR_{mn}) holds the initial value until start trigger generation.
 - <3> Rising edge of the TI_{mn} input is detected.
 - <4> On start trigger detection, the value of 0000H is loaded to the TCR_{mn} register and count starts.
 - <5> On detection of the falling edge of the TI_{mn} input, the value of the TCR_{mn} register is captured to timer data register mn (TDR_{mn}) and $INTT_{mn}$ is generated.

Figure 7 - 33 Timing during Operation in Capture & One-Count Mode (High-Level Width Measurement)

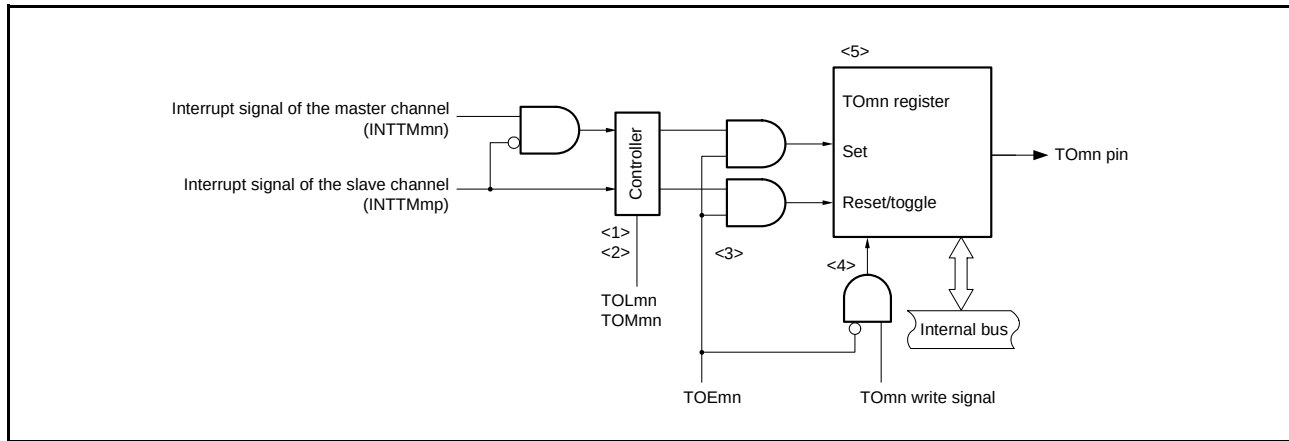


Remark Figure 7 - 33 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the count clock (f_{MCK}).

7.6 Channel Output (TOMn Pin) Control

7.6.1 TOMn pin output circuit configuration

Figure 7 - 34 Output Circuit Configuration



The following describes the TOMn pin output circuit.

- <1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOM).
- <2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOM register.

At this time, the TOLm register becomes valid and the signals are controlled as follows:

- When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
- When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

- <3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOM register. Writing to the TOM register (TOMn write signal) becomes invalid.

When TOEmn = 1, the TOMn pin output never changes with signals other than interrupt signals.

To initialize the TOMn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOM register.

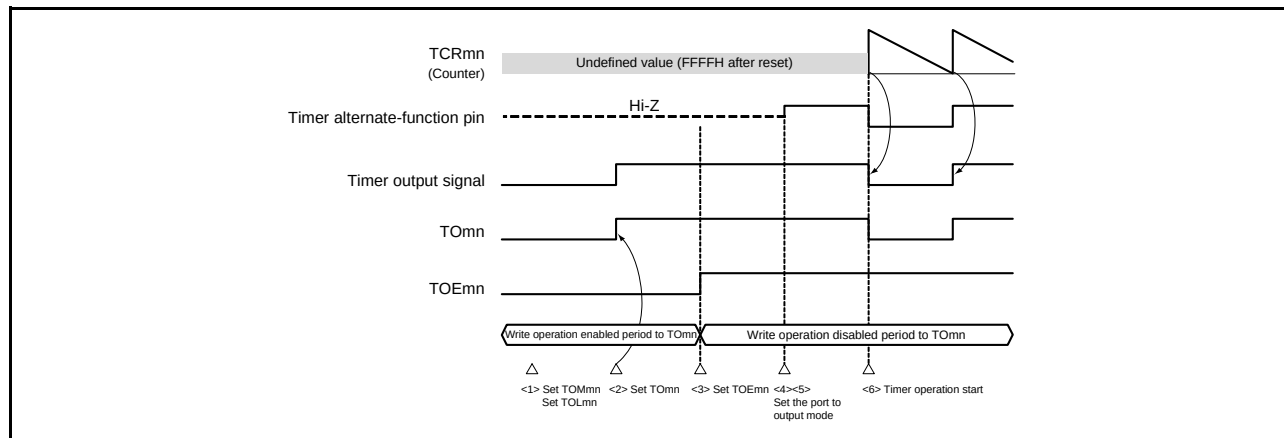
- <4> While timer output is disabled (TOEmn = 0), writing to the TOMn bit to the target channel (TOMn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOM register.
- <5> The TOM register can always be read, and the TOMn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

7.6.2 TOMn pin output setting

The following figure shows the procedure and state transitions from the initial settings of a TOMn output pin to the start of timer operation.

Figure 7 - 35 State Transitions from the settings for timer output to the start of timer operation.



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial state by setting timer output register m (TOM).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOM register is disabled).

<4> The port is set to digital I/O by the port mode control A register (PMCAxx) and port mode control T register (PMCTxx) (see **7.3.17 Registers controlling port functions of pins to be used for timer I/O**).

<5> The port I/O setting is set to output (see **7.3.17 Registers controlling port functions of pins to be used for timer I/O**).

<6> The timer operation is enabled (TSMn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.6.3 Cautions on channel output operation

(1) Changing values set in the registers TOM, TOEm, and TOLm during timer operation

Since the timer operations (operations of timer counter register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOMn output circuit and changing the values set in timer output register m (TOM), timer output enable register m (TOEm), and timer output level register m (TOLm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOMn pin by timer operation, however, set the TOM, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown in 7.7 and 7.8.

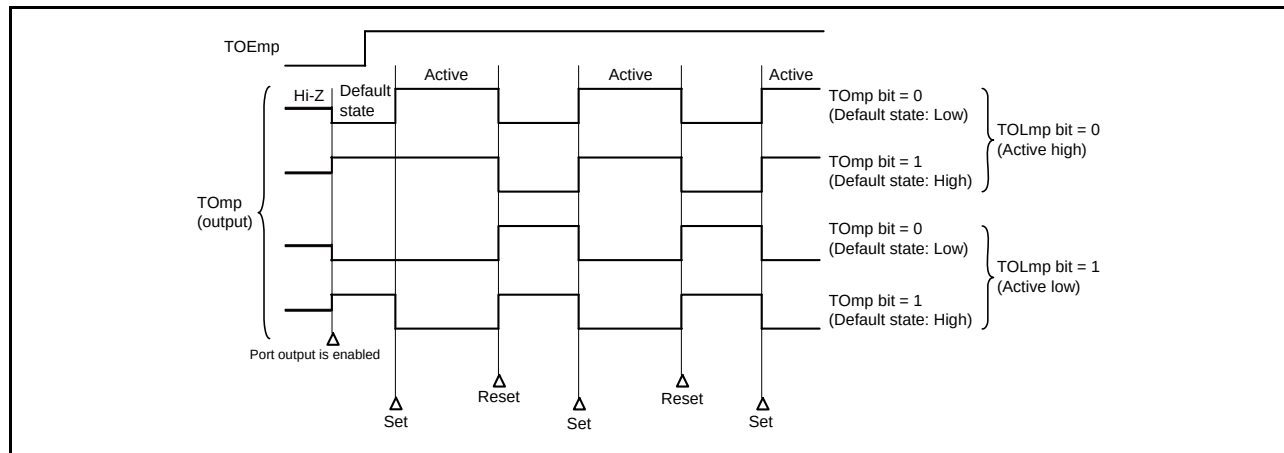
When the values set to the TOEm, and TOMm registers (but not the TOM register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOMn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

- (b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output)

When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 7 - 37 TOmp Pin Output States with PWM Output (TOMmp = 1)



Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.

Reset: The output signal of the TOmp pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

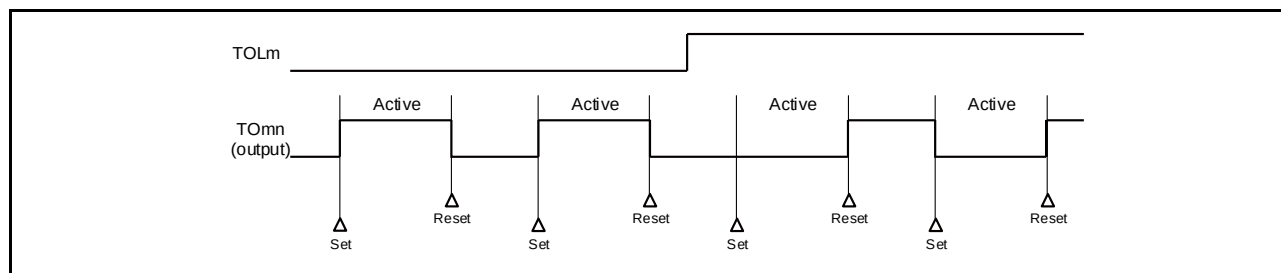
(3) Operation of TOMn pin in slave channel output mode (TOMmn = 1)

(a) When the relevant bit of timer output level register m (TOLm) is changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOMn pin change condition. Rewriting the TOLm register does not change the output level of the TOMn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 7 - 38 Operation When the Relevant Bit of the TOLm Register is Changed during Timer Operation



Remark 1. Set: The output signal of the TOMn pin changes from inactive level to active level.

Reset: The output signal of the TOMn pin changes from active level to inactive level.

Remark 2. m: Unit number ($m = 0$), **n:** Channel number ($n = 0$ to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOMn pin/TOMn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

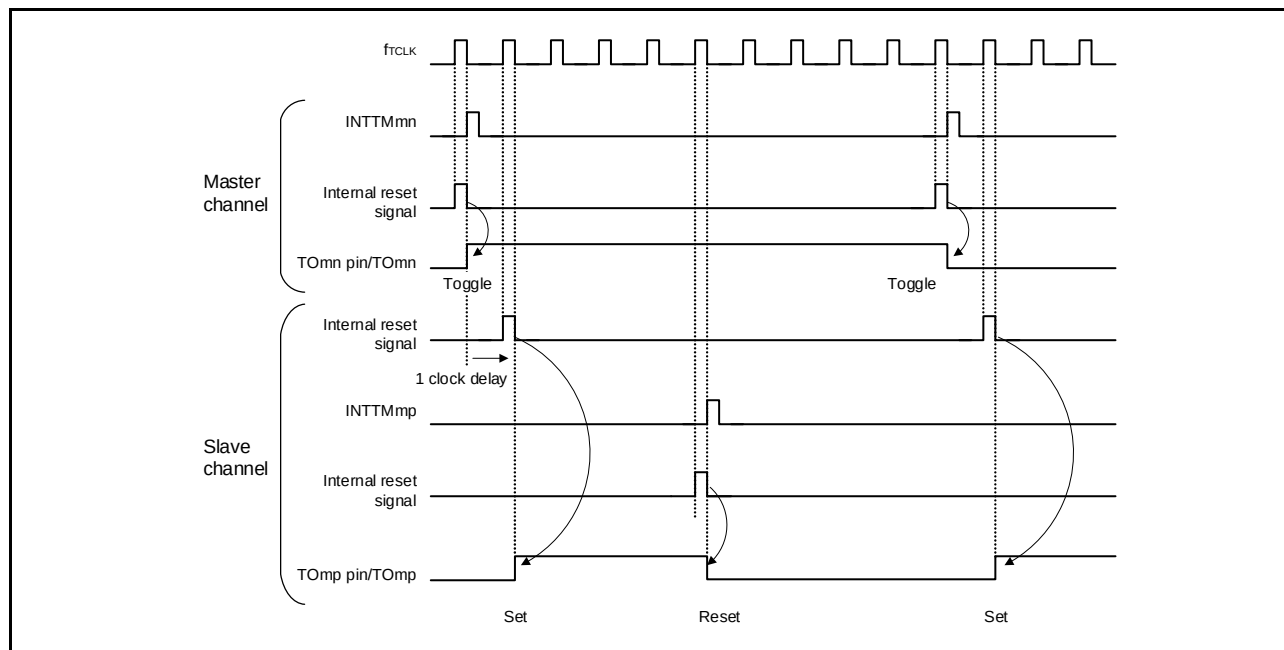
Figure 7 - 39 shows the states of operation following set and reset signals when the master and slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0

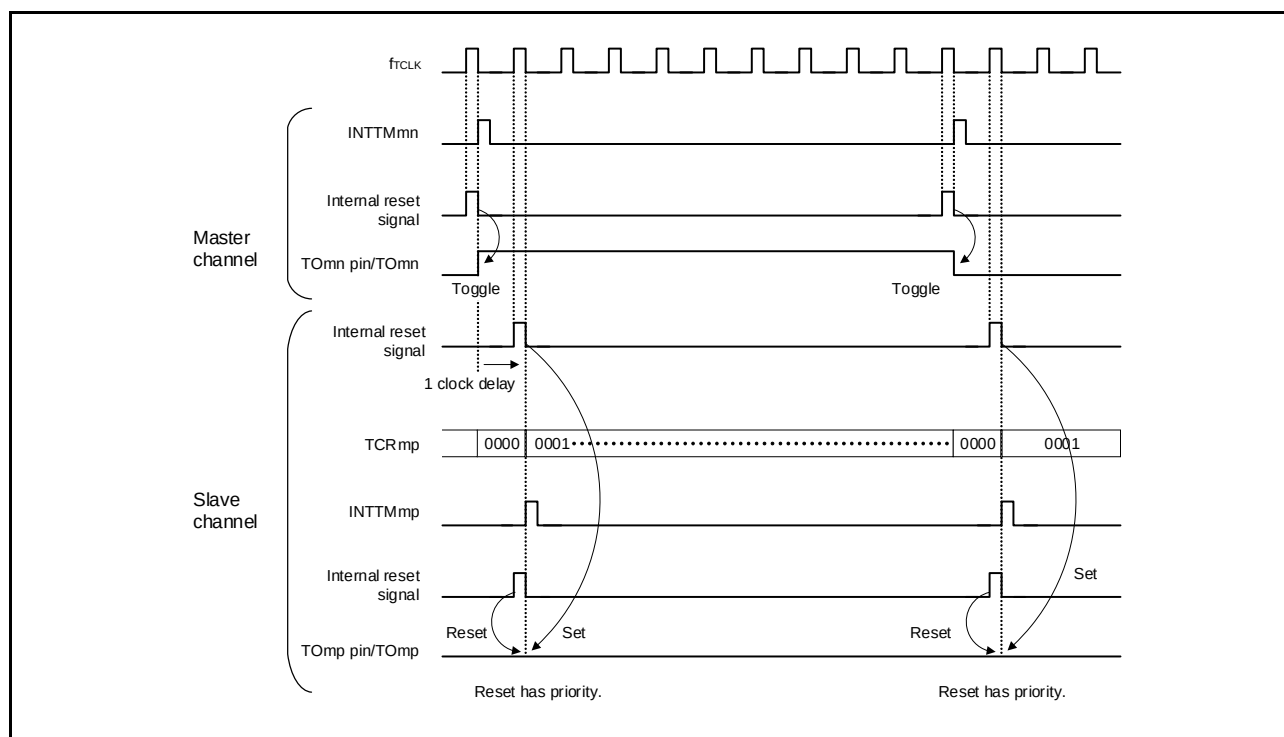
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 7 - 39 States of Operation following Set and Reset Signals

(1) Basic timing during operation



(2) Timing during operation with the duty cycle set to 0%



Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

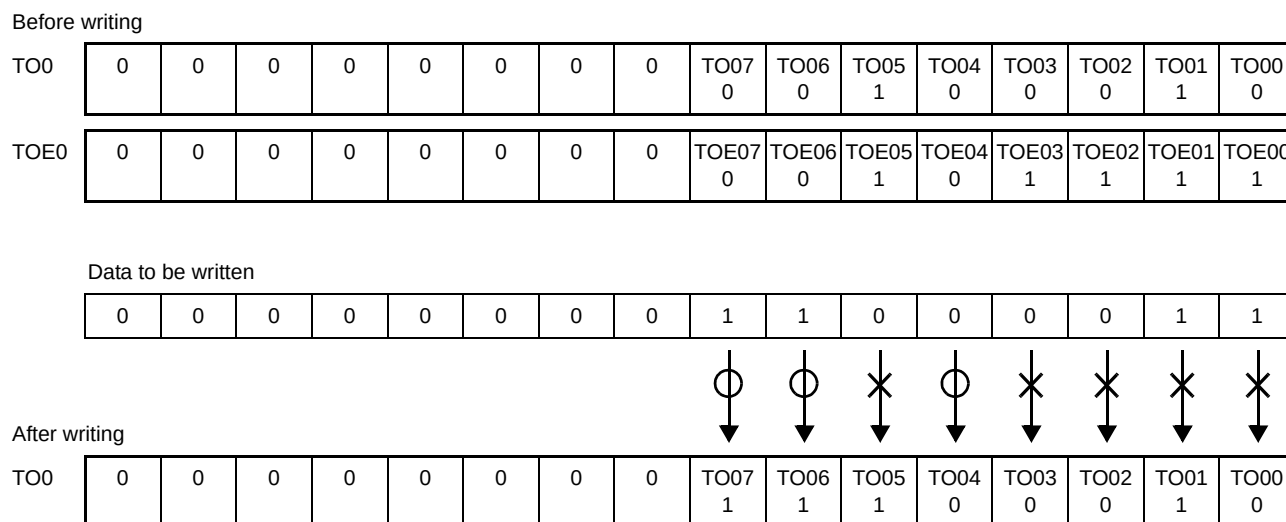
Remark 2. m: Unit number ($m = 0$)
n: Channel number
 $n = 0$ to 7 ($n = 0, 2, 4, 6$ for master channel)
p: Slave channel number
 $n < p \leq 7$

7.6.4 Collective manipulation of TOMn bit

In timer output register m (TOM), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSM). Therefore, the TOMn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOMn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOMn).

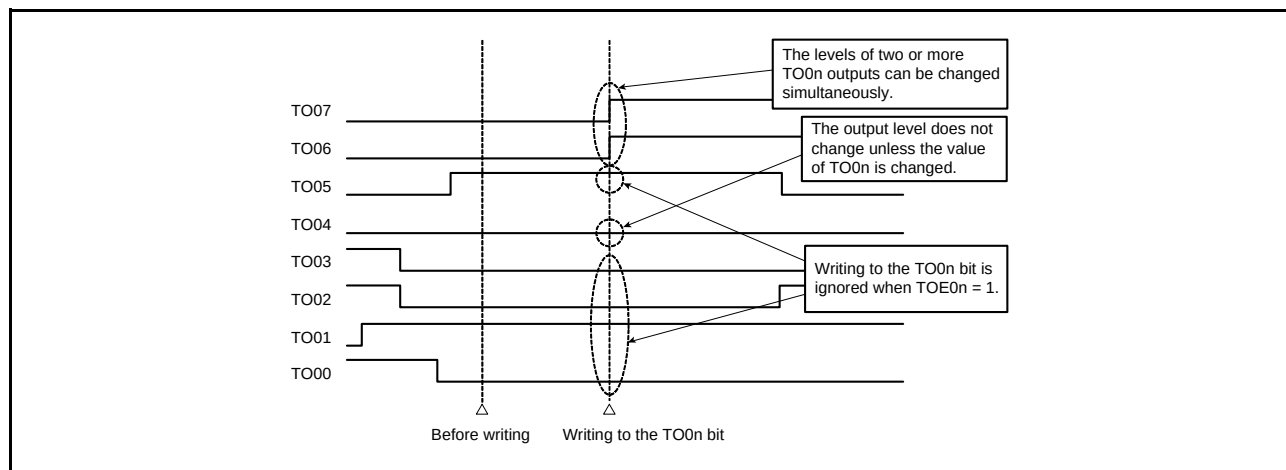
Figure 7 - 40 Example of TO0n Bit Collective Manipulation



Writing is done only to the TOMn bit with TOEmn = 0, and writing to the TOMn bit with TOEmn = 1 is ignored.

TOMn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOMn bit, it is ignored and the output change by timer operation is normally done.

Figure 7 - 41 TO0n Pin States by Collective Manipulation of TO0n Bits



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.6.5 Timer interrupts and TOMn outputs when counting is started

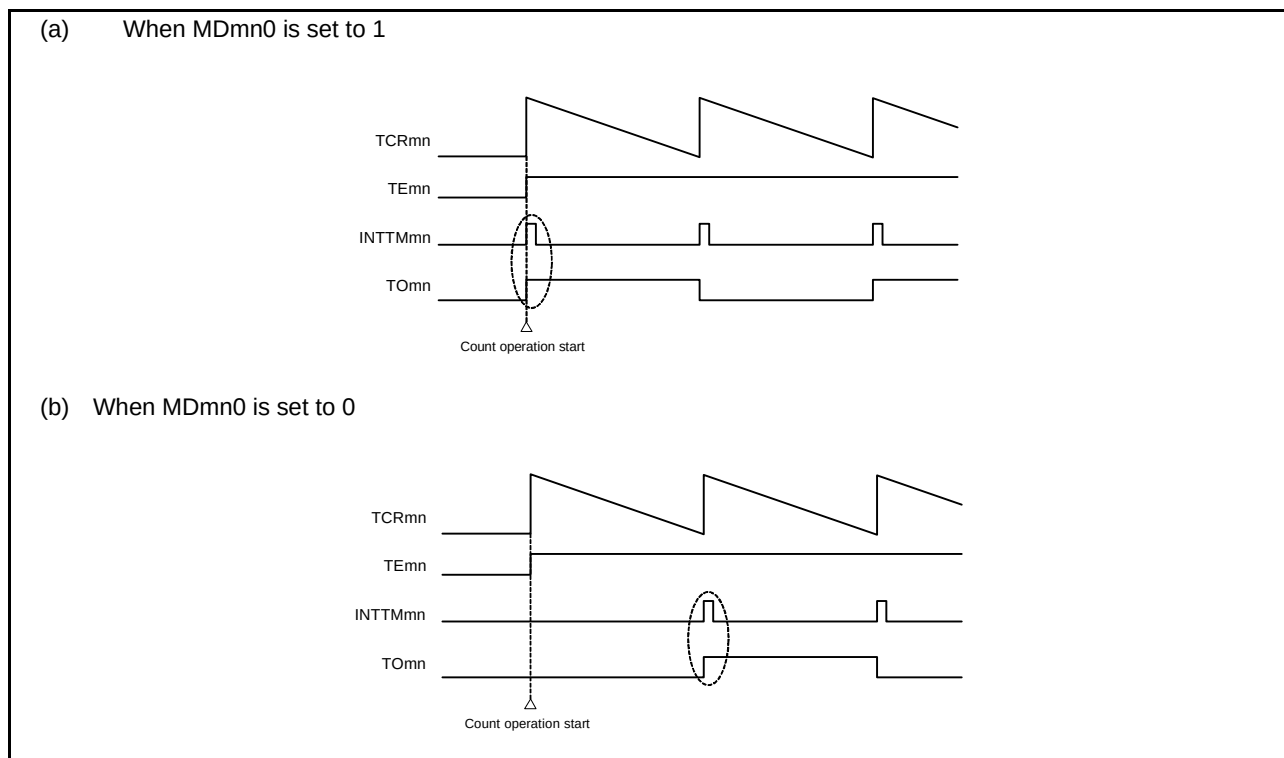
In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOMn output is controlled.

Figure 7 - 42 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 7 - 42 Examples of the Operation of Timer Interrupts and TOMn Outputs When Counting is Started



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOMn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOMn does not change either. After counting one cycle, INTTMmn is output and TOMn performs a toggle operation.

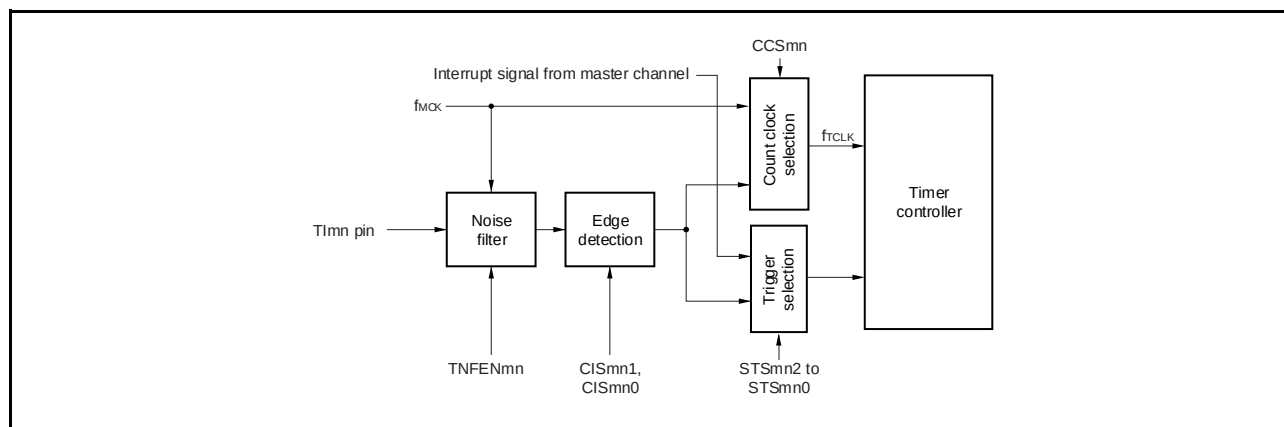
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.7 Timer Input (TImn) Control

7.7.1 TImn input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

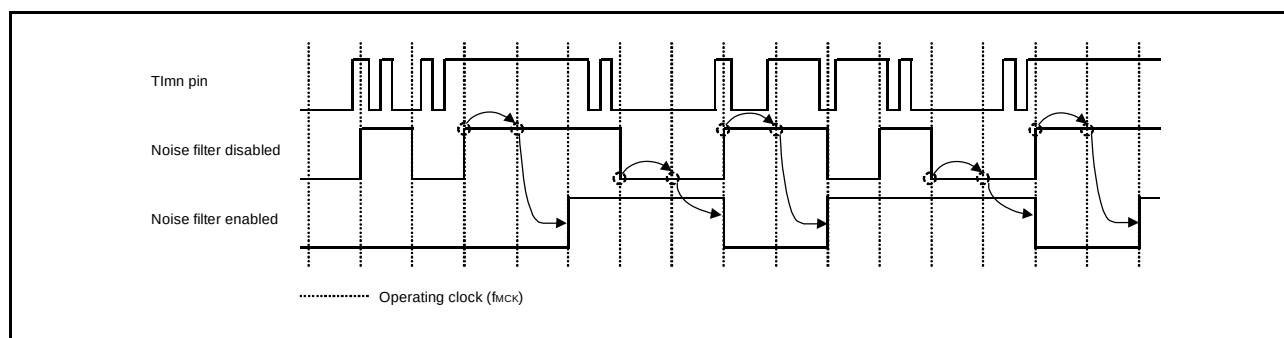
Figure 7 - 43 Input Circuit Configuration



7.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fmck) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fmck) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 7 - 44 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



7.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (fMCK), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (fMCK), and then set the operation enable trigger bit in the timer channel start register (TSm).

7.8 Independent Channel Operation Function of Timer Array Unit

7.8.1 Operation as an interval timer or for square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$

(2) Operation for square wave output

TOMn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOMn can be calculated by the following expressions.

• Period of square wave output from TOMn = Period of count clock $\times$ (Set value of TDRmn + 1) $\times$ 2

• Frequency of square wave output from TOMn = Frequency of count clock / {(Set value of TDRmn + 1) $\times$ 2}

Timer counter register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSMn, TSHm1, TSHm3) of timer channel start register m (TSM) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and the output on TOMn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and the output on TOMn is toggled.

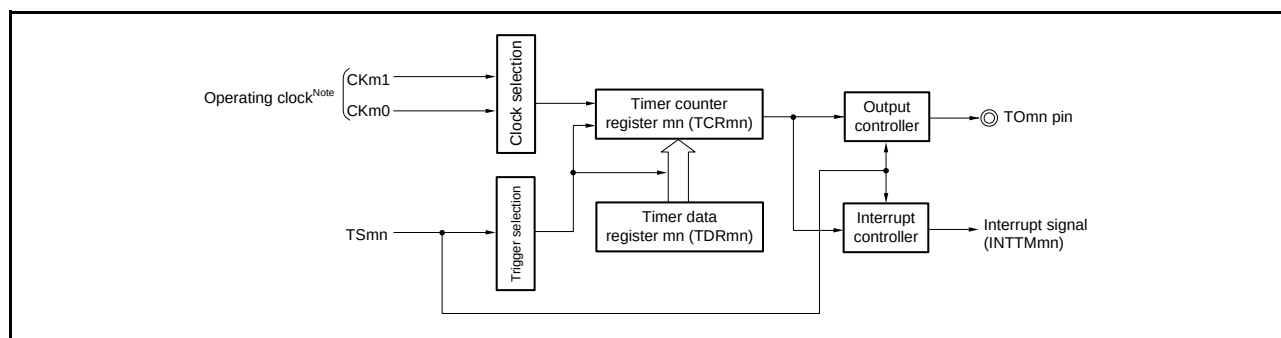
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and the output on TOMn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

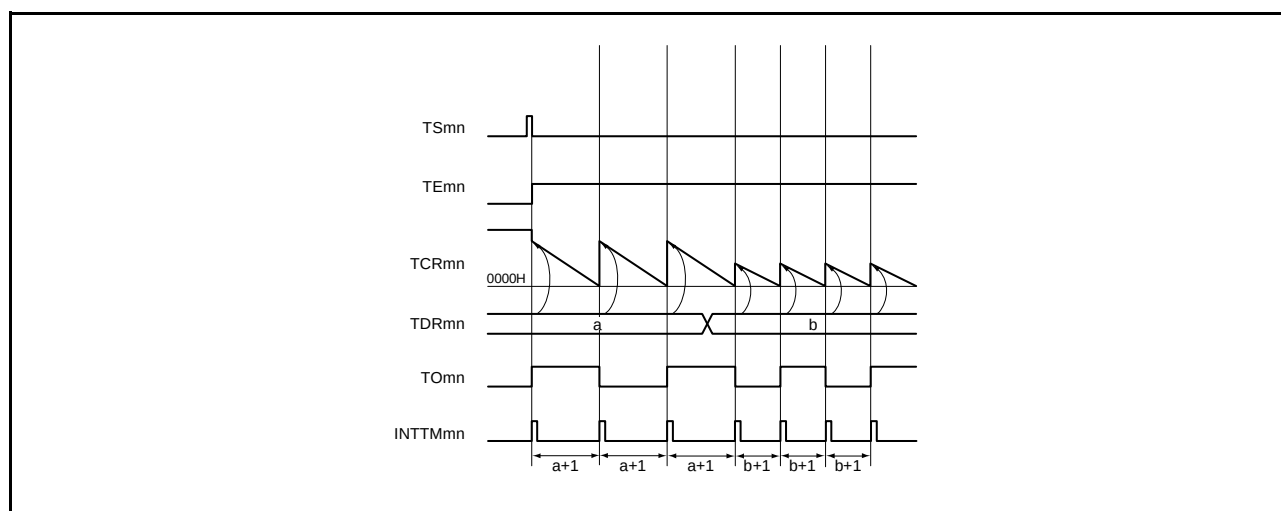
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 45 Block Diagram for Operation as an Interval Timer or for Square Wave Output



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 7 - 46 Example of Basic Timing during Operation as an Interval Timer or for Square Wave Output (MDmn0 = 1)



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)

TE mn: Bit n of timer channel enable status register m (TE m)

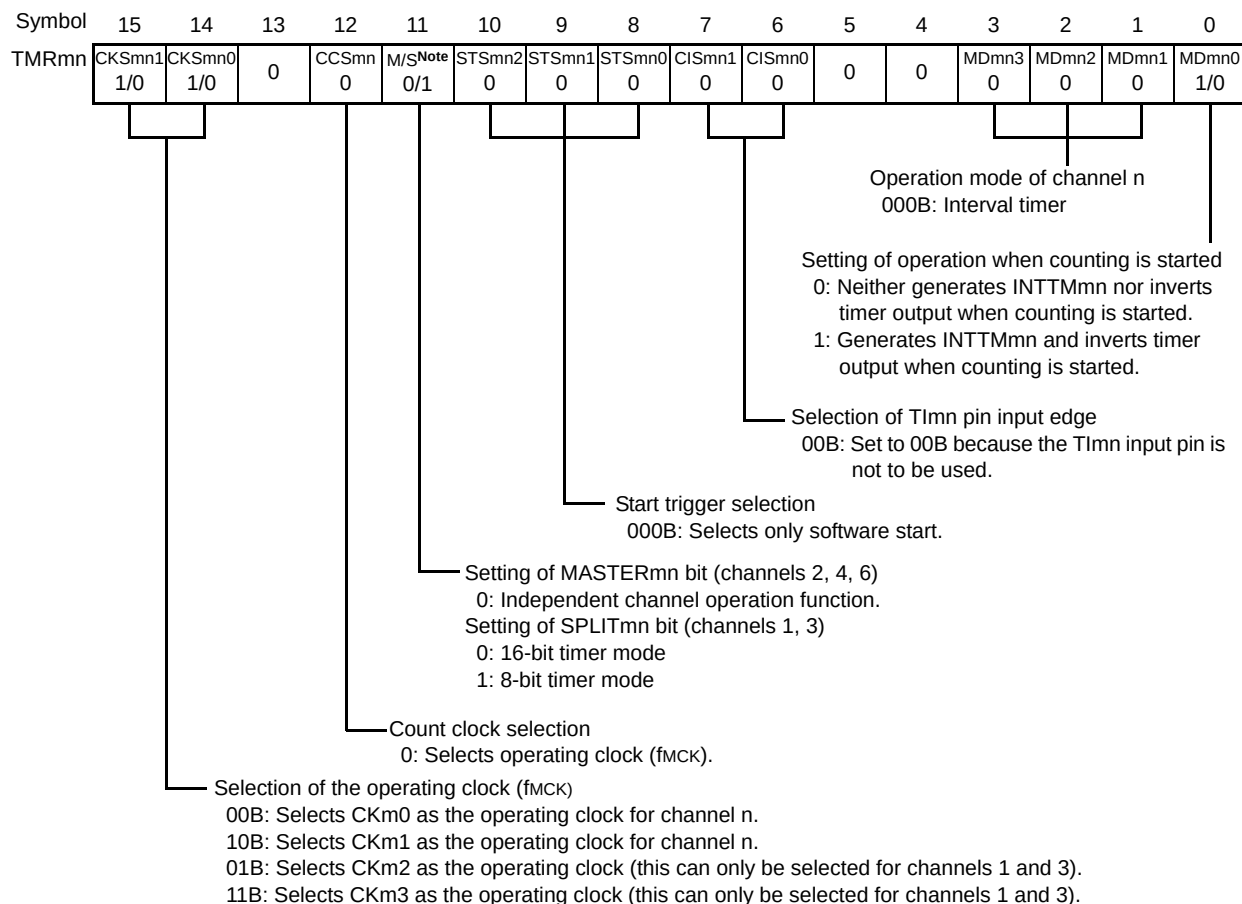
TCRmn: Timer counter register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

TOMn: TOMn pin output signal

Figure 7 - 47 Example of Register Settings for Operation as an Interval Timer or for Square Wave Output

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOM)

Bit n	
TOM	TOMn 1/0
	0: Outputs 0 from TOMn. 1: Outputs 1 from TOMn.

(c) Timer output enable register m (TOEm)

Bit n	
TOEm	TOEmn 1/0
	0: Stops the TOMn output operation by counting operation. 1: Enables the TOMn output operation by counting operation.

(d) Timer output level register m (TOLm)

Bit n	
TOLm	TOLmn 0
	0: Set this bit to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

Bit n	
TOMm	TOMmn 0
	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 48 Procedure for Operations When the Interval Timer or Square Wave Output Function is to be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOMn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOMn bit and determines default level of the TOMn output. →	The TOMn pin goes into Hi-Z output state. The TOMn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOMn. →	TOMn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOMn pin outputs the TOMn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOMn output and resuming operation.) Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer counter register mn (TCRmn). INTTMmn is generated and TOMn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOM and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOMn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 0, and count operation stops. The TCRmn register holds count value and stops. The TOMn output is not initialized and retains its current state.
	The TOEmn bit is cleared to 0 and value is set to the TOMn bit. →	The TOMn pin outputs the TOMn bit set level.
TAU stop	To hold the TOMn pin output level Clears the TOMn bit to 0 after the value to be held is set to the port register. → When holding the TOMn pin output level is not necessary Setting not required.	The TOMn pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOMn bit is cleared to 0 and the TOMn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.8.2 Operation as an external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

$$\text{Specified number of counts} = \text{Set value of TDRmn} + 1$$

Timer counter register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn) of timer channel start register m (TSM) to 1.

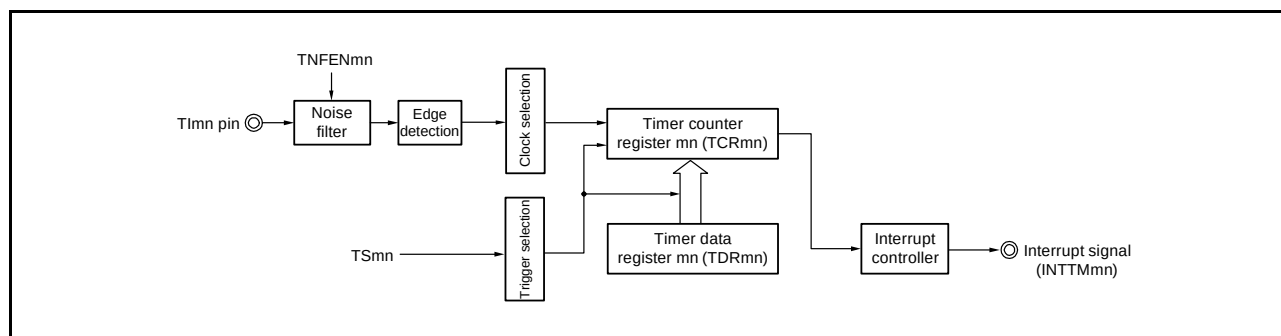
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOMn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

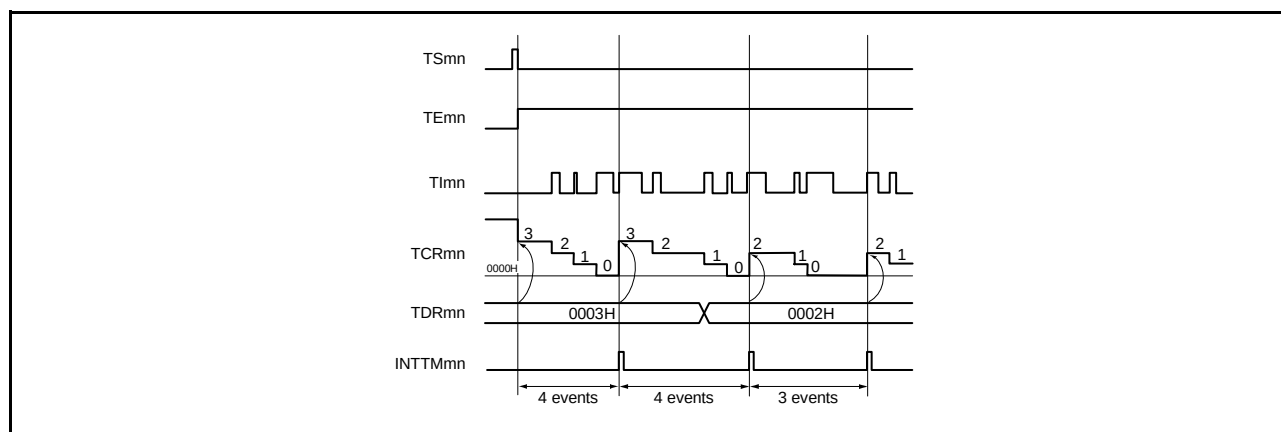
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 7 - 49 Block Diagram for Operation as an External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 50 Example of Basic Timing during Operation as an External Event Counter



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)

TE mn: Bit n of timer channel enable status register m (TE m)

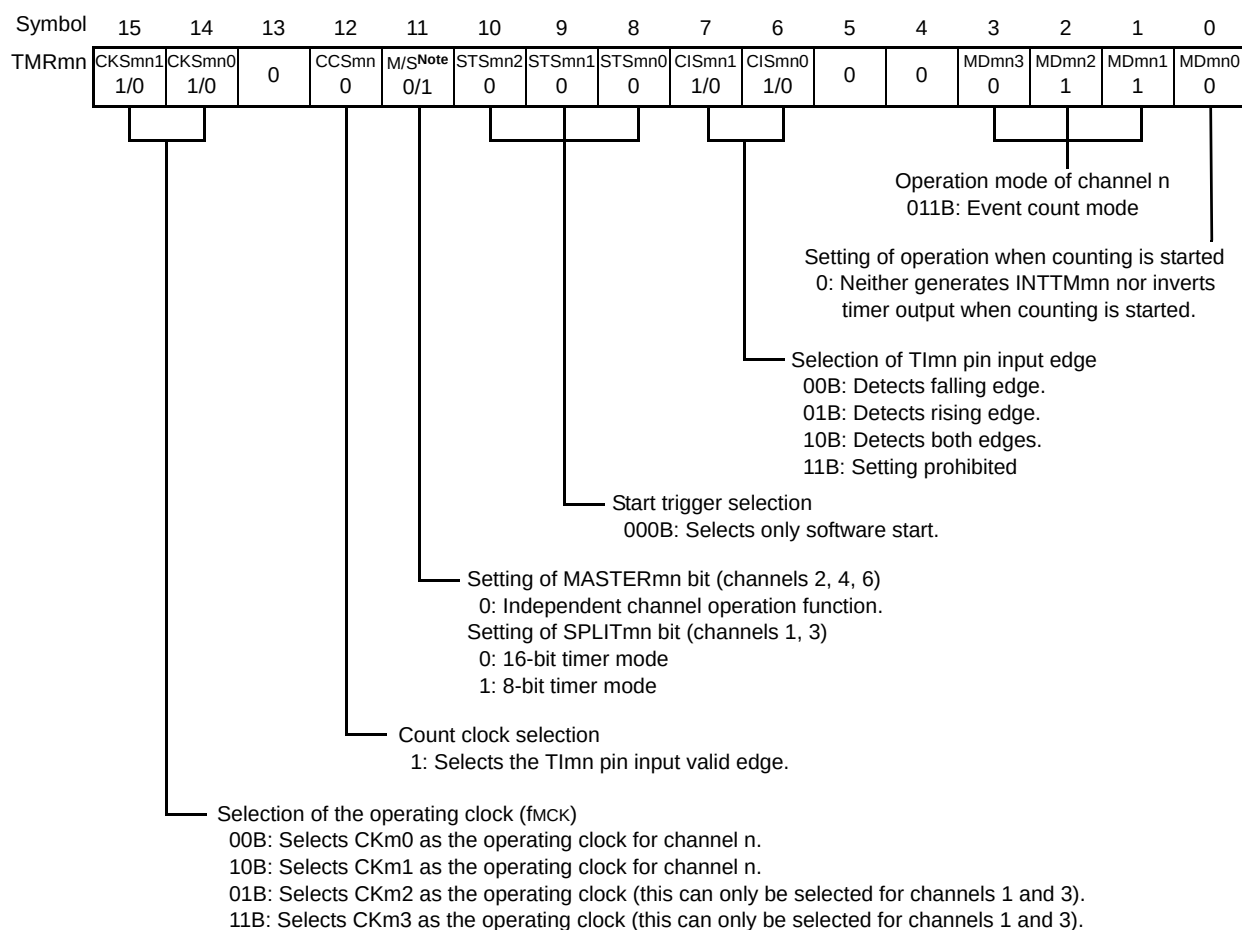
TImn: TImn pin input signal

TCRmn: Timer counter register mn (TCRmn)

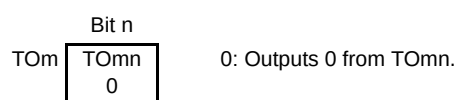
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 51 Example of Register Settings in External Event Counter Mode

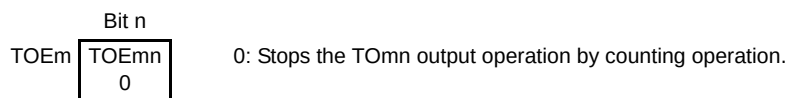
(a) Timer mode register mn (TMRmn)



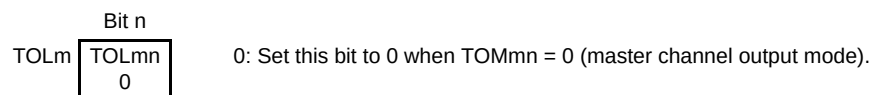
(b) Timer output register m (TOM)



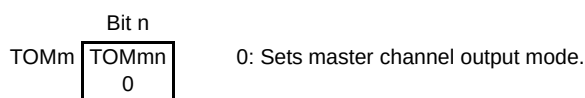
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 52 Procedure for Operations When the External Event Counter Function is to be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation is resumed	Operation start Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer counter register mn (TCRmn) and detection of the Tlmn pin input edge is awaited.
	During operation Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOMn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the Tlmn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
	Operation stop The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
	TAU stop The TAUmEN bit of the PER0 register is cleared to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.8.3 Operation as a frequency divider (channel 0 of unit 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
Divided clock frequency = Input clock frequency / {(Set value of TDR00 + 1) × 2}
- When both edges are selected:
Divided clock frequency ≈ Input clock frequency / (Set value of TDR00 + 1)

Timer counter register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and the output on TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and the output on TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles the output on TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting. If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operating clock.

$$\text{Clock period of TO00 output} = \text{Ideal TO00 output clock period} \pm \text{Operating clock period (error)}$$

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 7 - 53 Block Diagram for Operation as a Frequency Divider

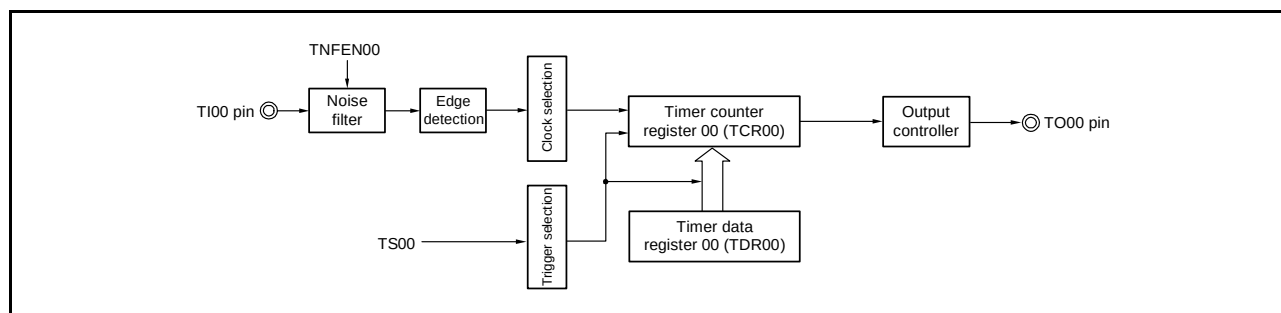
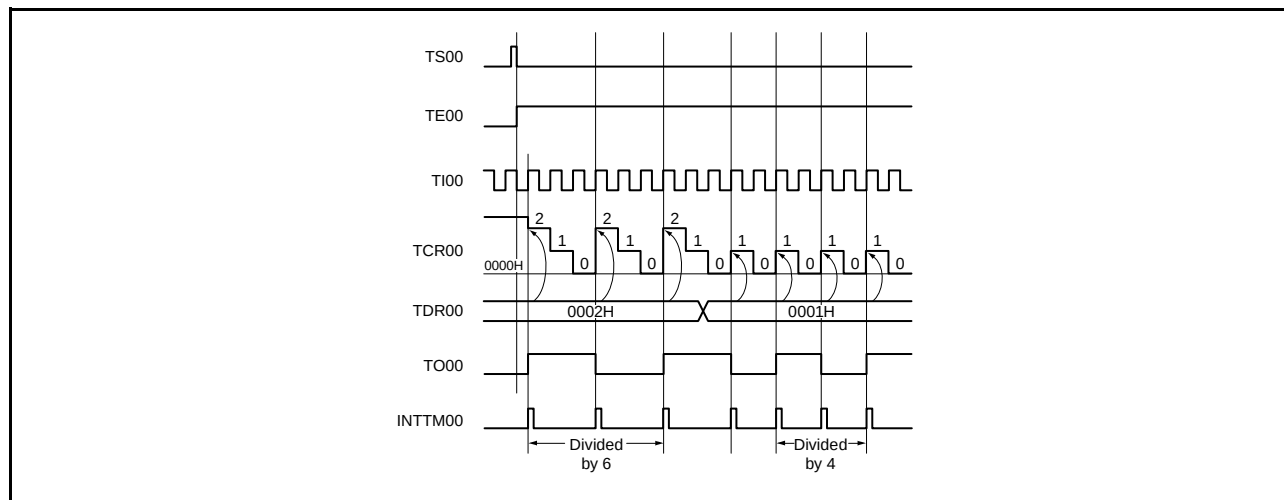


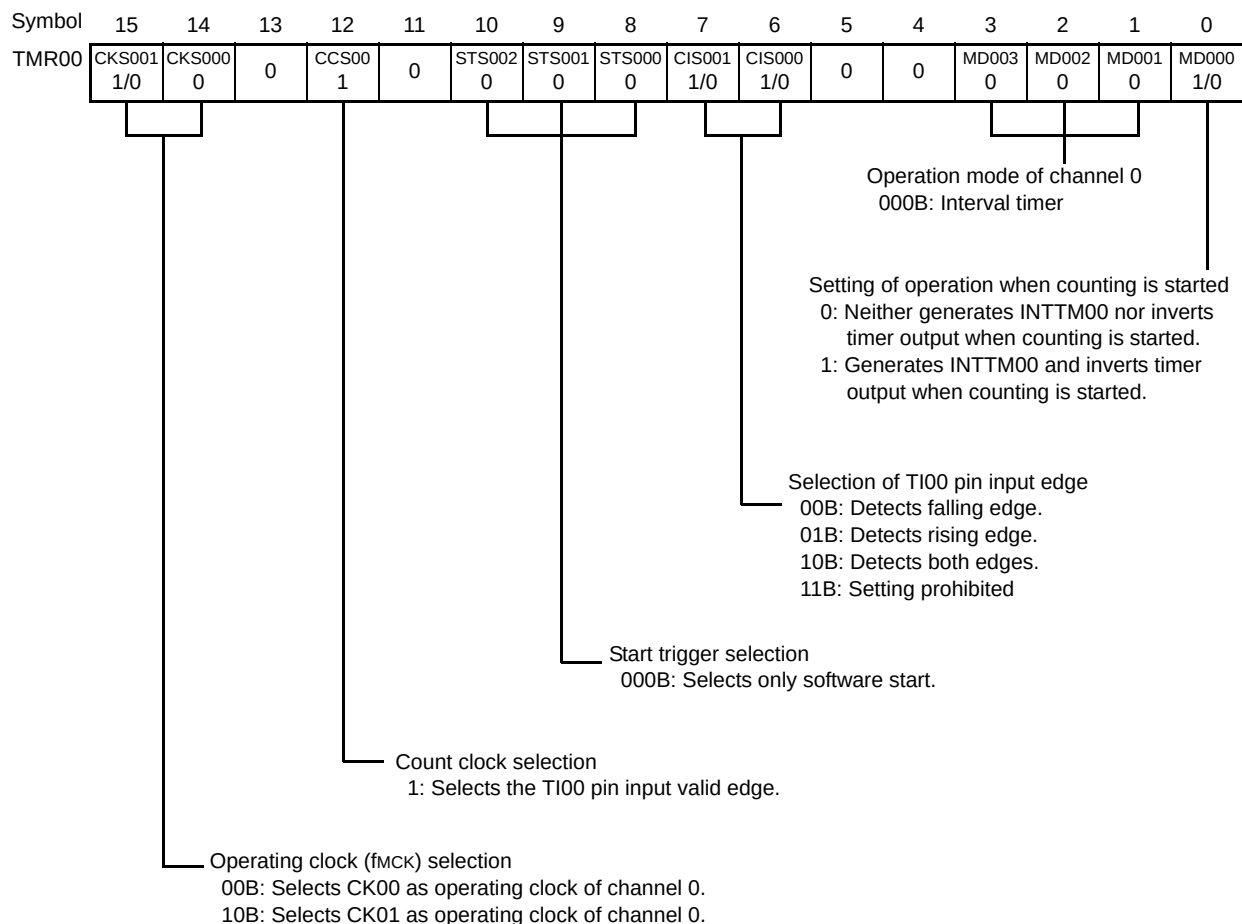
Figure 7 - 54 Example of Basic Timing during Operation as a Frequency Divider (MD000 = 1)



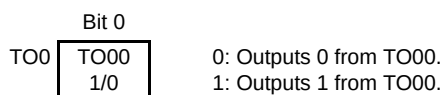
- Remark**
- TS00: Bit 0 of timer channel start register 0 (TS0)
 - TE00: Bit 0 of timer channel enable status register 0 (TE0)
 - TI00: TI00 pin input signal
 - TCR00: Timer counter register 00 (TCR00)
 - TDR00: Timer data register 00 (TDR00)
 - TO00: TO00 pin output signal

Figure 7 - 55 Example of Register Settings for Operation as a Frequency Divider

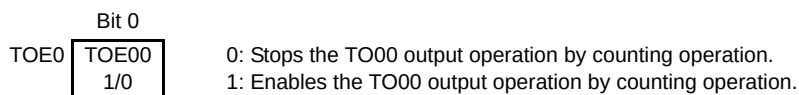
(a) Timer mode register 00 (TMR00)



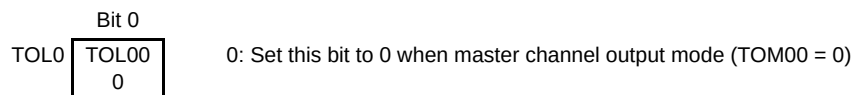
(b) Timer output register 0 (TO0)



(c) Timer output enable register 0 (TOE0)



(d) Timer output level register 0 (TOL0)



(e) Timer output mode register 0 (TOM0)

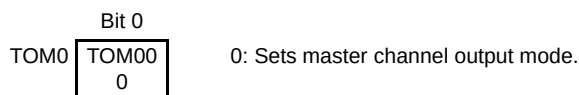


Figure 7 - 56 Procedure for Operations When the Frequency Divider Function is to be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state.
	Sets the TOE00 bit to 1 and enables operation of TO00. →	The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Clears the port register and port mode register to 0. →	TO00 does not change because channel stops operating. The TO00 pin outputs the TO00 set level.

Figure 7 - 56 Procedure for Operations When the Frequency Divider Function is to be Used (2/2)

	Software Operation	Hardware State
Operation is resumed	Operation start Sets the TOE00 bit to 1 (only when operation is resumed). Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer counter register 00 (TCR00). INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
	During operation Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
	Operation stop The TT00 bit is set to 1. The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized and retains its current state.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit.	The TO00 pin outputs the TO00 set level.
	TAU stop To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0. Set the TAUMRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit.	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

7.8.4 Operation for input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEMn bit is set to 1.

The pulse interval can be calculated by the following expression.

$$\text{TImn input pulse interval} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Captured value of TDRmn} + 1))$$

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSM) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

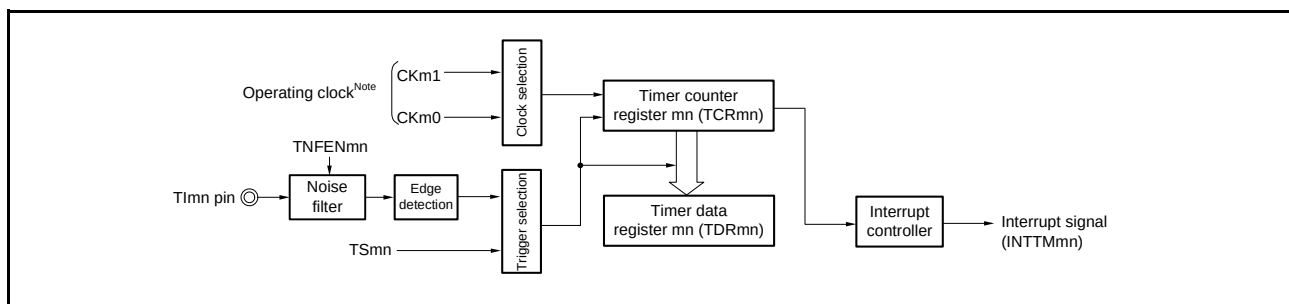
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow state of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

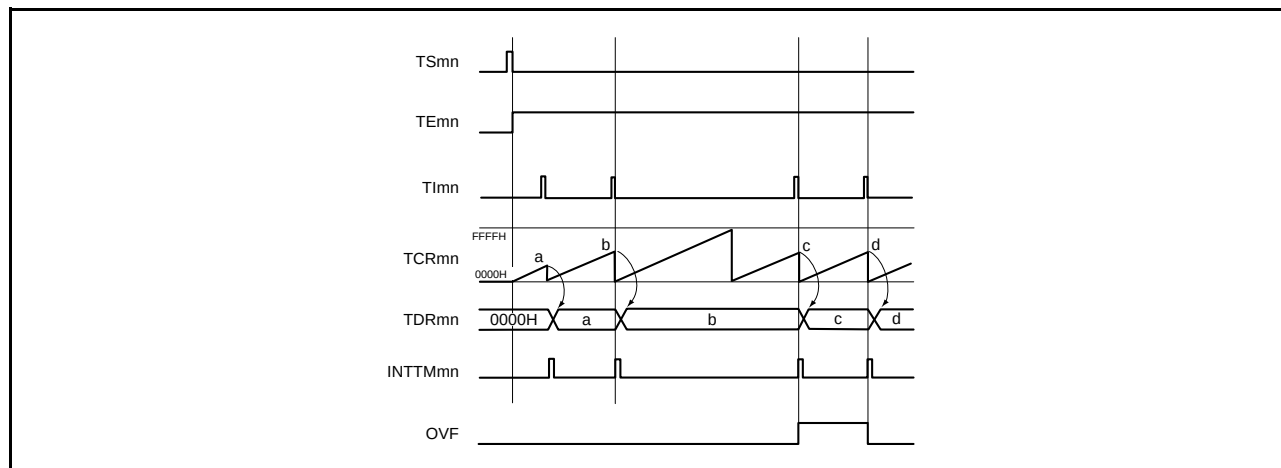
Figure 7 - 57 Block Diagram for Operation for Input Pulse Interval Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 58 Example of Basic Timing during Operation for Input Pulse Interval Measurement (MDmn0 = 0)

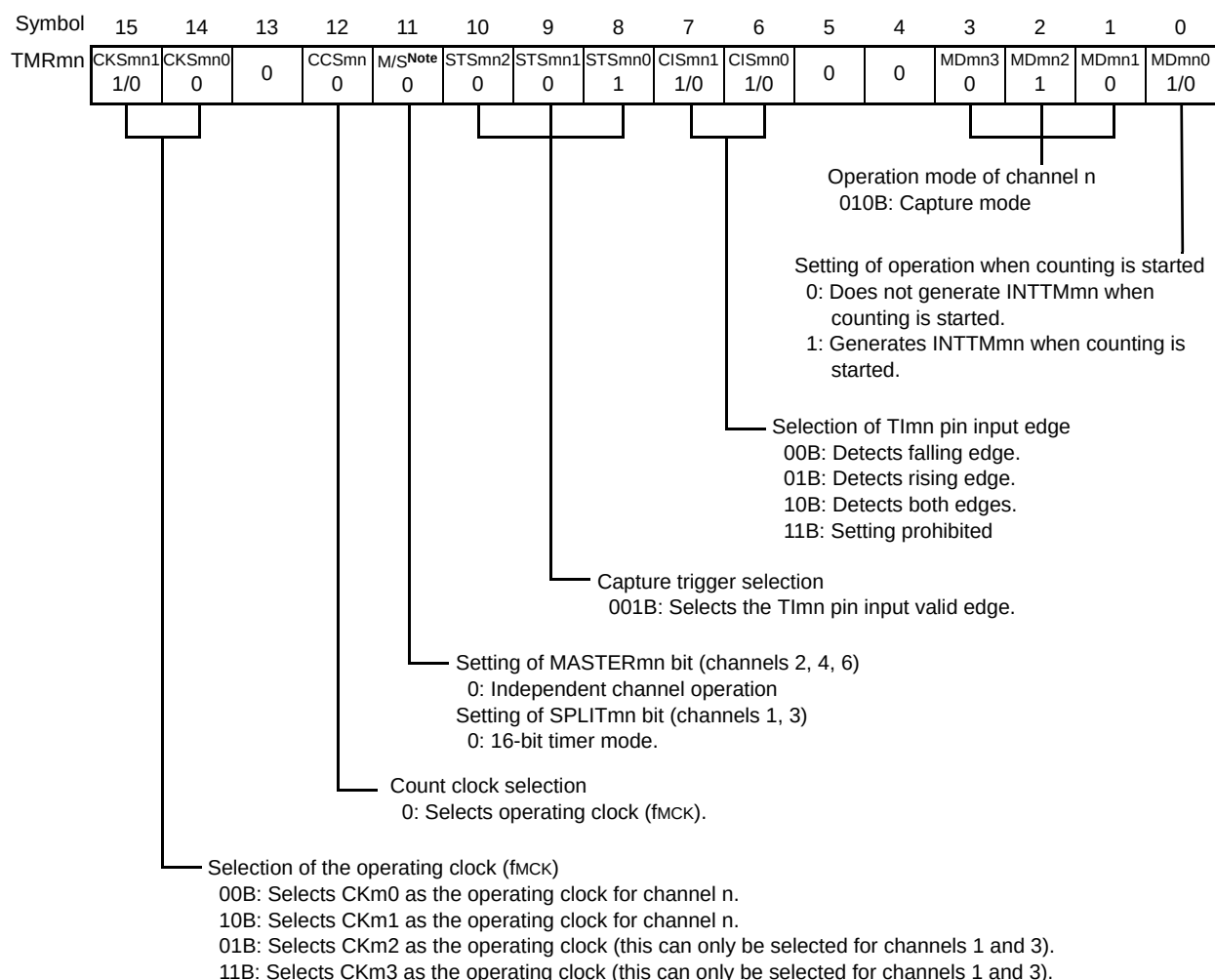


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

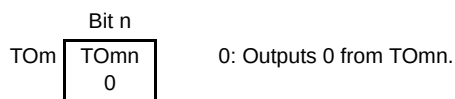
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TE mn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 59 Example of Register Settings for Operation for Input Pulse Interval Measurement

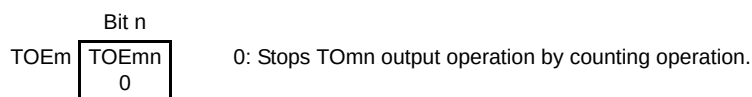
(a) Timer mode register mn (TMRmn)



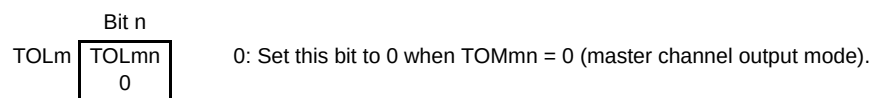
(b) Timer output register m (TOM)



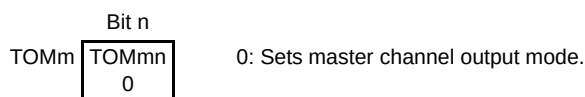
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



(Note and Remark are listed on the next page.)

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
 TMRm1, TMRm3: SPLITmn bit
 TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 60 Procedure for Operations When the Input Pulse Interval Measurement Function is to be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer counter register mn (TCRmn) is cleared to 0000H. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOMn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the valid edge of the TImn pin input is detected or the TSmn bit is set to 1, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	This stops supply of the input clock to timer array unit m.
	Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.8.5 Operation for input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD2.

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

$$\text{Signal width of TImn input} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Captured value of TDRmn} + 1))$$

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operating clock occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSM) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait state is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value "value transferred to the TDRmn register + 1", and the TImn pin start edge detection wait state is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow state of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

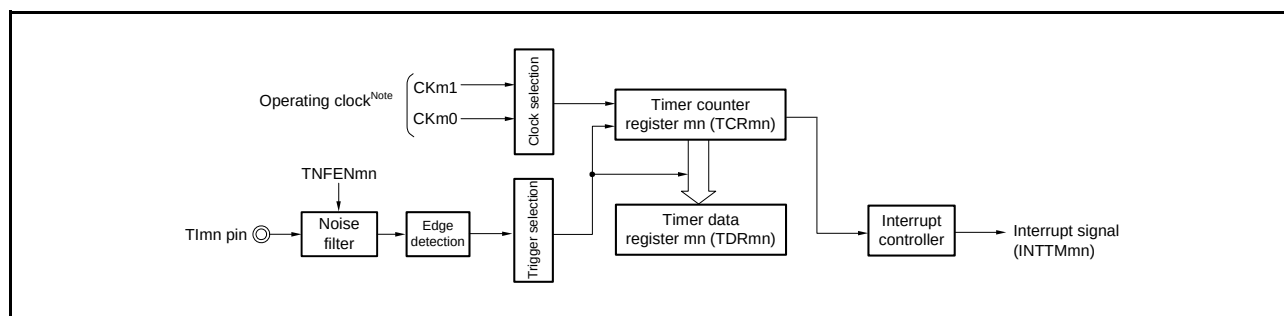
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.

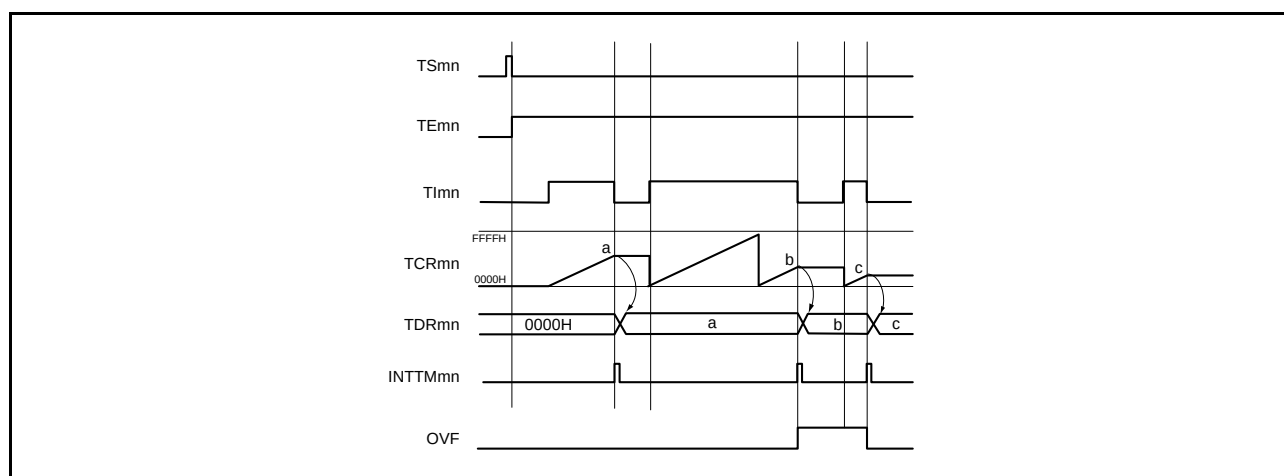
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 7 - 61 Block Diagram for Operation for Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 7 - 62 Example of Basic Timing during Operation for Input Signal High-/Low-Level Width Measurement



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

Timn: Timn pin input signal

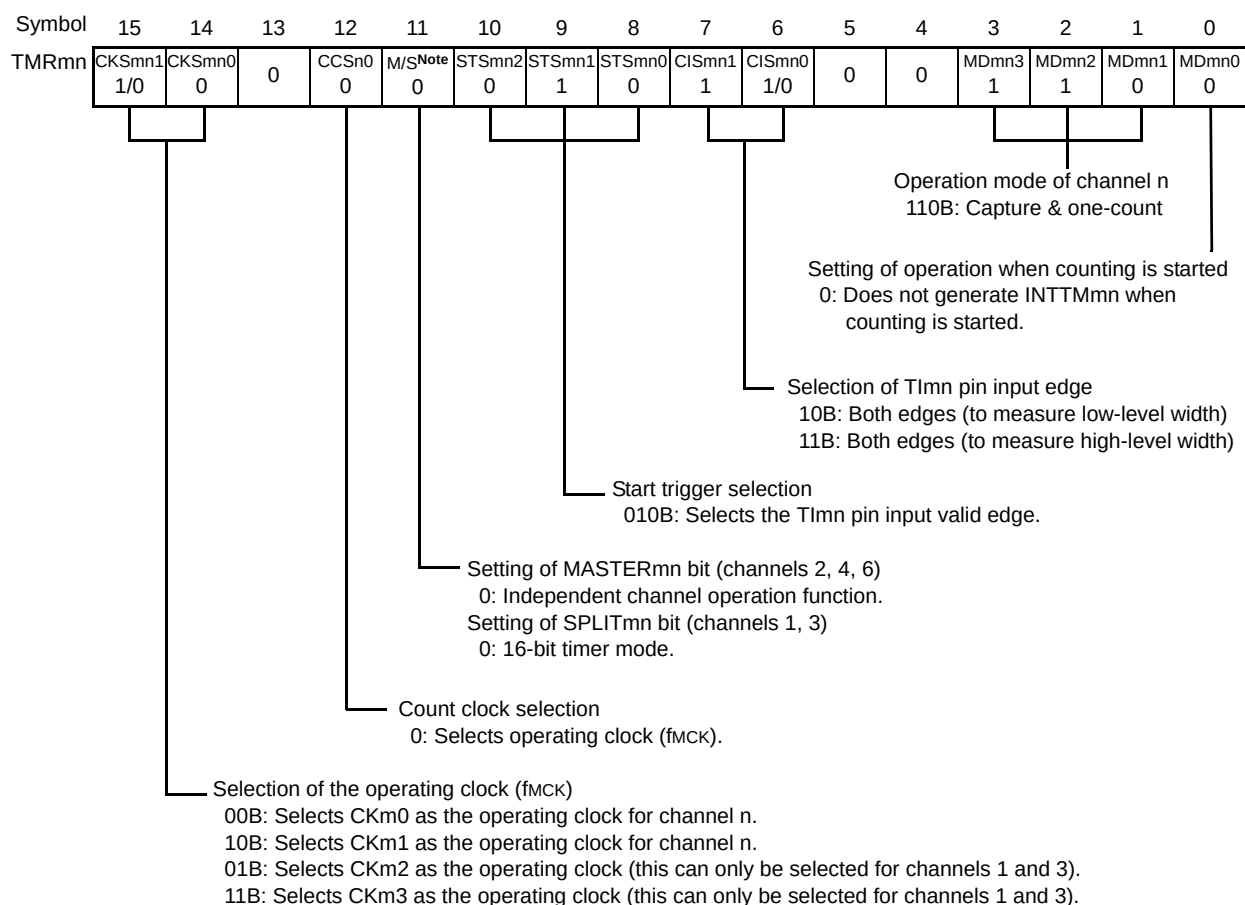
TCRmn: Timer counter register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

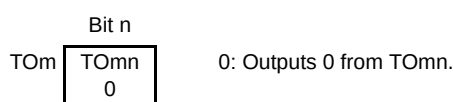
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 63 Example of Register Settings for Operation for Input Signal High-/Low-Level Width Measurement

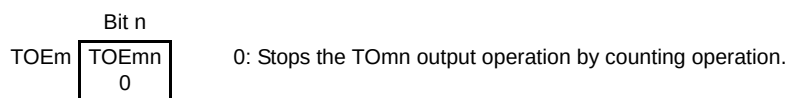
(a) Timer mode register mn (TMRmn)



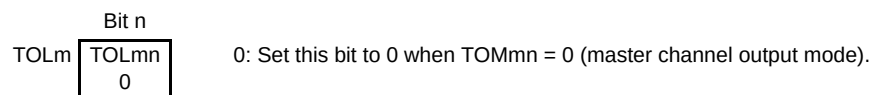
(b) Timer output register m (TOM)



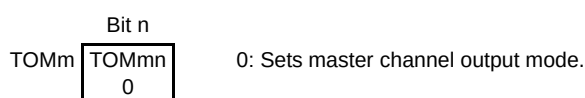
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 64 Procedure for Operations When the Input Signal High-/Low-Level Width Measurement Function is to be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOMn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait state is set.
	Detects the TImn pin input count start valid edge. →	Clears timer counter register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOMn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.8.6 Operation as a delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate INTTMmn (timer interrupt) at any interval by setting TSmn to 1 by software while TEMn = 1.

The interrupt generation period can be calculated by the following expression.

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

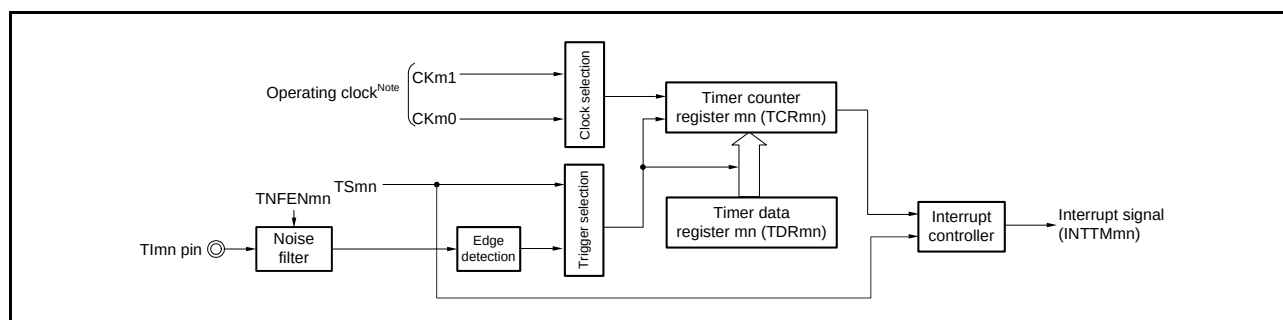
Timer counter register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSM) is set to 1, the TEMn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait state is set.

Timer counter register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

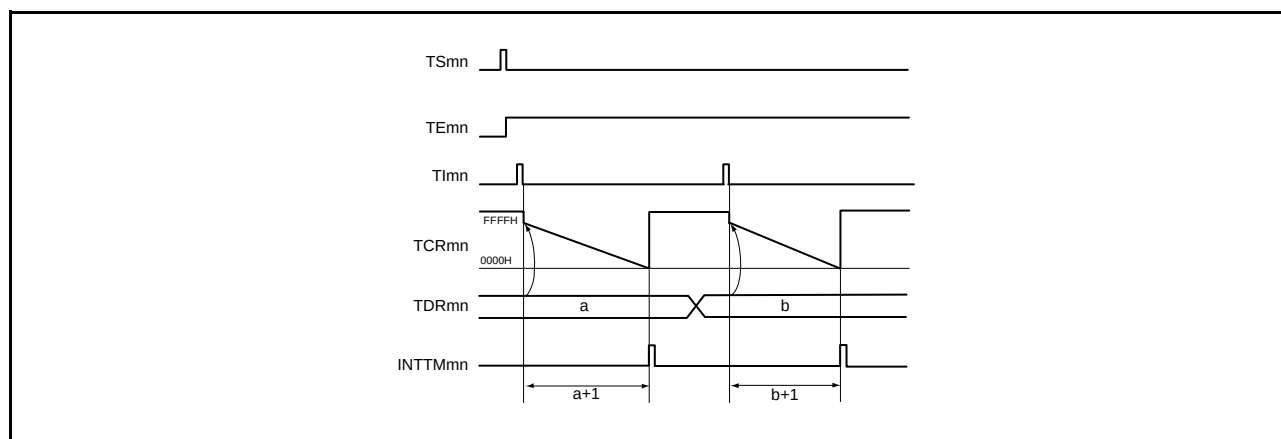
Figure 7 - 65 Block Diagram for Operation as a Delay Counter



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 66 Example of Basic Timing during Operation as a Delay Counter

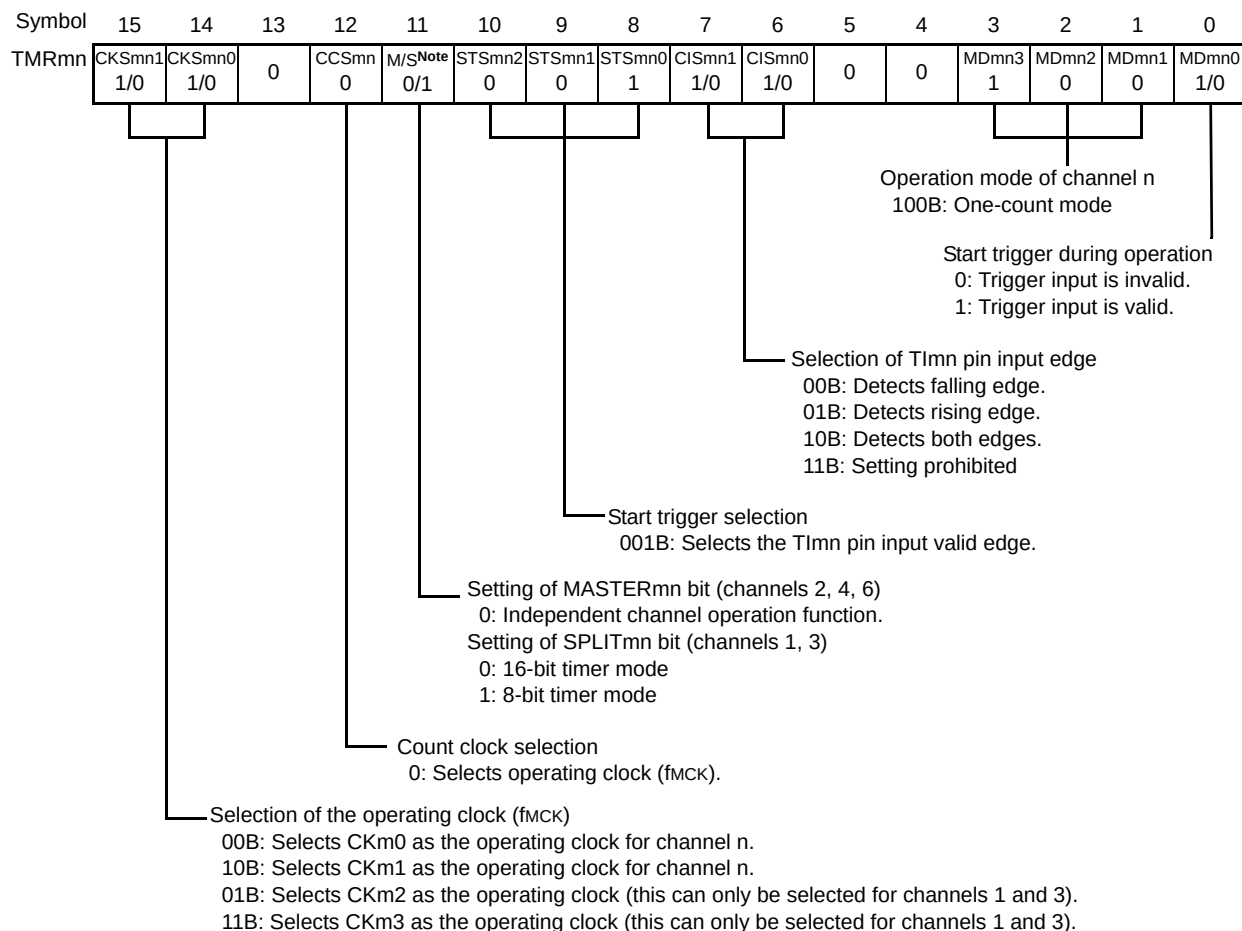


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

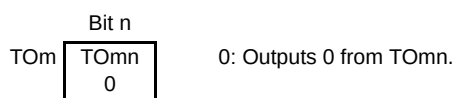
Remark 2. TSmn: Bit n of timer channel start register m (TSM)
TEmn: Bit n of timer channel enable status register m (TEM)
TImn: TImn pin input signal
TCRmn: Timer counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 67 Example of Register Settings for Operation as a Delay Counter

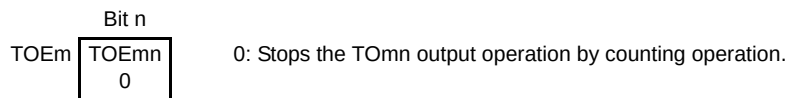
(a) Timer mode register mn (TMRmn)



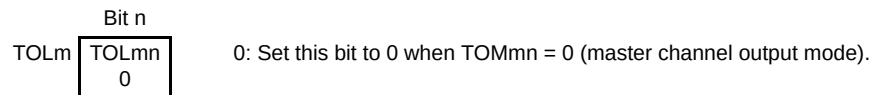
(b) Timer output register m (TOM)



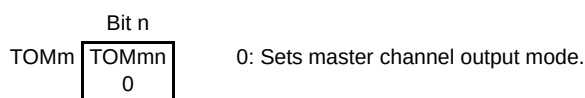
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 68 Procedure for Operations When the Delay Counter Function is to be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOMn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait state is set.
	The counter starts counting down by the next start trigger detection. • Detects the TImn pin input valid edge. → • Sets the TSmn bit to 1 by the software.	Value of the TDRmn register is loaded to the timer counter register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

7.9 Simultaneous Channel Operation Function of Timer Array Unit

7.9.1 Operation for the one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

$$\begin{aligned}\text{Delay time} &= \{\text{Set value of TDRmn (master)} + 2\} \times \text{Count clock period} \\ \text{Pulse width} &= \{\text{Set value of TDRmp (slave)}\} \times \text{Count clock period}\end{aligned}$$

The master channel operates in the one-count mode and counts the delays. Timer counter register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

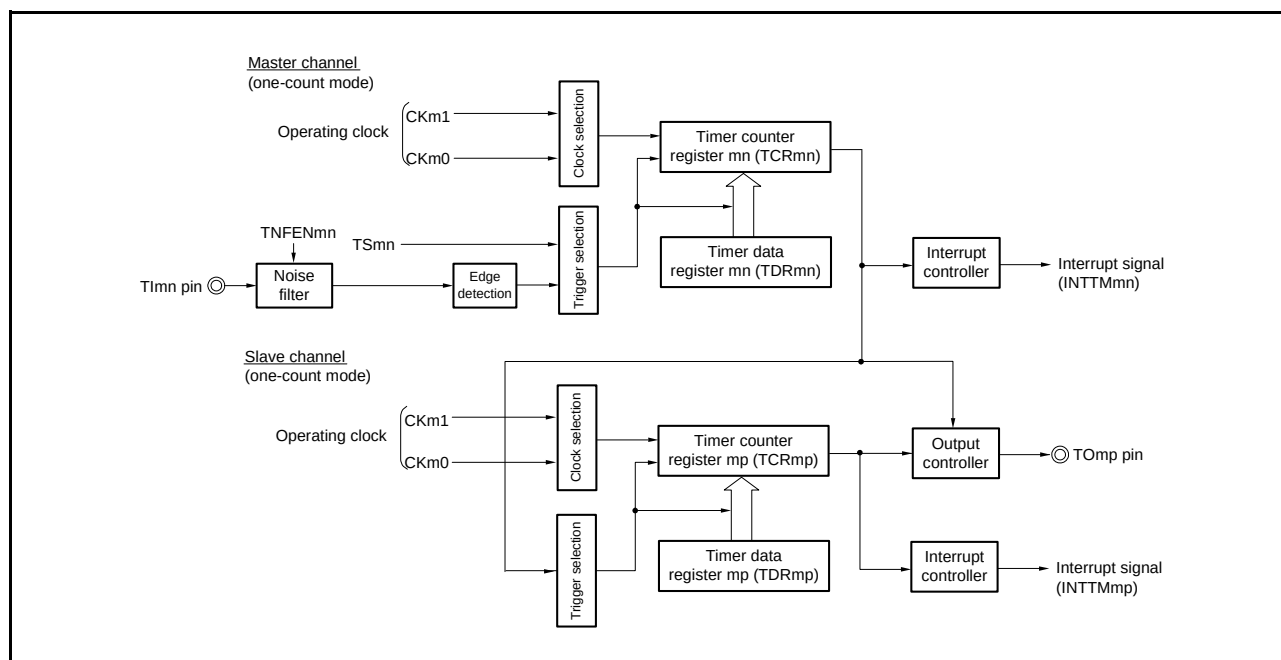
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution **The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

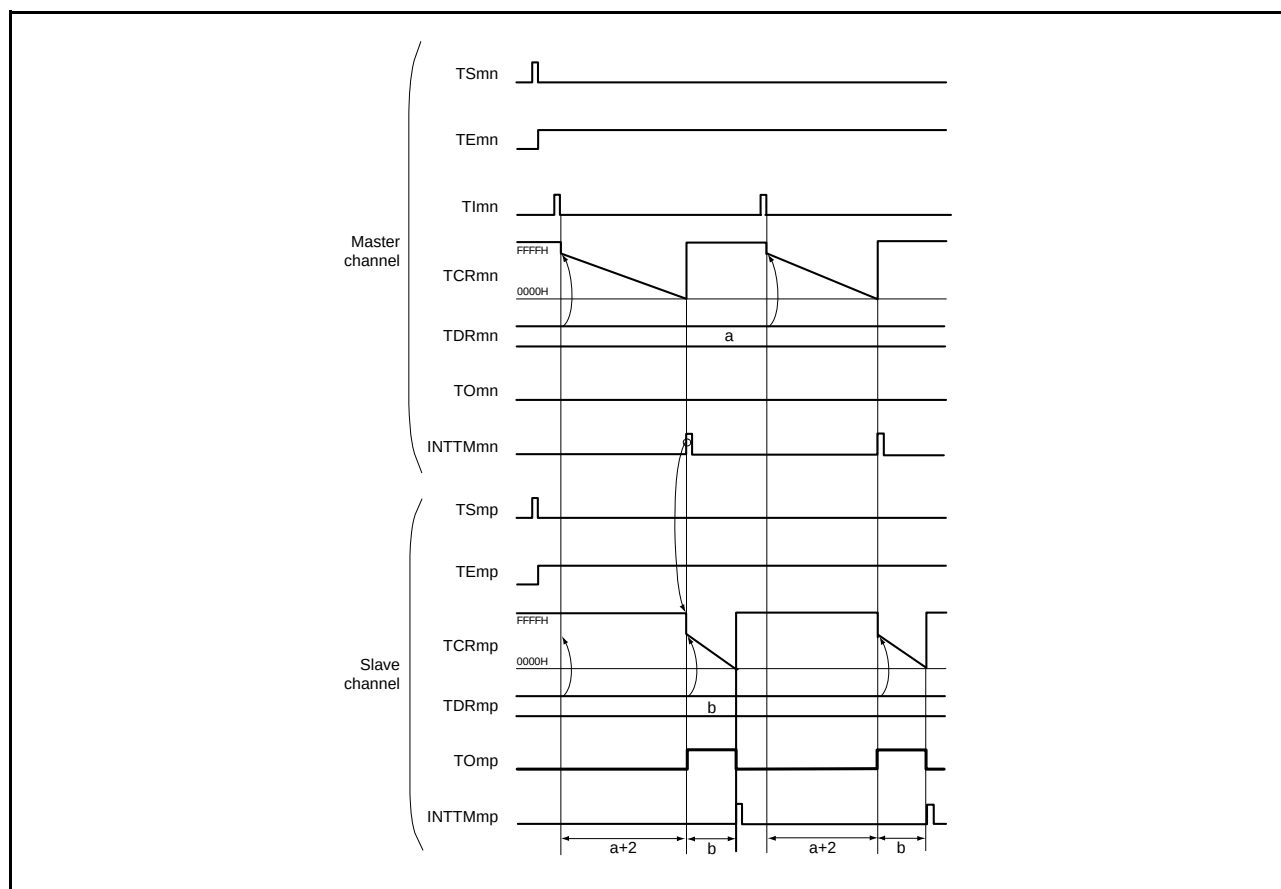
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 69 Block Diagram for Operation for the One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 70 Example of Basic Timing during Operation for the One-Shot Pulse Output Function



Remark 1. m: Unit number ($m = 0$), n: Master channel number ($n = 0, 2, 4, 6$)

p: Slave channel number ($n < p \leq 7$)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)

TEmn, TEm: Bits n and p of timer channel enable status register m (TEm)

TImn, TImp: Signals on the TImn and TImp input pins

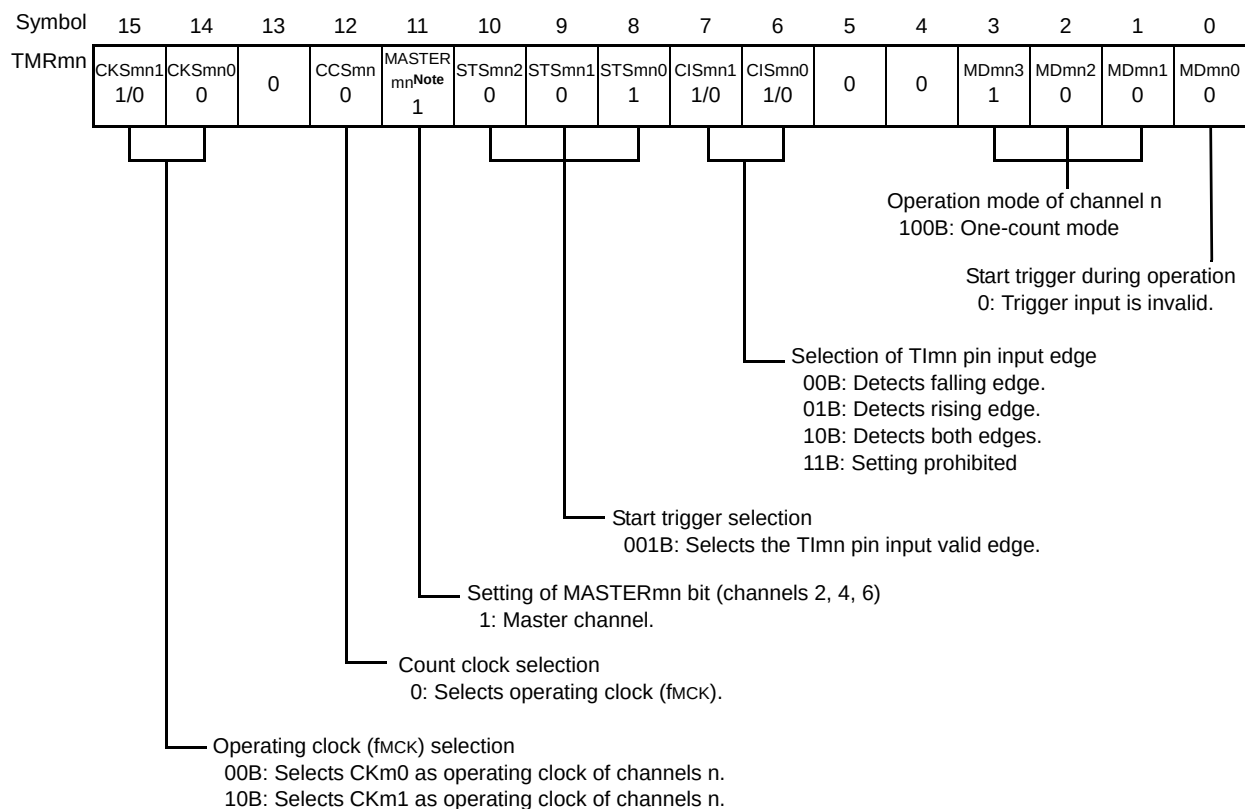
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

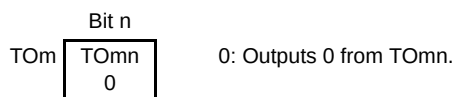
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 7 - 71 Example of Register Settings for the Master Channel When the One-Shot Pulse Output Function is to be Used

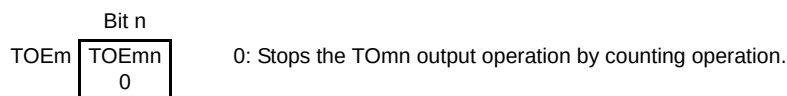
(a) Timer mode register mn (TMRmn)



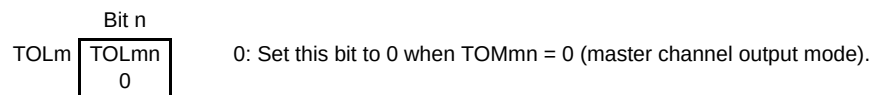
(b) Timer output register m (TOM)



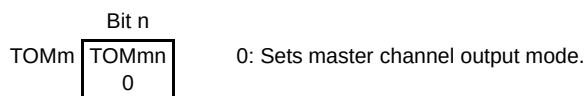
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

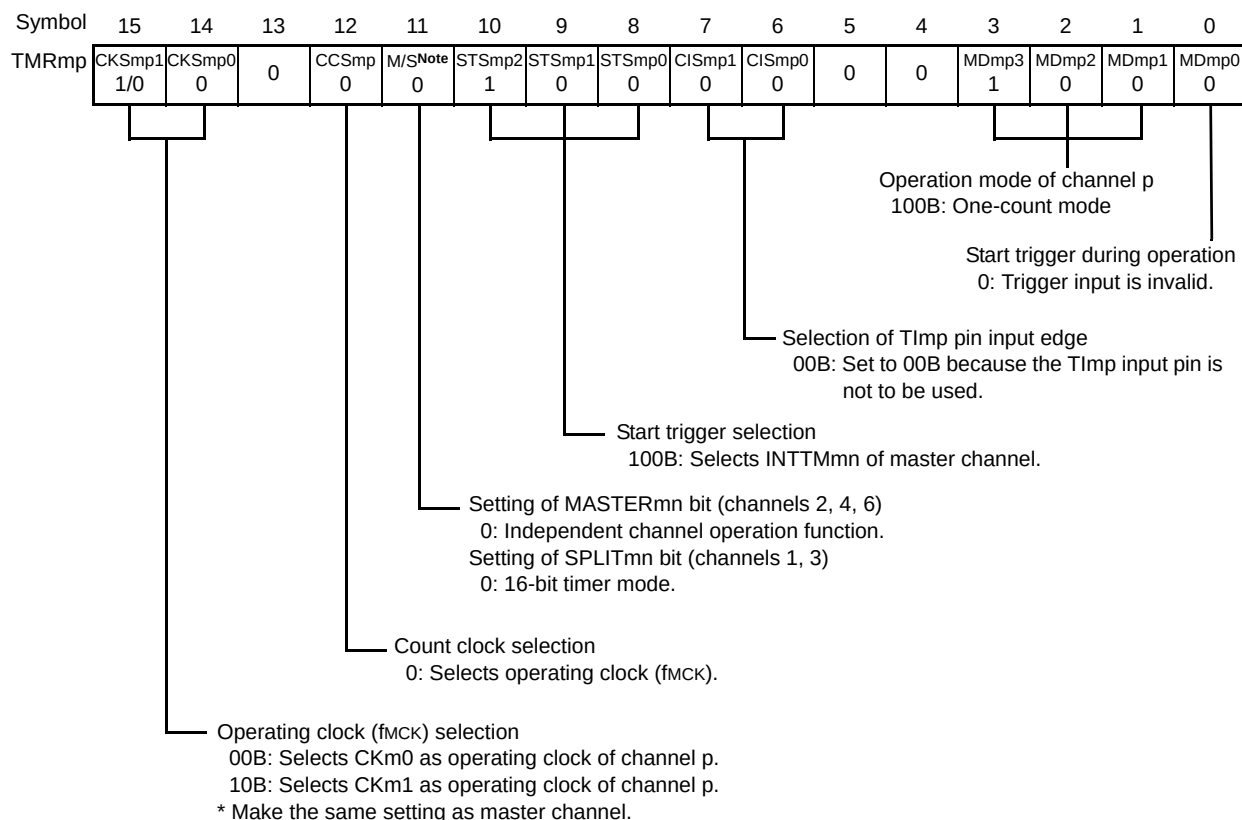


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 72 Example of Register Settings for the Slave Channel When the One-Shot Pulse Output Function is to be Used

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOM)

Bit p	
TOM	TOMP 1/0
0: Outputs 0 from TOMP. 1: Outputs 1 from TOMP.	

(c) Timer output enable register m (TOEm)

Bit p	
TOEm	TOEMP 1/0
0: Stops the TOMP output operation by counting operation. 1: Enables the TOMP output operation by counting operation.	

(d) Timer output level register m (TOLm)

Bit p	
TOLm	TOLMP 1/0
0: Positive logic output (active-high) 1: Negative logic output (active-low)	

(e) Timer output mode register m (TOMm)

Bit p	
TOMm	TOMMP 1
1: Sets the slave channel output mode.	

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
 TMRm1, TMRm3: SPLITmp bit
 TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
 p: Slave channel number (n < p ≤ 7)

Figure 7 - 73 Procedure for Operations When the One-Shot Pulse Output Function is to be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. → Sets the TOEmp bit to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

Figure 7 - 73 Procedure for Operations When the One-Shot Pulse Output Function is to be Used (2/2)

	Software Operation	Hardware State
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. —————→ The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEMn and TEm bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait state. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by softwareNote.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer counter register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. —————→ The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized and retains its current state.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. —————→	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. —————→ When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUMEN bit of the PER0 register is cleared to 0. —————→ Set the TAUMRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. —————→	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

7.9.2 Operation for the PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)} / {Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSM) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer counter register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTM) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

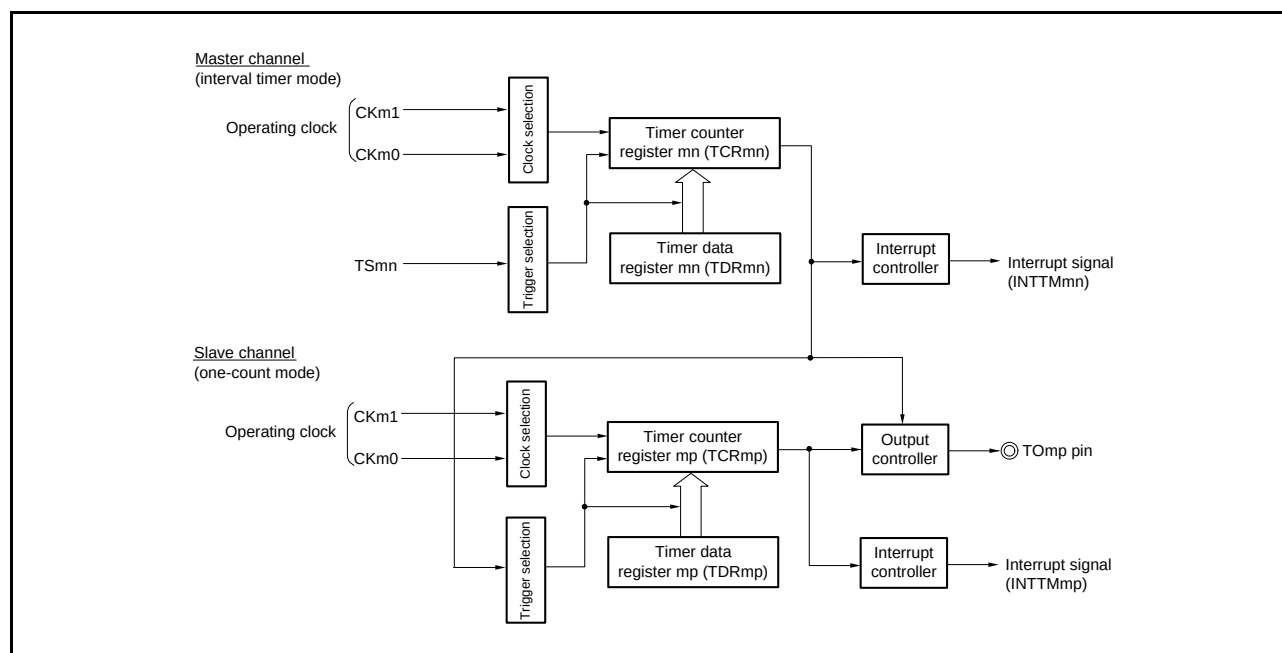
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

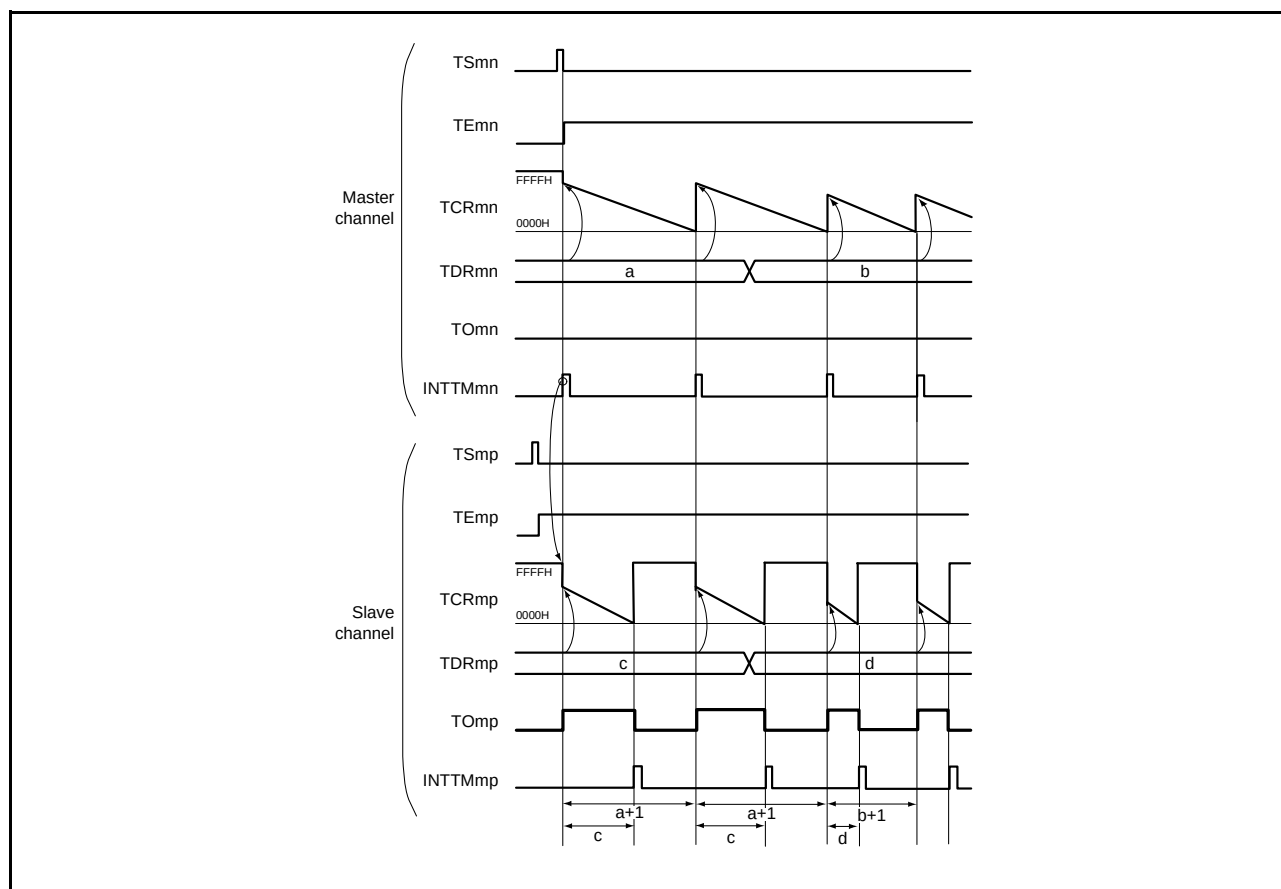
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 74 Block Diagram for Operation for the PWM Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 75 Example of Basic Timing during Operation for the PWM Function



Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

p: Slave channel number (n < p ≤ 7)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSM)

TEmn, TEmn: Bits n and p of timer channel enable status register m (TEM)

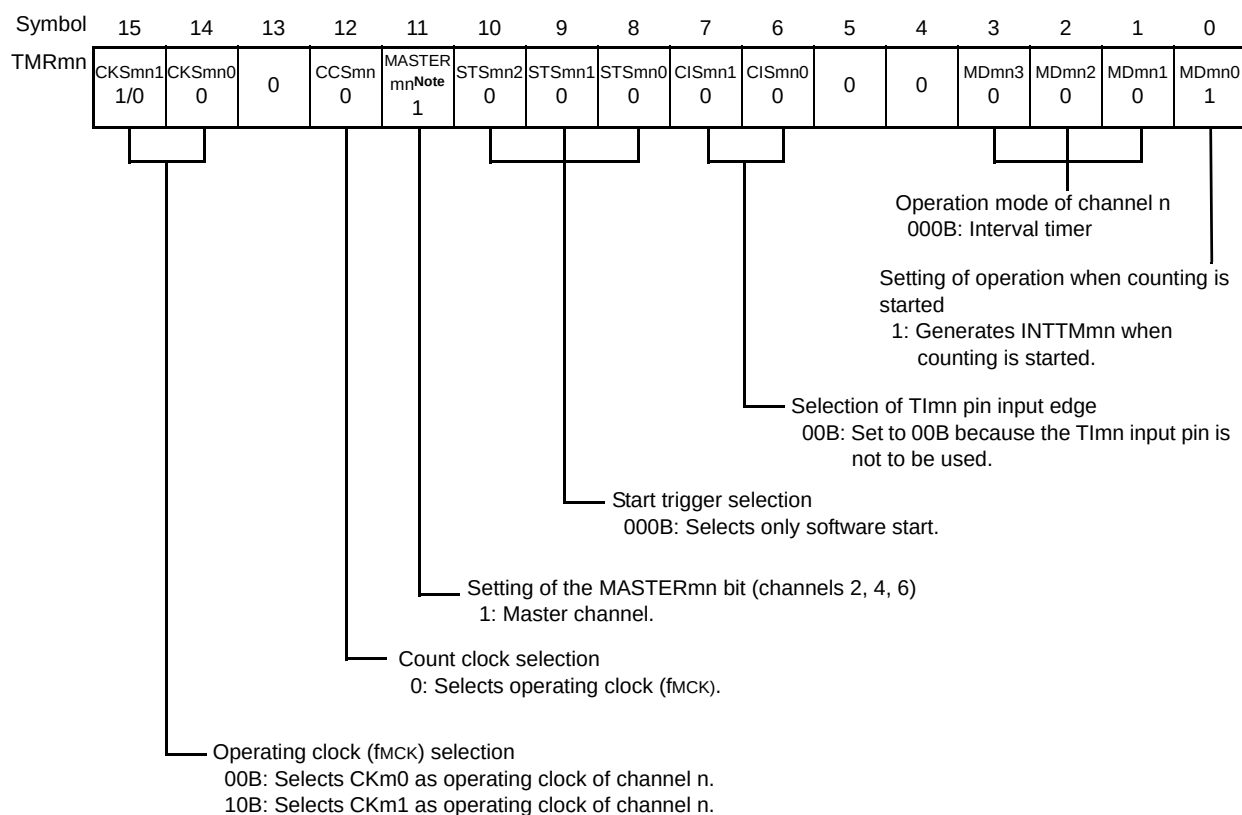
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

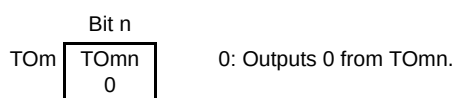
TOMn, TOMp: Signals on the TOMn and TOMp output pins

Figure 7 - 76 Example of Register Settings for the Master Channel When the PWM Function is to be Used

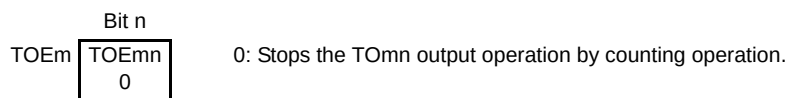
(a) Timer mode register mn (TMRmn)



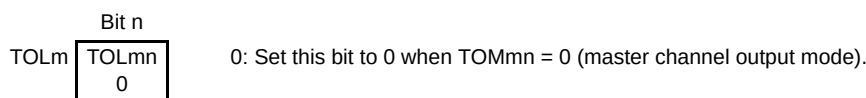
(b) Timer output register m (TOM)



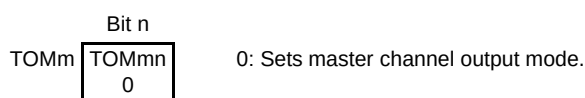
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

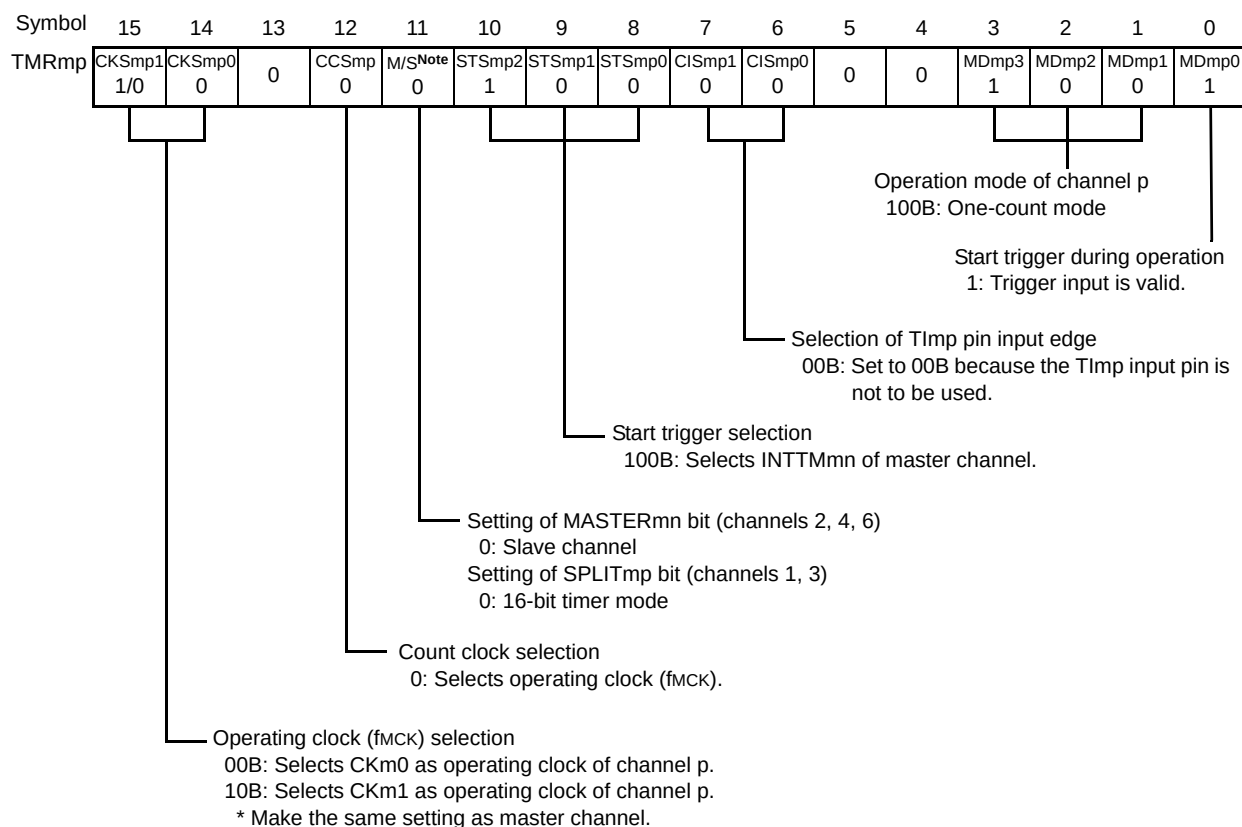


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 77 Example of Register Settings for the Slave Channel When the PWM Function is to be Used

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOM)

Bit p	
TOM	TOMP 1/0
	0: Outputs 0 from TOMP. 1: Outputs 1 from TOMP.

(c) Timer output enable register m (TOEm)

Bit p	
TOEm	TOEMP 1/0
	0: Stops the TOMP output operation by counting operation. 1: Enables the TOMP output operation by counting operation.

(d) Timer output level register m (TOLm)

Bit p	
TOLm	TOLMP 1/0
	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

Bit p	
TOMm	TOMMP 1
	1: Sets the slave channel output mode.

Note

TMRm2, TMRm4, TMRm6:	MASTERmn bit
TMRm1, TMRm3:	SPLITmp bit
TMRm5, TMRm7:	Fixed to 0

Remark

m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 78 Procedure for Operations When the PWM Function is to be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Clears the port register and port mode register to 0. →	TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

Figure 7 - 78 Procedure for Operations When the PWM Function is to be Used (2/2)

	Software Operation	Hardware State
Operation is resumed.	Operation start Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSM) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEm = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	During operation Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer counter register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
	Operation stop The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEm = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized and retains its current state.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
	TAU stop To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUMEN bit of the PER0 register is cleared to 0. → Set the TAUMRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

7.9.3 Operation for the multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

$$\begin{aligned}\text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period} \\ \text{Duty factor 1 [\%]} &= \{\text{Set value of TDRmp (slave 1)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{Duty factor 2 [\%]} &= \{\text{Set value of TDRmq (slave 2)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100\end{aligned}$$

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer counter register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods. The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

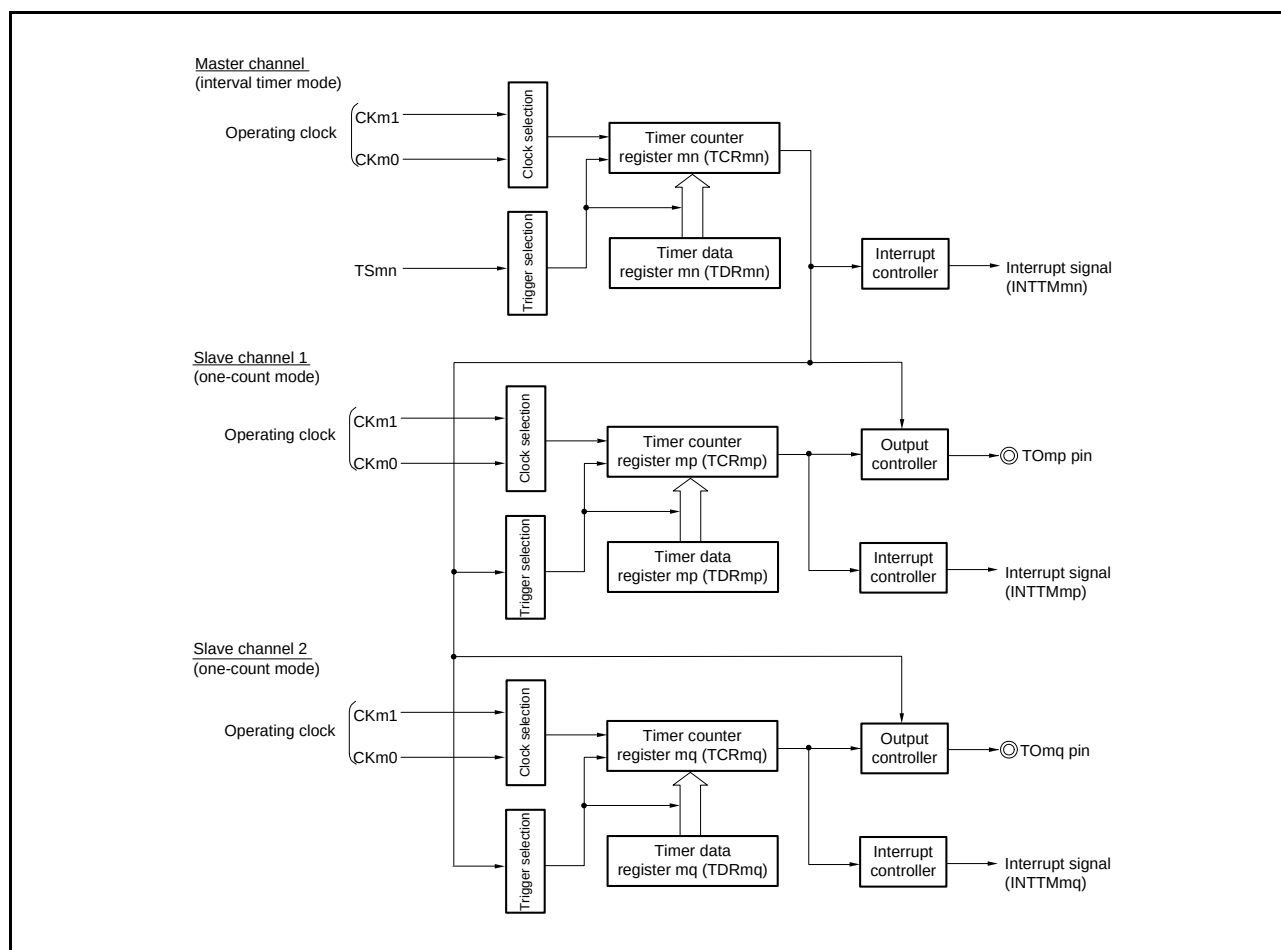
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).

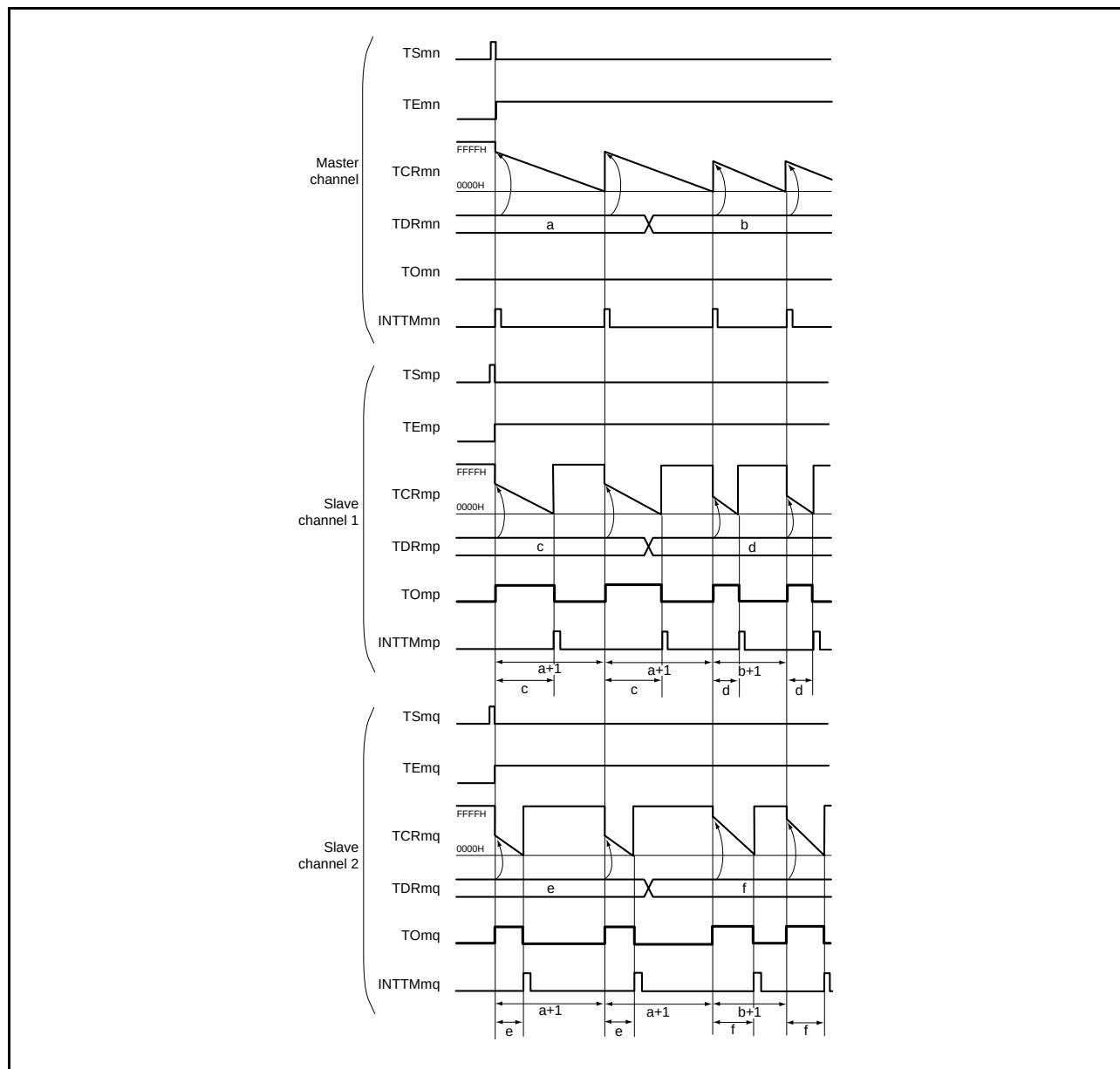
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
 $n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 7 - 79 Block Diagram for Operation for the Multiple PWM Output Function (for Two Types of PWM Output)



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
 $n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 7 - 80 Example of Basic Timing during Operation for the Multiple PWM Output Function
(for Two Types of PWM Output)



Remark 1. m: Unit number ($m = 0$), n: Master channel number ($n = 0, 2, 4$)

p: Slave channel number, q: Slave channel number

$n < p < q \leq 7$ (Where p and q are integers greater than n)

Remark 2. TS_{mn} , TS_{mp} , TS_{mq} : Bits n, p, and q of timer channel start register m (TS_m)

TE_{mn} , TE_{mp} , TE_{mq} : Bits n, p, and q of timer channel enable status register m (TE_m)

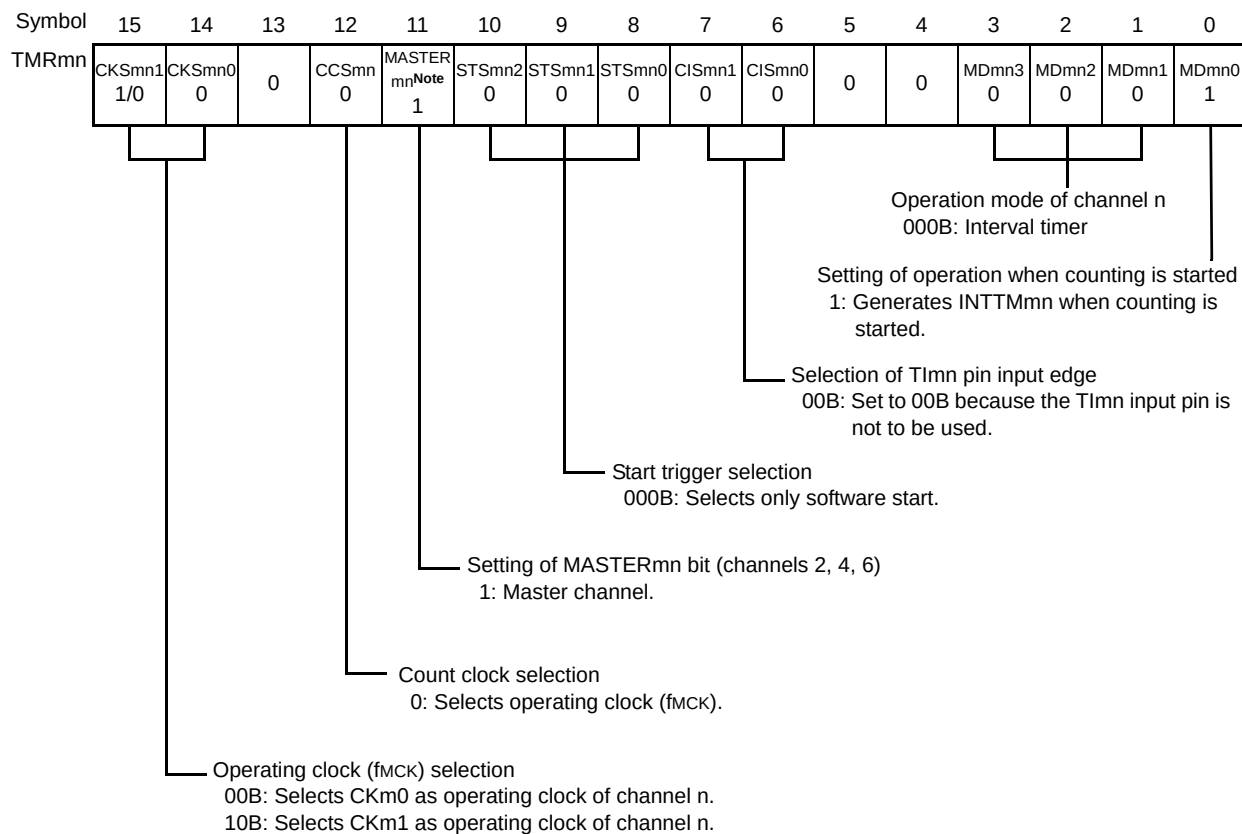
TCR_{mn} , TCR_{mp} , TCR_{mq} : Timer counter registers mn, mp, mq (TCR_{mn} , TCR_{mp} , TCR_{mq})

TDR_{mn} , TDR_{mp} , TDR_{mq} : Timer data registers mn, mp, mq (TDR_{mn} , TDR_{mp} , TDR_{mq})

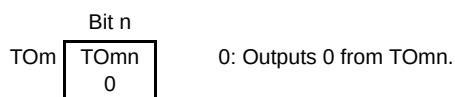
TOM_{mn} , TOM_{mp} , TOM_{mq} : Signals on the TOM_{mn} , TOM_{mp} , and TOM_{mq} output pins

Figure 7 - 81 Example of Register Settings for the Master Channel When the Multiple PWM Output Function is to be Used

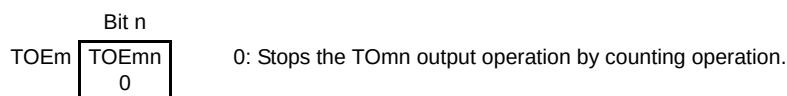
(a) Timer mode register mn (TMRmn)



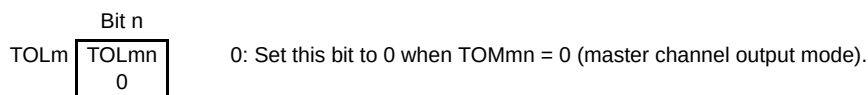
(b) Timer output register m (TOM)



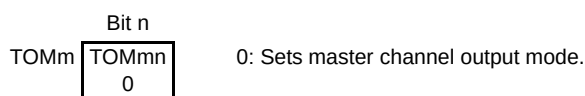
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

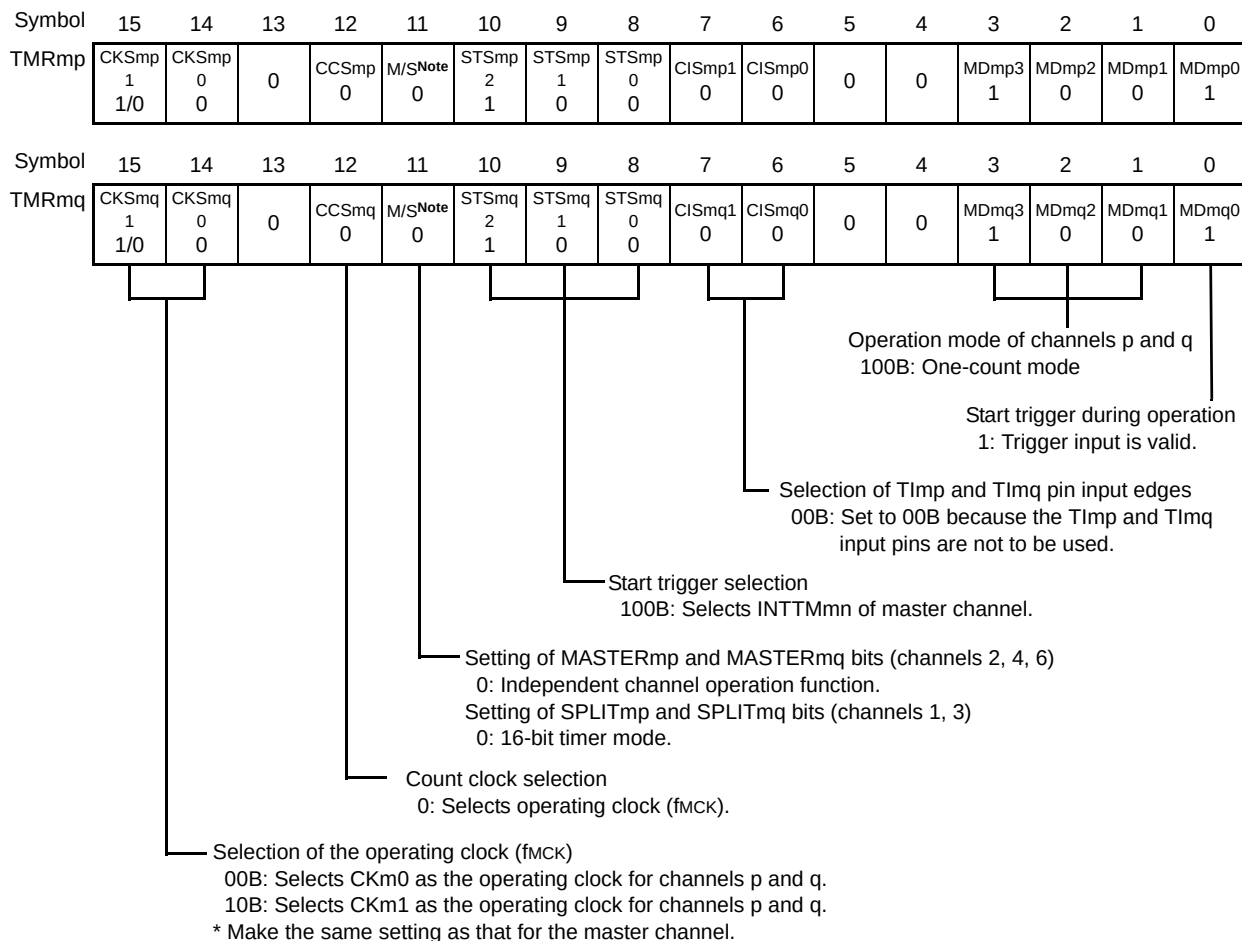


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)

Figure 7 - 82 Example of Register Settings for the Slave Channel When the Multiple PWM Output Function is to be Used (for Two Types of PWM Output)

(a) Timer mode registers mp and mq (TMRmp and TMRmq)



(b) Timer output register m (TOM)

	Bit q	Bit p	
TOM	TOMq 1/0	TOMP 1/0	0: Outputs 0 from TOMP or TOMq. 1: Outputs 1 from TOMP or TOMq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p	
TOEm	TOEmq 1/0	TOEmp 1/0	0: Stops the TOMP or TOMq output operation by counting operation. 1: Enables the TOMP or TOMq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p	
TOLm	TOLmq 1/0	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p	
TOMm	TOMmq 1	TOMmp 1	1: Sets the slave channel output mode.

(Note and Remark are listed on the next page.)

Note TMRm2, TMRm4, TMRm6: MASTERmp and MASTERmq bits
 TMRm1, TMRm3: SPLITmp and SPLITmq bits
 TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
 p: Slave channel number, q: Slave channel number
 $n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 7 - 83 Procedure for Operations When the Multiple PWM Output Function is to be Used (for Two Types of PWM Output) (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, 0q (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Sets the TOLmp and TOLmq bits. Sets the TOmp and TOMq bits and determines default level of the TOmp and TOMq outputs. →	The TOmp and TOMq pins go into Hi-Z output state. The TOmp and TOMq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOMq. →	TOmp and TOMq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOMq pins output the TOmp and TOMq set levels.

(Remark is listed on the next page.)

Figure 7 - 83 Procedure for Operations When the Multiple PWM Output Function is to be Used (for Two Types of PWM Output) (2/2)

	Software Operation	Hardware State
Operation is resumed.	Operation start (Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSM) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	During operation Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSR0q registers are not used.	The counter of the master channel loads the TDRmn register value to timer counter register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOMq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
	Operation stop The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. → The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOMq outputs are not initialized and retain their current states.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOMq bits. →	The TOmp and TOMq pins output the TOmp and TOMq set levels.
	TAU stop To hold the TOmp and TOMq pin output levels Clears the TOmp and TOMq bits to 0 after the value to be held is set to the port register. → When holding the TOmp and TOMq pin output levels are not necessary Setting not required	The TOmp and TOMq pin output levels are held by port function.
	The TAUMEN bit of the PER0 register is cleared to 0. → Set the TAUMRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOMq bits are cleared to 0 and the TOmp and TOMq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are consecutive integers greater than n)

7.10 Cautions When Using Timer Array Unit

7.10.1 Cautions when using timer output

Pins may be assigned multiplexed timer output and other alternate functions. The assignment depends on the product. If you intend to use a timer output, set the outputs from all other multiplexed pin functions to their initial values. For details, see **4.5 Register Settings When Using Alternate Function**.

7.10.2 Point for caution when a timer output is to be used as an event input for the ELC

The timer outputs (TO00 to TO03) of channels 0 to 3 of timer array unit 0 can be used as event inputs for the event link controller (ELC).

Section 8 Realtime Clock (RTC)

8.1 Functions of Realtime Clock

The realtime clock has the following features.

- Capable of counting years, months, days of the week, dates, hours, minutes, and seconds, for up to 99 years
- Fixed-cycle interrupt (with period selectable from among 0.5 of a second, 1 second, 1 minute, 1 hour, 1 day, or 1 month)
- Alarm interrupt (alarm set by day of week, hour, and minute)
- Pin output function of 1 Hz

The realtime clock interrupt signal (INTRTC) can be used to wake up the MCU from the STOP mode, or to trigger transitions of the A/D converter to the SNOOZE mode.

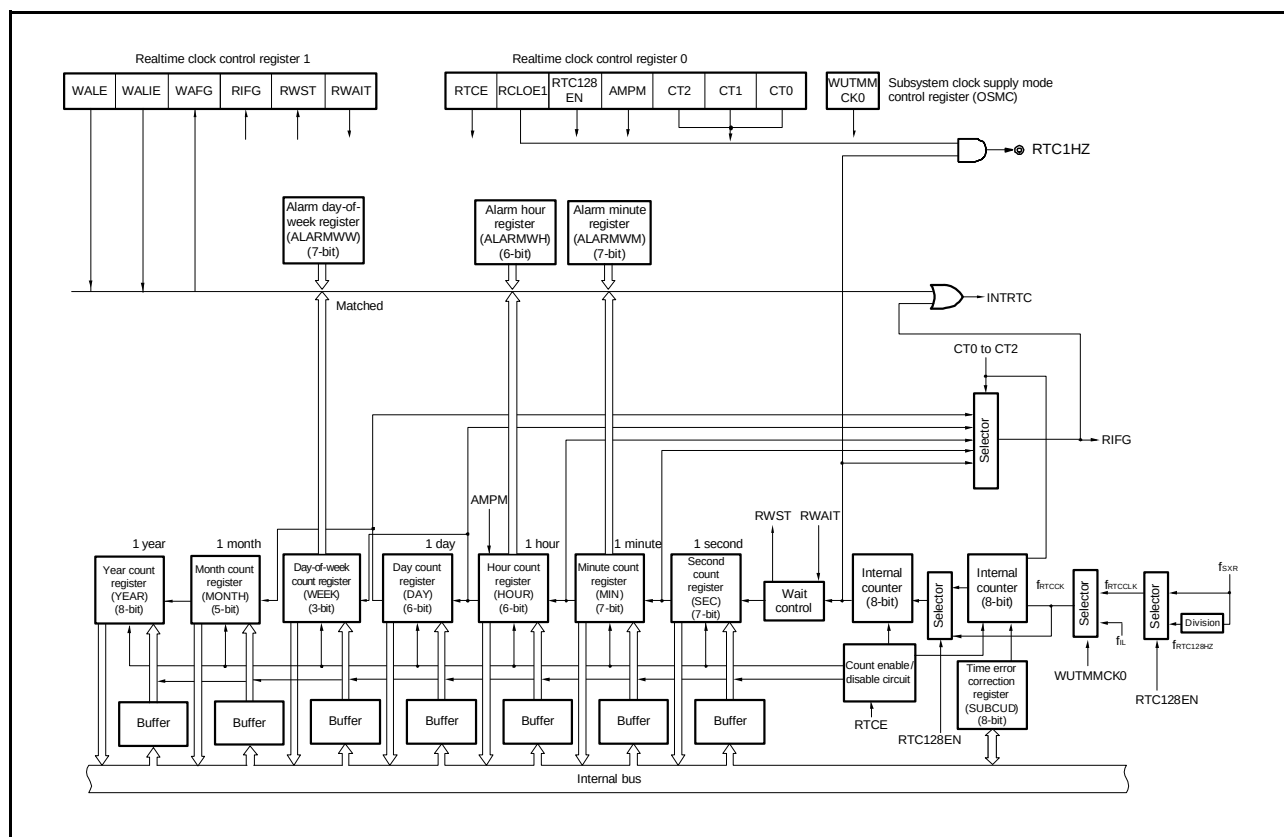
8.2 Configuration of the Realtime Clock

The realtime clock includes the following hardware blocks.

Table 8 - 1 Configuration of the Realtime Clock

Item	Configuration
Counter	Internal counter (16 bits)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Realtime clock control register 0 (RTCC0)
	Realtime clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Day-of-week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Time error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm day-of-week register (ALARMWW)

Figure 8 - 1 Block Diagram of the Realtime Clock



Caution The count of years, months, weeks, days, hours, minutes, and seconds can only proceed when a subsystem clock ($f_{sxr} = 32.768 \text{ kHz}$) is selected as the operating clock of the realtime clock. When the low-speed on-chip oscillator clock ($f_{il} = 32.768 \text{ kHz}$) is selected, only the fixed-cycle interrupt is available.

8.3 Registers for Controlling the Realtime Clock

The following registers are used to control the realtime clock.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Realtime clock control register 0 (RTCC0)
- Realtime clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Day-of-week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Time error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm day-of-week register (ALARMWW)
- Port mode register 3 (PM3)
- Port register 3 (P3)

The following shows the register states depending on reset sources.

Reset Source	System-related registers ^{Note 1}	Calendar-related registers ^{Note 2}
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset sources	Retained	Retained

Note 1. RTCC0, RTCC1, and SUBCUD

Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, and ALARMWW

Assertion of the reset signal does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW register. Initialize all the registers after power on.

8.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the realtime clock is to be used, be sure to set bit 7 (RTCWEN) of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 8 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCWEN	Control of access to the realtime clock (RTC)
0	- SFR used by the realtime clock (RTC) cannot be written. - Operation of the realtime clock (RTC) can be enabled.
1	- SFR used by the realtime clock (RTC) can be read and written. - Operation of the realtime clock (RTC) can be enabled.

Caution 1. When the realtime clock is to be used, start by setting the RTCWEN bit to 1 and then set the following registers once oscillation of the counter clock (fRTCK) has become stable.

- Realtime clock control register 0 (RTCC0)
- Realtime clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Day-of-week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Time error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm day-of-week register (ALARMWW)

If the setting of RTCWEN is 0, attempted writing to the control registers of the realtime clock is ignored, and 00H is read. Note, however, writing to the following registers is valid.

- Subsystem clock supply mode control register (OSMC)
- Port mode register 3 (PM3)
- Port register 3 (P3)

Caution 2. The subsystem clock supply to peripheral functions other than the realtime clock can be stopped in STOP mode or HALT mode when the subsystem clock is in use, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

8.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the control clock (fRTCK) for the realtime clock. The RTCLPC bit of this register can be used to reduce the power consumption by disabling supply of the clock signal to peripheral functions that are not in use. For details about setting the RTCLPC bit, see **Section 6 Clock Generator**. The OSMC register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 8 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined
R/W: R/W

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	HIPREC
WUTMMCK0		Selection of the operating clock (fRTCK) for the control block of the realtime clock						
0		Subsystem clock XR (fsXR) or fRTC128HZ (fRTC128HZ is selected when the setting of the RTC128EN bit is 1.)						
1		Low-speed on-chip oscillator clock (fIL) ^{Notes 1, 2}						

Note 1. Setting the WUTMMCK0 bit to 1 is prohibited when the subsystem clock X is selected as the operating clock for the control block of the realtime clock.

Note 2. Switching between the subsystem clock and the low-speed on-chip oscillator clock by using the WUTMMCK0 bit is only possible while operations of the realtime clock, 32-bit interval timer, serial interface UARTA0, and clock output/buzzer output controller are all stopped.

Caution Counting of years, months, weeks, days, hours, minutes, and seconds can only proceed when the subsystem clock XR (fsXR = 32.768 kHz) or fRTC128HZ is selected as the operating clock for the control block of the realtime clock. When the low-speed on-chip oscillator clock (fIL = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

8.3.3 Realtime clock control register 0 (RTCC0)

The RTCC0 is an 8-bit register that is used to start or stop the realtime clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the fixed-cycle interrupt. The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (1/2)

Address: F022BH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RTC128EN	AMPM	CT2	CT1	CT0

RTCE	Note	Realtime clock operation control
0		Stops counter operation.
1		Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

RTC128EN	Selection of the operating clock for the realtime clock (fRTCCLK)
0	32.768 kHz
1	128 Hz

- Setting this bit to 1 enables the realtime clock to operate with the 128-Hz clock for lower-power operation.
- Time error correction cannot be used when the setting of this bit is 1.
- The WUTMMCK bit in the OSMC register should be set to 0 when setting this bit to 1.

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system

- Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of the realtime clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system.
- Table 8 - 2** shows the time (hour) digits indicated according to the setting of this bit.

Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (2/2)

CT2	CT1	CT0	Fixed-cycle interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)

To change the values of the CT2 to CT0 bits while counting is in progress (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt processing after clearing the RIFG and RTCIF flags.

Note To shift to the STOP mode immediately after setting the RTCE bit to 1, follow the procedure in **Figure 8 - 20 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1.**

Caution 1. Do not change the value of the RCLOE1 bit when RTCE is 1.

Caution 2. 1 Hz is not output even if RCLOE1 is set to 1 when RTCE is 0.

Caution 3. Be sure to set bit 6 to 0.

Remark ×: Don't care

8.3.4 Realtime clock control register 1 (RTCC1)

The RTCC1 is an 8-bit register that is used to control the alarm interrupt and the wait time of the counter. The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (1/2)

Address: F022CH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while counting is in progress (RTCE = 1) and WALIE is 1, rewrite the WALE bit after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting any of the alarm-related registers (WALIE flag of realtime clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm day-of-week register (ALARMWW)), set the WALE bit to 0 to disable matching.	

WALIE	Control of alarm interrupt (INTRTC)
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is only valid when WALE is 1 and is set to 1 one cycle of f_{RTCC} after matching of the alarm is detected. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

RIFG	Fixed-cycle interrupt status flag
0	Fixed-cycle interrupt is not generated.
1	Fixed-cycle interrupt is generated.
This flag indicates the status of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to 1. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (2/2)

RWST	Wait status flag of realtime clock ^{Note 3}
0	Counting is in progress.
1	Counter values are readable and writable.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of realtime clock
0	Counting proceeds.
1	Stops the SEC to YEAR counters. Counter values are readable and writable.
This bit controls the operation of the counter. Be sure to write 1 to this bit to read or write the counter value. So that the 16-bit internal counter continues to run, return the value of this bit to 0 on completion of reading or writing within one second. After setting this bit to 1, it takes up to one cycle of f _{RTCK} until the counter value can be actually read or written (RWST = 1). ^{Notes 1, 2} When the internal counter (16 bits) overflows while the setting of this bit is 1, an indicator of the counter having overflowed is retained after RWAIT has become 0, after which counting up continues. Note that, when the second count register has been written to, the overflow is not retained.	

Note 1. When the RWAIT bit is set to 1 within one cycle of f_{RTCK} clock after setting the RTCE bit to 1, the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCK}).

Note 2. When the RWAIT bit is set to 1 within one cycle of f_{RTCK} clock after release from the standby mode (HALT mode, STOP mode, or SNOOZE mode), the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCK}).

Note 3. Bit 1 is read-only.

Caution Note that using a bit manipulation instruction for writing to the RTCC1 register may lead to clearing of the RIFG and WAFG flags. Therefore, when writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting 1 to the corresponding bit. However, if the RIFG and WAFG flags are not in use and a change to the value does not matter, using a bit manipulation instruction for writing to the RTCC1 register does not create a problem.

Remark 1. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

8.3.5 Second count register (SEC)

The SEC is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. This counter is incremented each time the internal counter (16-bit) overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCLK} later. Set a decimal value of 00 to 59 to this register in BCD code. The SEC register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 6 Format of Second Count Register (SEC)

Address: F0220H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When reading from or writing to this register while the counter is in operation ($RTCE = 1$), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

8.3.6 Minute count register (MIN)

The MIN is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. This counter is incremented each time the second counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCLK} later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code. The MIN register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 7 Format of Minute Count Register (MIN)

Address: F0221H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When reading from or writing to this register while the counter is in operation ($RTCE = 1$), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

8.3.7 Hour count register (HOUR)

The HOUR is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours. This counter is incremented each time the minute counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of `RTCCLK` later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of the realtime clock control register 0 (RTCC0). If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system. The HOUR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 8 Format of Hour Count Register (HOUR)

Address: F0222H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Caution 2. When reading from or writing to this register while the counter is in operation (`RTCE = 1`), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Table 8 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 8 - 2 Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is 0 and to 24-hour display when the AMPM bit is 1. In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

8.3.8 Day count register (DAY)

The DAY is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. This counter is incremented each time the hour counter overflows. Counting by the date counter proceeds as shown below.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCLK} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code. The DAY register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 9 Format of Day Count Register (DAY)

Address: F0224H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When reading from or writing to this register while the counter is in operation ($RTCE = 1$), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

8.3.9 Day-of-week count register (WEEK)

The WEEK is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of days of the week. This counter is incremented in synchronization with the date counter. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCLK} later. Set a decimal value of 00 to 06 to this register in BCD code. The WEEK register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 10 Format of Day-of-Week Count Register (WEEK)

Address: F0223H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the day-of-week count register (WEEK) automatically. After reset release, set the day-of-week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Caution 2. When reading from or writing to this register while the counter is in operation ($RTCE = 1$), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

8.3.10 Month count register (MONTH)

The MONTH is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months. This counter is incremented each time the day counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of `FRTCCK` later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. The MONTH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 11 Format of Month Count Register (MONTH)

Address: F0225H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When reading from or writing to this register while the counter is in operation (`RTCE = 1`), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

8.3.11 Year count register (YEAR)

The YEAR is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the value of the counter of years. This counter is incremented each time the month count register (MONTH) overflows. Values 00, 04, 08, ..., 92, and 96 indicate a leap year. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of `FRTCCK` later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. The YEAR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 12 Format of Year Count Register (YEAR)

Address: F0226H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When reading from or writing to this register while the counter is in operation (`RTCE = 1`), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

8.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm. The ALARMWM register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 8 - 14 Format of Alarm Minute Register (ALARMWM)

Address: F0228H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

8.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm. The ALARMWH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 8 - 15 Format of Alarm Hour Register (ALARMWH)

Address: F0229H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

8.3.15 Alarm day-of-week register (ALARMWW)

This register is used to set days of the week of alarm. The ALARMWW register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 16 Format of Alarm Day-of-Week Register (ALARMWW)

Address: F022AH
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day of week							12-Hour Display				24-Hour Display			
	Sunday W W 0	Monday W W 1	Tuesday W W 2	Wednesday W W 3	Thursday W W 4	Friday W W 5	Saturday W W 6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

8.3.16 Port mode register 3 (PM3)

The PM3 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is FFH. When using the port 3 as the RTC1HZ pin for output of 1 Hz, set the PM30 bit to 0.

Figure 8 - 17 Format of Port Mode Register 3 (PM3)

Address: FFF23H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

8.3.17 Port register 3 (P3)

The P3 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H. When using the port 3 as 1-Hz output to the RTC1Hz pin, set the P30 bit to 0.

Figure 8 - 18 Format of Port Register 3 (P3)

Address: FFF03H

After reset: 00H

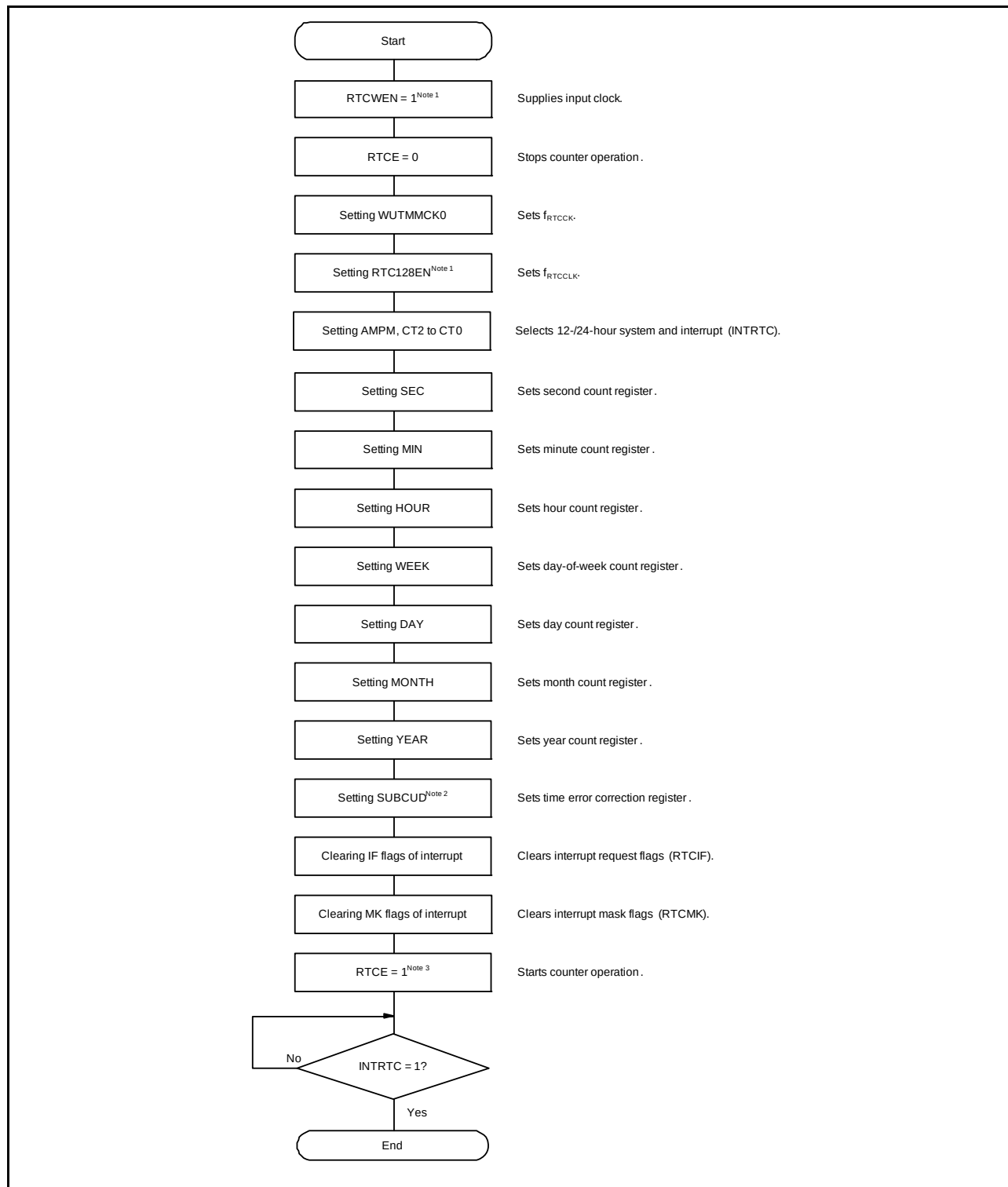
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	0	0	0	P31	P30

8.4 Operations of the Realtime Clock

8.4.1 Starting the realtime clock operation

Figure 8 - 19 Procedure for Starting the Realtime Clock Operation



(Notes are listed the next page.)

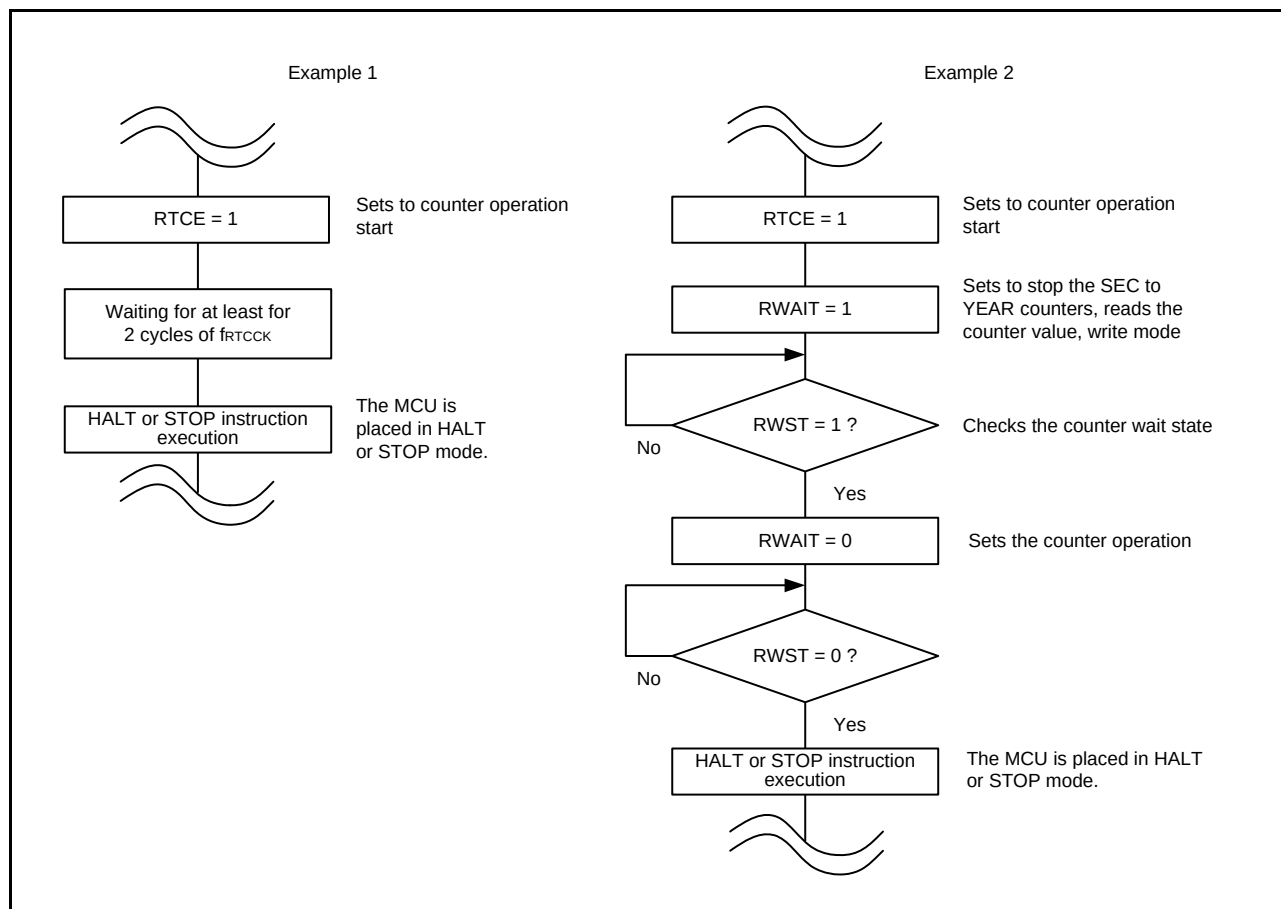
- Note 1.** First set the RTCWEN bit to 1 and set the RTC128EN bit as desired, while oscillation of the count clock (f_{RTCC}) is stable.
- Note 2.** Set up the SUBCUD register only if the time error must be corrected. For details about how to calculate the correction value, see **8.4.6 Example of time error correction by the realtime clock**.
Time error correction cannot be used while the setting of the RTC128EN bit is 1.
- Note 3.** Confirm the procedure described in **8.4.2 Shifting to HALT or STOP mode after starting operation** when shifting to HALT or STOP mode without waiting for INTRTC = 1 after RTCE = 1.

8.4.2 Shifting to HALT or STOP mode after starting operation

Take either of the steps listed below when shifting to HALT or STOP mode immediately after setting the RTCE bit to 1. Note that any of these steps is not required when shifting to the HALT or STOP mode after the INTRTC interrupt has occurred.

- Make a transition to HAL or STOP mode when at least two counter clock cycles (f_{RTCCK}) have elapsed after setting the RTCE bit to 1 (see **Figure 8 - 20, Example 1**).
- After setting the RTCE bit to 1 and then setting the RWAIT bit to 1, poll the RWST bit to check if it has become 1 yet. After setting the RWAIT bit to 0 and polling the RWST bit to check if it has become 0 yet, a transition to HALT/STOP mode will proceed (see **Figure 8 - 20, Example 2**).

Figure 8 - 20 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1

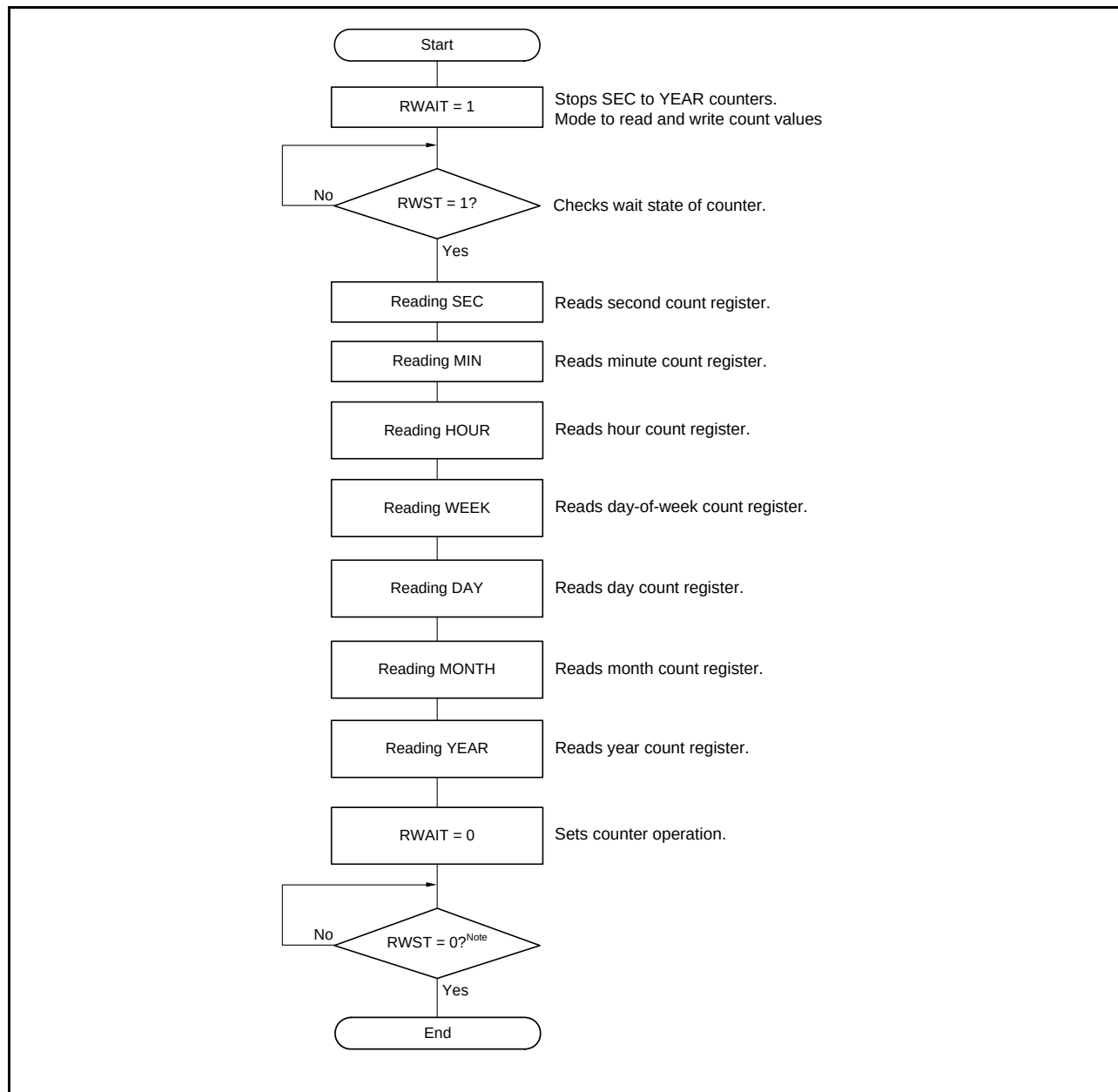


8.4.3 Reading from and writing to the counters of the realtime clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 8 - 21 Procedure for Reading Realtime Clock

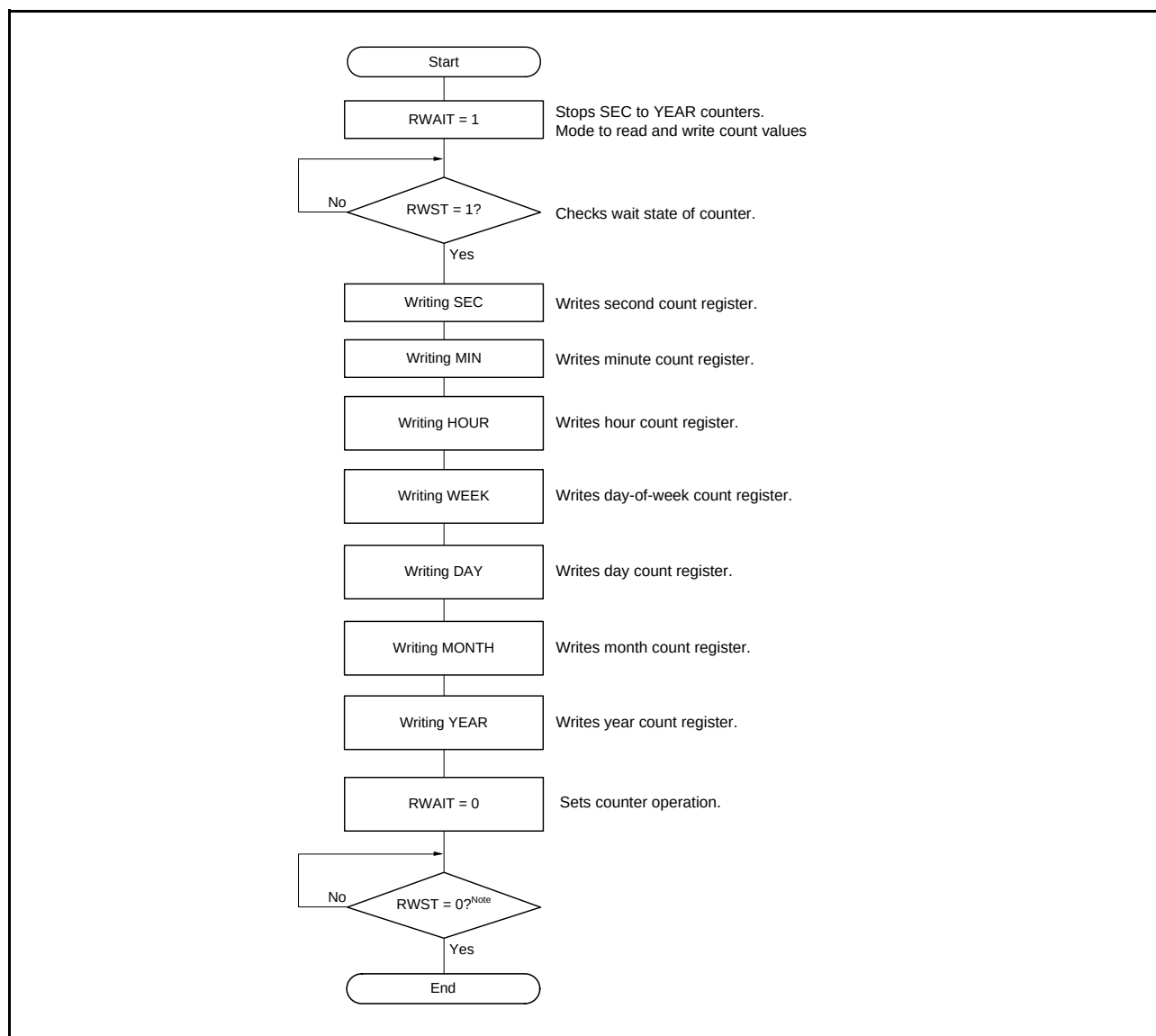


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to read and only some registers may be read.

Figure 8 - 22 Procedure for Writing Realtime Clock



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

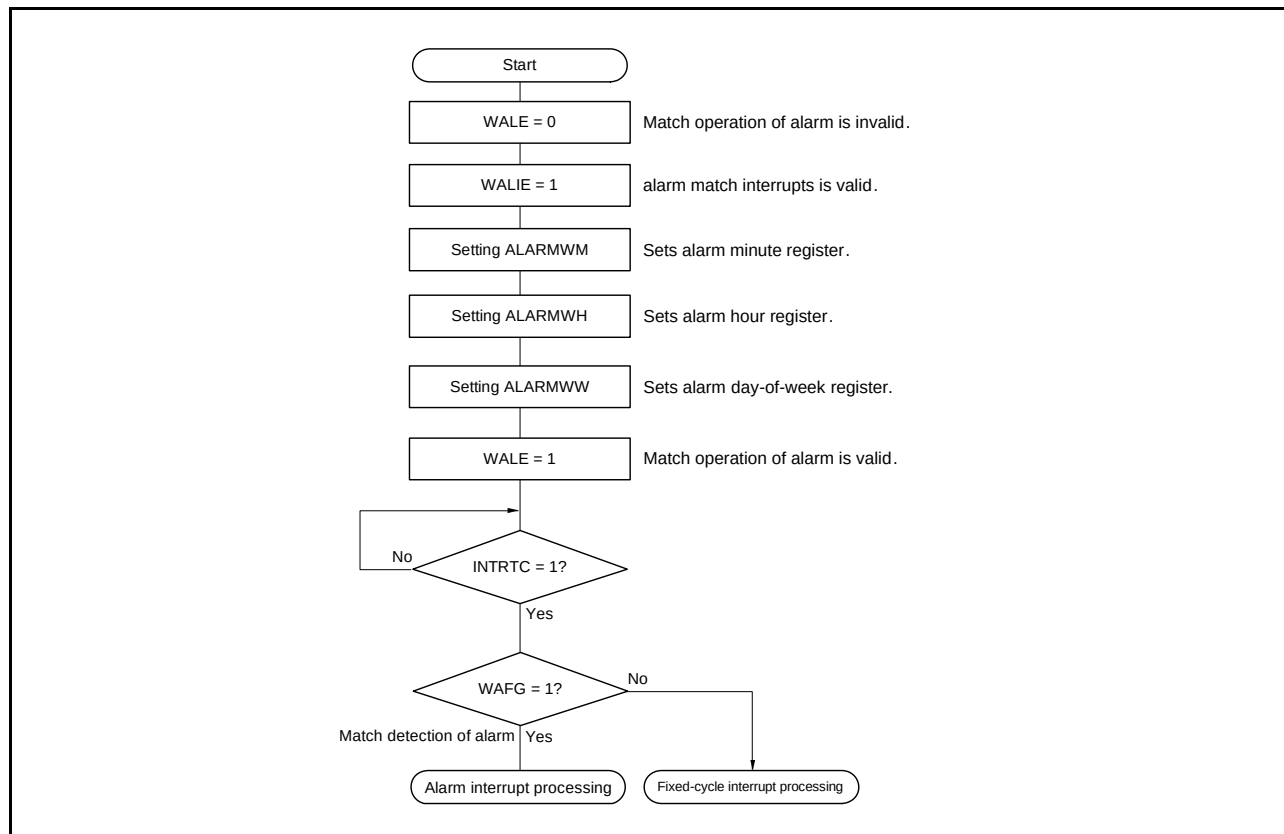
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counting is in progress (RTCE = 1), rewrite the values of the MIN register after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence. All the registers do not have to be set and only some registers may be written.

8.4.4 Setting alarm by the realtime clock

Set time of alarm after setting 0 to WALE (alarm operation invalid.) first.

Figure 8 - 23 Alarm Processing Procedure

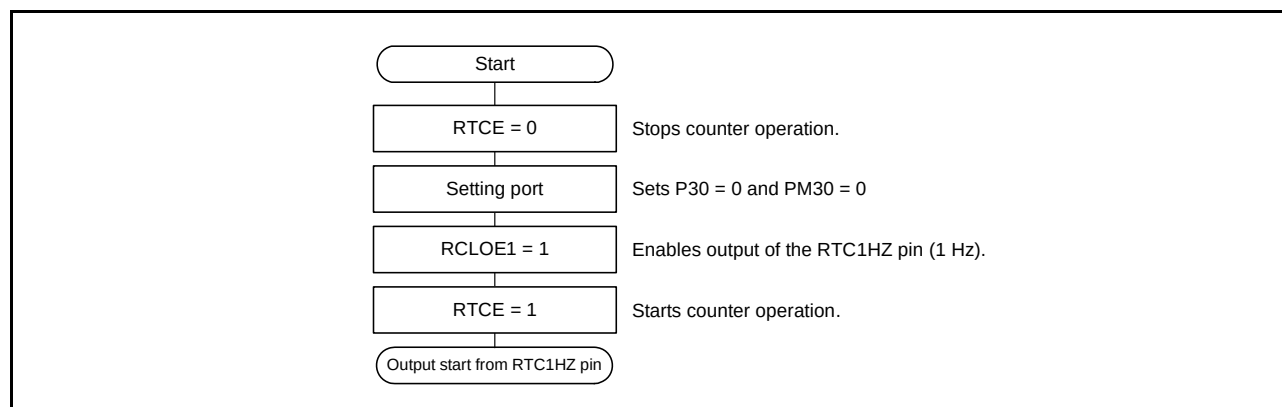


Remark 1. The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm day-of-week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). To use these two types of interrupts at the same time, the source of the interrupt can be identified by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) when an INTRTC is generated.

8.4.5 1 Hz output by the realtime clock

Figure 8 - 24 1 Hz Output Setting Procedure



Caution First set the `RTCWEN` bit to 1, while oscillation of the count clock (`fRTCCK`) is stable.

8.4.6 Example of time error correction by the realtime clock

Time can be corrected with high accuracy when it is slow or fast, by setting a value to the time error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16 bits) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value^{Note} = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1)
× 32768 × 60 ÷ 3

(When DEV = 1)

Correction value^{Note} = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1)
× 32768 × 60

Note The correction value is the time error correction value calculated by using bits 6 to 0 of the time error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), time error correction is not performed. “*” is 0 or 1.
/F5 to /F0 are bit-inverted values (000011 when 111100).

Remark 1. The correction value is 2, 4, 6, 8, ... 120, 122, 124 or –2, –4, –6, –8, ... –120, –122, –124.

Remark 2. The oscillation frequency is a value of the count clock (fRTCCK).

It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the time error correction register is set to its initial value (00H).

Remark 3. The target frequency is the frequency resulting after correction performed by using the time error correction register.

Correction example 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency^{Note} of each product, a signal at about 32.768 kHz can be output from the PCLBUZ0 pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See **8.4.5 1 Hz output by the realtime clock** for the setting procedure of the RTC1Hz output, and see **10.4 Operations of the Clock Output/Buzzer Output Controller** for the setting procedure for output of about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

When the output frequency from the PCLBUZ0 pin is 32772.3 Hz:

Given that the target frequency is 32768 Hz (32772.3 Hz -131.2 ppm) and the extent of correction is -131.2 ppm (not in the range below -63.1 ppm), set DEV to 0. Accordingly, the expression for calculating the correction value when DEV is 0 is applicable.

$$\begin{aligned}\text{Correction value} &= \text{Error for correction of counting of 1 minute} \div 3 \\ &= (\text{Oscillation frequency} \div \text{target frequency} - 1) \times 32768 \times 60 \div 3 \\ &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3 \\ &= 86\end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

When the correction value is 86:

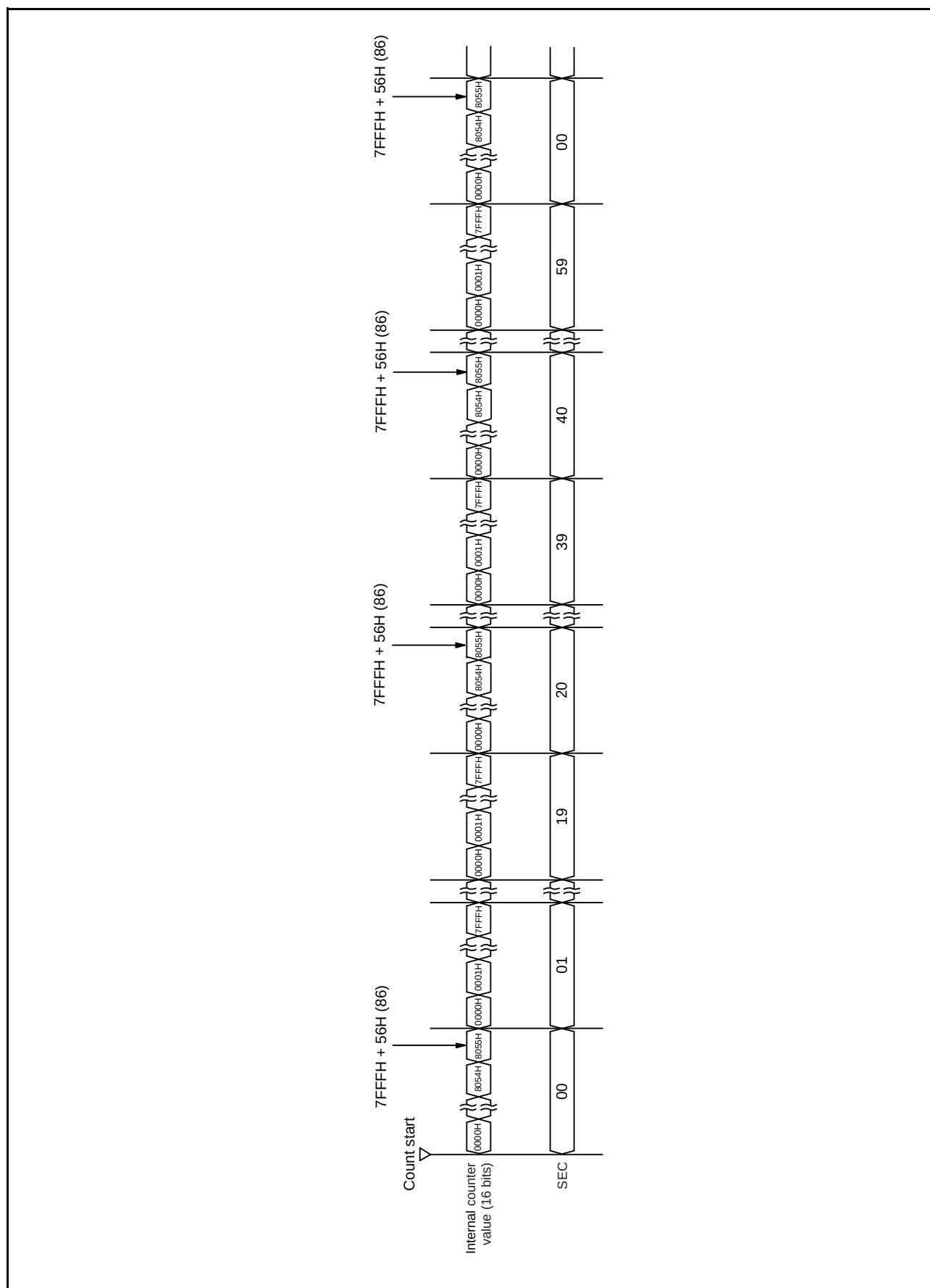
If the correction value is 0 or larger (the clock is running slow), set F6 to 0. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86 \\ (F5, F4, F3, F2, F1, F0) &= 44 \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 8 - 25 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 8 - 25 Operation for Correction when the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example 2

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency^{Note} of each product, a signal at about 1 Hz can be output from the RTC1HZ pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See 8.4.5 1 Hz output by the realtime clock for the setting procedure for output of about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

When the output frequency from the RTC1HZ pin is 0.9999817 Hz:

Oscillation frequency = $32768 \times 0.9999817 \approx 32767.4$ Hz

Given that the target frequency is 32768 Hz (32767.4 Hz + 18.3 ppm), set DEV to 1. Accordingly, the expression for calculating the correction value when DEV is 1 is applicable.

$$\begin{aligned} \text{Correction value} &= \text{Error for correction of counting of 1 minute} \\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \\ &= (32767.4 \div 32768 - 1) \times 32768 \times 60 \\ &= -36 \end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

When the correction value is -36:

If the correction value is 0 or less (the clock is running fast), set F6 to 1. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned} -\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2 &= -36 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1) \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0) \end{aligned}$$

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is -36 (bits 6 to 0 of the SUBCUD register: 1101110) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 8 - 26 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

The diagram illustrates the internal counter value (16 bits) and the SEC signal. The internal counter value is shown as a sequence of 16-bit values (0000H to 7FFFH) and the SEC signal is shown as a sequence of 16-bit values (00, 01, 19, 20, 39, 40, 59, 00). A 'Count start' signal is shown at the beginning. Two arrows point to the 7FFFH-24H (36) range.

Section 9 32-bit Interval Timer (TML32)

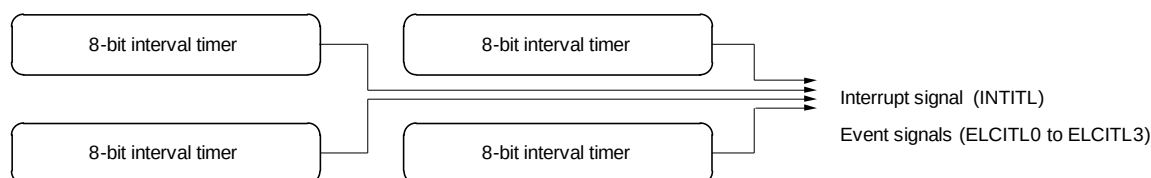
The 32-bit interval timer is made up of four 8-bit interval timers (referred to as channels 0 to 3). Each is capable of operating independently and in that case they all have the same functions. Two 8-bit interval timer channels can be connected to operate as a 16-bit interval timer. Four 8-bit interval timer channels can be connected to operate as a 32-bit interval timer.

9.1 Overview

The 32-bit interval timer operates with the f_{MXP} , f_{SXP} , f_{IHP} , or f_{IMP} clock or the event input from the ELC, which is asynchronous with the CPU operation. The 32-bit interval timer has the following functions.

(1) 8-bit counter mode

The 8-bit timers are usable as four 8-bit interval timers that generate interrupts (INTITL) and event signal (ELCITL n) at fixed intervals.



(2) 16-bit counter mode

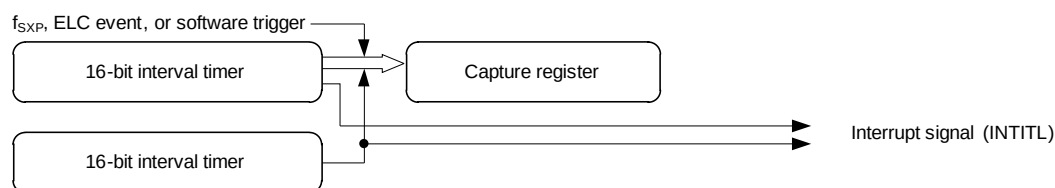
The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (INTITL) and event signal (ELCITL n) at fixed intervals.



(3) 16-bit capture mode

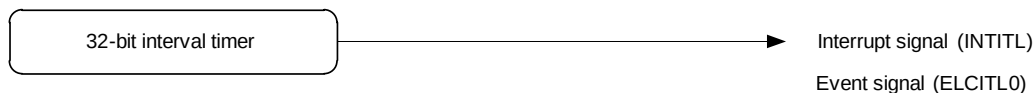
The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (INTITL) at fixed intervals.

The value of one of the 16-bit interval timers can also be stored in the capture register in response to a selected capture trigger.



(4) 32-bit counter mode

The 8-bit timers are usable as a 32-bit interval timer that generates interrupts (INTITL) and event signal (ELCITLn) at fixed intervals.



Remark n: Channel number (0 to 3)

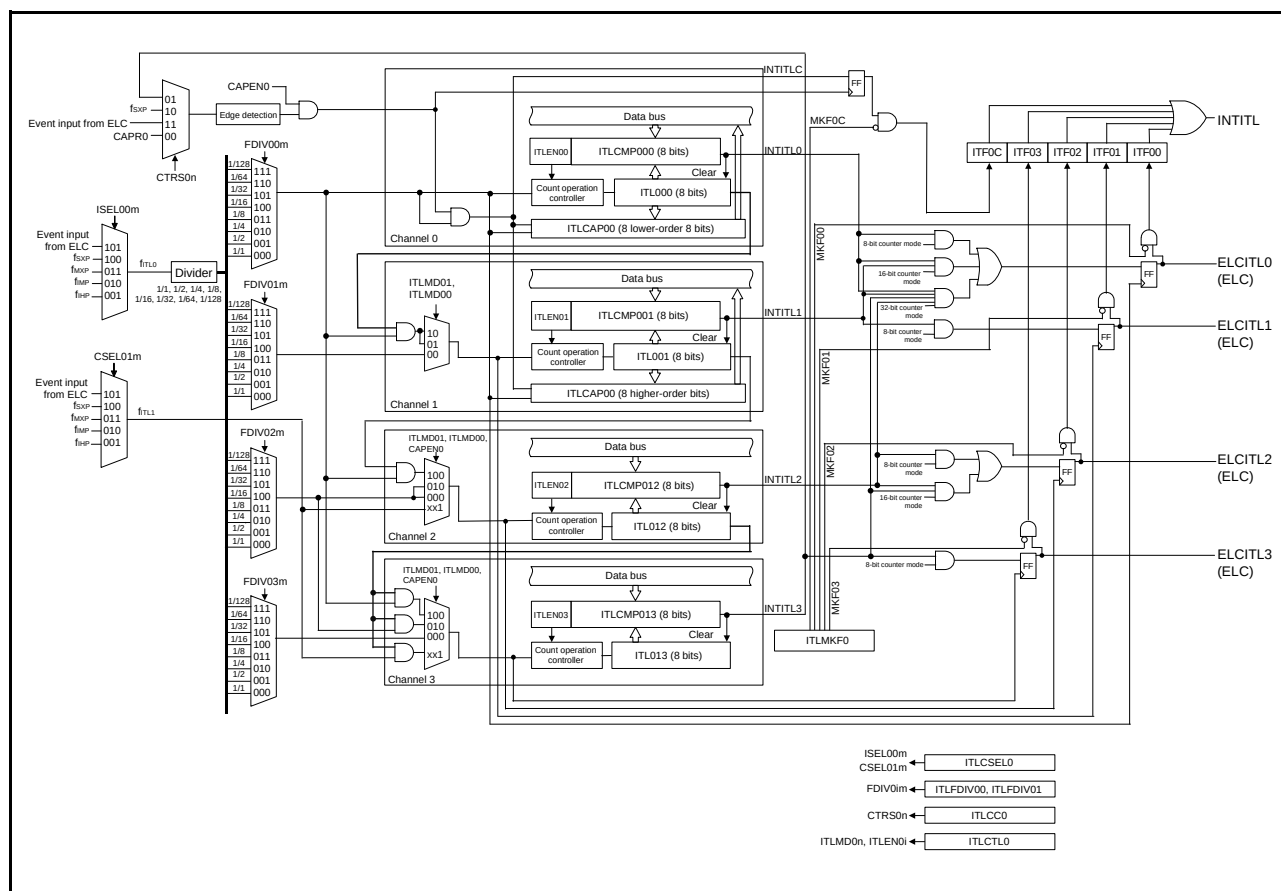
Table 9 - 1 lists the specifications of the 32-bit interval timer operations and **Figure 9 - 1** shows a block diagram of the 32-bit interval timer.

Table 9 - 1 Specifications of 32-bit Interval Timer Operations

Item	Description
Count source (operating clock)	• fMXP • fSXP • fIHP • fIMP • Event input from the ELC
Capture clock (Selectable sources for counting by the timer which can generate a capture trigger)	• fMXP • fSXP • fIHP • fIMP • Event input from the ELC
Frequency division ratio	• 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128
Operating mode	• 8-bit counter mode Channels 0 to 3 independently operate as 8-bit counters. • 16-bit counter mode The combinations of channels 0 and 1 and channels 2 and 3 are cascade-connectable to operate as two 16-bit counters. • 32-bit counter mode Channels 0 to 3 are connected to operate as a 32-bit counter. • 16-bit capture mode Channels 0 and 1 are connected to operate as a 16-bit counter using the count source, channels 2 and 3 are connected to operate as a 16-bit counter using the capture clock, and the connected counters are used for capture operation.
Interrupt	• Five interrupt sources are integrated into one interrupt signal and output as the INTITL signal. - Output when the counter value in any of channels 0 to 3 matches the compare value. - Output when the capturing of the counter value is completed in capture mode.
ELC	• Four trigger signals ELCITL0 to ELCITL3 for the ELC are output. - Output when the counter value in any of channels 0 to 3 matches the compare value.

Remark fMXP: High-speed peripheral clock frequency
 fSXP: Low-speed peripheral clock frequency
 fIHP: High-speed on-chip oscillator peripheral clock frequency
 fIMP: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
 Event input from the ELC: The ELC function is used to generate this signal from a peripheral module such as a timer, serial communications interface, or port pin. For details, see **Section 17 Event Link Controller (ELC)**.

Figure 9 - 1 Block Diagram of 32-bit Interval Timer



ITL000, ITL001, ITL012, ITL013: 8-bit counters

Note In 16-bit counter mode, the counters in channels 0 and 1 are connected (ITL000 + ITL001) and the counters in channels 2 and 3 are connected (ITL012 + ITL013).

In 32-bit counter mode, the counters in channels 0 to 3 are connected (ITL000 + ITL001 + ITL012 + ITL013).

ISEL00m, CSEL01m: Bits in the interval timer clock select register (ITLCSEL0)

FDIV0im: Bits in the interval timer frequency division register (ITLFDIV0n)

CTR0n: Bits in the interval timer capture control register (ITLCC0)

ITLMD0n, ITLEN0i: Bits in the interval timer control register (ITLCTL0)

Remark n = 0, 1 m = 0, 1, 2 i = 0, 1, 2, 3

9.2 Registers for Controlling the 32-bit Interval Timer

The following registers are used to control the 32-bit interval timer.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)
- Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1)**Note**
- Interval timer capture register 00 (ITLCAP00)**Note**
- Interval timer control register (ITLCTL0)
- Interval timer clock select register 0 (ITLCSEL0)
- Interval timer frequency division register 0 (ITLFDIV00)
- Interval timer frequency division register 1 (ITLFDIV01)
- Interval timer capture control register 0 (ITLCC0)
- Interval timer status register (ITLS0)
- Interval timer match detection mask register (ITLMKF0)

Note This register is only accessible in 16 bits.

9.2.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the 32-bit interval timer is to be used, be sure to set bit 4 (TML32EN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>
PER1	0	SMSSEN	0	TML32EN	DTCEN	UTAEN	0	CTSUEN

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. - The SFRs used by the 32-bit interval timer cannot be written. - When an SFR used by the 32-bit interval timer is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the 32-bit interval timer can be read and written.

Caution The functions that are mounted depend on the product. For details on the PER1 register, see the description in Section 6 Clock Generator.

9.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

Set bit 4 (TML32RES) of this register to 1 to reset the 32-bit interval timer.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH

After reset: 00H

R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	<0>
PRR1	0	SMSRES	0	TML32RES	0	0	0	CTSURES
Control resetting of the 32-bit interval timer								
0 The 32-bit interval timer is released from the reset state.								
1 The 32-bit interval timer is in the reset state. • The SFRs for use with the 32-bit interval timer are initialized.								

Caution Be sure to clear bits 7, 5, 3, 2, and 1.

9.2.3 Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)

These are compare value registers used in 8-bit counter mode.

These registers can be set by an 8-bit memory manipulation instruction.

The value of each ITLCMP0mn register is FFH following a reset. A value from 01H to FFH can be specified. Setting these registers to 00H is prohibited.

These registers hold values to be compared with the ITL000 to ITL013 counter values.

Figure 9 - 4 Format of Interval Timer Compare Registers 0mn (ITLCMP0mn)

Address: F0360H (ITLCMP000), F0361H (ITLCMP001), F0362H (ITLCMP012), F0363H (ITLCMP013)

After reset: FFH

R/W: R/W>Note

Symbol	7	6	5	4	3	2	1	0
ITLCMP0mn								

Note Write to ITLCMP000 while the ITLEN00 bit is 0.
Write to ITLCMP001 while the ITLEN01 bit is 0.
Write to ITLCMP012 while the ITLEN02 bit is 0.
Write to ITLCMP013 while the ITLEN03 bit is 0.

9.2.4 Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1)

These are compare value registers used in 16-bit or 32-bit counter mode.

These registers can be set by a 16-bit memory manipulation instruction.

The value of each ITLCMP0n register is FFFFH following a reset.

A value from 0001H to FFFFH can be specified. Setting these registers to 0000H is prohibited.

These registers hold values to be compared with the ITL0n counter values.

When the ITLMD01 and ITLMD00 bits are set to 1 and 0, respectively, these registers are used as compare registers in 32-bit counter mode; specify the upper 16-bit compare value in the ITLCMP01 register and the lower 16-bit compare value in the ITLCMP00 register.

Figure 9 - 5 Format of Interval Timer Compare Registers 0n (ITLCMP0n)

Address: F0360H, F0361H (ITLCMP00), F0362H, F0363H (ITLCMP01)

After reset: FFFFH

R/W: R/W>Note

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCMP0n																

Note Write to ITLCMP00 while the ITLEN00 bit is 0.
Write to ITLCMP01 while the ITLEN02 bit is 0 in 16-bit counter mode or while the ITLEN00 bit is 0 in 32-bit counter mode.

9.2.5 Interval timer capture register 00 (ITLCAP00)

This register holds 16-bit captured values when the interval timers are operating in 16-bit counter mode.
The values of the 16-bit counters (ITL000 + ITL001) are stored in the ITLCAP00 register in response to the capture trigger selected in the ITLCC0 register when the CAPEN0 bit in the ITLCC0 register is 1.
When an interrupt on compare match with the ITLCMP01 register is to be used, select the counter clock in the ITLCSEL0 register and set the comparison value in the TLCMP01 register.

Figure 9 - 6 Format of Interval Timer Capture Register 00 (ITLCAP00)

Address: F0364H, F0365H

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCAP00																

9.2.6 Interval timer control register (ITLCTL0)

This register is used to start or stop counting by the interval timer and to select 8-bit, 16-bit, or 32-bit counter mode.

This register can be set by a 1-bit or 8-bit manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0) (1/2)

Address: F0366H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ITLCTL0	ITLMD01	ITLMD00	0	0	ITLEN03	ITLEN02	ITLEN01	ITLEN00

ITLMD01	ITLMD00	Selection of 8-bit, 16-bit, or 32-bit counter mode ^{Note 1}
0	0	The interval timer operates in 8-bit counter mode.
0	1	The interval timer operates in 16-bit counter mode (channel 0 is connected with channel 1 and channel 2 is connected with channel 3).
1	0	The interval timer operates in 32-bit counter mode (channels 0 to 3 are connected).
1	1	Setting prohibited.

ITLEN03	8-bit counter mode: ITL013 count enable ^{Note 2}
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL013 counter and writing 0 stops it. In 16-bit counter mode, set this bit to 0. In 32-bit counter mode, set this bit to 0.	

ITLEN02	8-bit counter mode: ITL012 count enable ^{Note 2} 16-bit counter mode: ITL012 + ITL013 count enable ^{Note 2}
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 counter and writing 0 stops it. In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 + ITL013 counter and writing 0 stops it. In 32-bit counter mode, set this bit to 0.	

ITLEN01	8-bit counter mode: ITL001 count enable ^{Note 2}
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL001 counter and writing 0 stops it. In 16-bit counter mode, set this bit to 0. In 32-bit counter mode, set this bit to 0.	

Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0) (2/2)

ITLEN00	8-bit counter mode: ITL000 count enableNote 2 16-bit counter mode: ITL000 + ITL001 count enableNote 2 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enableNote 2
0	Counting stops.
1	Counting begins.
In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 counter and writing 0 stops it. In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 counter and writing 0 stops it. In 32-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 + ITL012 + ITL013 counter and writing 0 stops it.	

Note 1. To change the timer mode, be sure to write to the ITLMD00 and ITLMD01 bits only while the ITLEN00, ITLEN01, ITLEN02, and ITLEN03 bits are all 0.

Note 2. When one of the ITLEN03 to ITLEN00 bits is cleared to 0, the corresponding counter is cleared to 0 without synchronization with the counter clock.

Mode	ITLMD01	ITLMD00	ITLEN03	ITLEN02	ITLEN01	ITLEN00	Target Counter
8-bit mode	0	0				√	ITL000
					√		ITL001
				√			ITL012
			√				ITL013
16-bit mode	0	1	Always set to 0.		Always set to 0.	√	ITL000+ITL001
			Always set to 0.	√	Always set to 0.		ITL012+ITL013
32-bit mode	1	0	Always set to 0.	Always set to 0.	Always set to 0.	√	ITL000+ITL001+ITL012+ITL013

Note √: Enables counting in the target counter.

Note In 8-bit counter mode, two or more bits of ITLEN03 to ITLEN00 can be set to 1 or 0 at the same time.

Note In 16-bit counter mode, the ITLEN02 and ITLEN00 bits can be set to 1 or 0 at the same time.

9.2.7 Interval timer clock select register 0 (ITLCSEL0)

This register is used to select the count source for the interval timer.

This register can be set by an 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 8 Format of Interval Timer Clock Select Register 0 (ITLCSEL0)

Address: F0367H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLCSEL0	0	CSEL012	CSEL011	CSEL010	0	ISEL002	ISEL001	ISEL000

CSEL012	CSEL011	CSEL010	Selection of interval timer count clock for capturing (fiTL1) ^{Note}
0	0	0	Counting stops.
0	0	1	fiHP
0	1	0	fIMP
0	1	1	fMXP
1	0	0	fsXP
1	0	1	Event input from the ELC
Others			Setting prohibited.

ISEL002	ISEL001	ISEL000	Selection of interval timer count clock (fiTL0) ^{Note}
0	0	0	Counting stops.
0	0	1	fiHP
0	1	0	fIMP
0	1	1	fMXP
1	0	0	fsXP
1	0	1	Event input from the ELC
Others			Setting prohibited.

Note Be sure to write to the CSEL012 to CSEL010 bits and ISEL002 to ISEL000 bits only while the ITLEN03 to ITLEN00 bits are all 0.

Remark fMXP: High-speed peripheral clock frequency
fsXP: Low-speed peripheral clock frequency
fiHP: High-speed on-chip oscillator peripheral clock frequency
fIMP: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
Event input from the ELC: The ELC function is used to generate this signal from a peripheral module such as a timer, serial communications interface, or port pin. For details, see **Section 17 Event Link Controller (ELC)**.

9.2.8 Interval timer frequency division register 0 (ITLFDIV00)

This register is used to select the counter clock for the interval timer.

This register can be set by an 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00) (1/2)

Address: F0368H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV00	0	FDIV012	FDIV011	FDIV010	0	FDIV002	FDIV001	FDIV000

FDIV012	FDIV011	FDIV010	8-bit counter mode: Counter clock for ITL001 ^{Note 1}
0	0	0	fITL0
0	0	1	fITL0/2
0	1	0	fITL0/4
0	1	1	fITL0/8
1	0	0	fITL0/16
1	0	1	fITL0/32
1	1	0	fITL0/64
1	1	1	fITL0/128

In 8-bit counter mode, ITL001 counts cycles of the counter clock specified in the FDIV010 to FDIV012 bits.
 In 16-bit counter mode, set these bits to 000B.
 In 32-bit counter mode, set these bits to 000B.

Figure 9 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00) (2/2)

FDIV002	FDIV001	FDIV000	8-bit counter mode: Counter clock for ITL000 ^{Note 2} 16-bit counter mode: Counter clock for ITL000 + ITL001 ^{Note 2} 32-bit counter mode: Counter clock for ITL000 + ITL001 + ITL012 + ITL013 ^{Note 2}
0	0	0	fITL0
0	0	1	fITL0/2
0	1	0	fITL0/4
0	1	1	fITL0/8
1	0	0	fITL0/16
1	0	1	fITL0/32
1	1	0	fITL0/64
1	1	1	fITL0/128

In 8-bit counter mode, ITL000 counts cycles of the counter clock specified in the FDIV000 to FDIV002 bits.
 In 16-bit counter mode, ITL000 + ITL001 counts cycles of the counter clock specified in the FDIV000 to FDIV002 bits.
 In 32-bit counter mode, ITL000 + ITL001 + ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV000 to FDIV002 bits.

Note 1. In 8-bit counter mode, be sure to write to the FDIV012 to FDIV010 bits only while the ITLEN01 bit is 0.

Note 2. Be sure to write to the FDIV002 to FDIV000 bits only while the ITLEN00 bit is 0.

9.2.9 Interval timer frequency division register 1 (ITLFDIV01)

This register is used to select the counter clock for the interval timer.

This register can be set by an 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 10 Format of Interval Timer Frequency Division Register 1 (ITLFDIV01)

Address: F0369H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV01	0	FDIV032	FDIV031	FDIV030	0	FDIV022	FDIV021	FDIV020

FDIV032	FDIV031	FDIV030	8-bit counter mode: Counter clock for ITL013 ^{Note 1}
0	0	0	fITL0
0	0	1	fITL0/2
0	1	0	fITL0/4
0	1	1	fITL0/8
1	0	0	fITL0/16
1	0	1	fITL0/32
1	1	0	fITL0/64
1	1	1	fITL0/128
In 8-bit counter mode, ITL013 counts cycles of the counter clock specified in the FDIV030 to FDIV032 bits. In 16-bit counter mode, set these bits to 000B. In 32-bit counter mode, set these bits to 000B.			

FDIV022	FDIV021	FDIV020	8-bit counter mode: Counter clock for ITL012 ^{Note 2} 16-bit counter mode: Counter clock for ITL012 and ITL013 ^{Note 2}
0	0	0	fITL0
0	0	1	fITL0/2
0	1	0	fITL0/4
0	1	1	fITL0/8
1	0	0	fITL0/16
1	0	1	fITL0/32
1	1	0	fITL0/64
1	1	1	fITL0/128
In 8-bit counter mode, ITL012 counts cycles of the counter clock specified in the FDIV020 to FDIV022 bits. In 16-bit counter mode, ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV020 to FDIV022 bits. In 32-bit counter mode, these bits are not used; write 000B to them.			

Note 1. In 8-bit counter mode, be sure to write to the FDIV032 to FDIV030 bits only while the ITLEN03 bit is 0.

Note 2. In 8-bit or 16-bit counter mode, be sure to write to the FDIV022 to FDIV020 bits only while the ITLEN02 bit is 0.

9.2.10 Interval timer capture control register 0 (ITLCC0)

This register is used to enable or disable the capture function of the interval timer, specify whether to hold or clear the capture completion flag, set up the software trigger, and select the capture trigger.

This register can be set by a 1-bit or 8-bit manipulation instruction.

The value of this register is 00H following a reset.

Figure 9 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0)

Address: F036AH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	0
ITLCC0	CAPEN0	CAPF0CR	CAPF0	CAPR0	CAPC0CR	0	CTRS01	CTRS00
	CAPEN0	Capture enable ^{Note 1}						
	0	Capturing is disabled.						
	1	Capturing is enabled.						
	CAPF0CR	Capture completion flag clear ^{Note 2}						
	0	The value of the capture completion flag CAPF0 is held.						
	1	The value of the capture completion flag CAPF0 is cleared.						
	CAPF0	Capture completion flag ^{Note 3}						
	0	Capturing has not been completed.						
	1	Capturing has been completed. This flag is set to 1 after a capture trigger selected in the CTRS01 and CTRS00 bits is generated and the captured data is stored in ITLCAP00. Writing 1 to the CAPF0CR bit clears this flag to 0.						
	CAPR0	Software capture trigger ^{Notes 4, 7}						
	0	Trigger operation does not proceed.						
	1	A software trigger for capturing is generated.						
	CAPC0CR	Selection of capture counter clearing after capturing ^{Note 5}						
	0	The capture counter value is held after the completion of capturing.						
	1	The capture counter value is cleared after the completion of capturing.						
	CTRS01	CTRS00	Selection of capture trigger ^{Notes 6, 7}					
	0	0	Software trigger					
	0	1	Interrupt on compare match with ITCMP01					
	1	0	fSXP (rising edge)					
	1	1	Event input from ELC (rising edge)					

(Notes are listed on the next page.)

- Note 1.** Be sure to write to the CAPEN0 bit only while the ITLEN03 to ITLEN00 bits are all 0.
- Note 2.** The CAPF0CR bit is always read as 0.
- Note 3.** Bit 5 is read-only.
- Note 4.** The CAPR0 bit is always read as 0.
- Note 5.** Be sure to write to the CAPC0CR bit only while the ITLEN03 to ITLEN00 bits are all 0.
- Note 6.** Be sure to write to the CTRS01 and CTRS00 bits only while the ITLEN03 to ITLEN00 bits are all 0.
- Note 7.** In the capture operation, the interval at which the capture trigger is generated should be two or more cycles of the counter clock.

9.2.11 Interval timer status register (ITLS0)

This is a status register for the interval timer.

This register can be set by an 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

When the value of the ITL0mn counter (mn = 00, 01, 12, 13) matches the value specified in the ITLCMP0mn, ITLCMP00, and ITLCMP01 registers, the compare-match flag for the corresponding channel is set.

When a capture trigger is generated while the CAPEN0 bit in the ITLCC0 register is 1, the capture detection flag is set after the value of the ITL0n counter is stored in the ITLCAP00 register.

The values of the ITF0C and ITF03 to ITF00 bits in this register are ORed and output as the INTITL interrupt signal.

Table 9 - 2 shows the conditions for setting the status flags in each timer mode selected by the ITLMD01 and ITLMD00 bits.

Figure 9 - 12 Format of Interval Timer Status Register (ITLS0)

Address: F036BH
After reset: 00H
R/W: R/W^{Note}

Symbol	7	6	5	4	3	2	1	0
ITLS0	0	0	0	ITF0C	ITF03	ITF02	ITF01	ITF00
ITF0C								
Capture detection flag								
0								
Completion of capturing has not been detected.								
1								
Completion of capturing has been detected.								
ITF03								
Compare match detection flag for channel 3								
0								
A compare match signal has not been detected in channel 3.								
1								
A compare match signal has been detected in channel 3.								
ITF02								
Compare match detection flag for channel 2								
0								
A compare match signal has not been detected in channel 2.								
1								
A compare match signal has been detected in channel 2.								
ITF01								
Compare match detection flag for channel 1								
0								
A compare match signal has not been detected in channel 1.								
1								
A compare match signal has been detected in channel 1.								
ITF00								
Compare match detection flag for channel 0								
0								
A compare match signal has not been detected in channel 0.								
1								
A compare match signal has been detected in channel 0.								

Note Writing 1 to each bit is ignored. To clear the ITF0C or ITF0i bit (i = 0, 1, 2, 3), write 0 to the desired bit and 1 to the other bits by using an 8-bit memory manipulation instruction.

Caution 1. If clearing any of the ITF0C, ITF03, ITF02, ITF01, ITF00 flag bits to 0 does not lead to the value of the ITLS0 register becoming 00H, an interrupt request (INTITL) is generated and the interrupt request flag (ITLIF) is set to 1.

Caution 2. To clear a flag bit in the ITLS0 register to 0, only write 0 to a bit that has the setting 1. This is because writing 0 to a bit that has the setting 0 may make detecting a compare match signal or capture detection signal generated at the same time as the writing of 0 impossible. For example, when the ITF01 flag bit is set to 1, write 00011101B to the ITLS0 register to clear the ITF01 flag bit.

Table 9 - 2 Conditions for Setting the Status Flags in Each Timer Mode

Mode	ITLMD01	ITLMD00	CAPEN0	Status Flag	Conditions for Setting Status Flag
8-bit mode	0	0	x	ITF00	The next rising edge of the counter clock following a match between the ITLCMP000 and ITL000 values
			x	ITF01	The next rising edge of the counter clock following a match between the ITLCMP001 and ITL001 values
			x	ITF02	The next rising edge of the counter clock following a match between the ITLCMP012 and ITL012 values
			x	ITF03	The next rising edge of the counter clock following a match between the ITLCMP013 and ITL013 values
16-bit mode	0	1	x	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 and ITL000 + ITL001 values
			x	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01 and ITL012 + ITL013 values
			1	ITF0C	The ITL000 + ITL001 value is stored in ITLCAP00 after a capture trigger is generated.
32-bit mode	1	0	—	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 + ITLCMP01 and ITL000 + ITL001 + ITL012 + ITL013 values

9.2.12 Interval timer match detection mask register (ITLMKF0)

This register is used to enable or disable setting of each effective bit in the interval timer status register (ITLS0) to 1.

This register can be set by an 8-bit memory manipulation instruction.

The value of this register is 00H following a reset.

Setting an MKF0C or MKF0i (i = 0 to 3) bit to 1 masks the corresponding status flag among ITF0C and ITF0i (i = 0 to 3), after which the given flag is not set to 1 even if a compare match with a compare register or capture completion is detected. Since the status flag will not be set to 1, masking also prevents generation of the interval detection interrupt (INTITL).

Figure 9 - 13 Format of Interval Timer Match Detection Mask Register (ITLMKF0)

Address: F036CH

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLMKF0	0	0	0	MKF0C	MKF03	MKF02	MKF01	MKF00
MKF0C								
Mask for capture detection status flag ^{Note}								
0								
ITF0C is not masked.								
1								
ITF0C is masked.								
MKF03								
Mask for compare match status flag for channel 3 ^{Note}								
0								
ITF03 is not masked.								
1								
ITF03 is masked.								
MKF02								
Mask for compare match status flag for channel 2 ^{Note}								
0								
ITF02 is not masked.								
1								
ITF02 is masked.								
MKF01								
Mask for compare match status flag for channel 1 ^{Note}								
0								
ITF01 is not masked.								
1								
ITF01 is masked.								
MKF00								
Mask for compare match status flag for channel 0 ^{Note}								
0								
ITF00 is not masked.								
1								
ITF00 is masked.								

Note Setting all functional bits to 1 for masking prevents setting of the corresponding bits in the ITLS0 register. This in turn prevents software detection of compare matches and completion of capture. When compare match for any of channels 0 to 3 is to be used, be sure to set the bit corresponding to the given status flag to 0 so that the flag is not masked. For the state of completion of capture, on the other hand, the CAPF0 flag in the interval timer capture control register 0 (ITLCC0) can be used to detect this even when the MKF0C bit is set to 1 to mask the ITF0C flag.

9.3 Operation

9.3.1 Counter mode settings

The 32-bit interval timer has three counter modes: 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode.

Table 9 - 3 to **Table 9 - 5** show the registers and settings for use in 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode, respectively.

Table 9 - 3 Registers and Settings Used in 8-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare registers 0mn (ITLCMP0mn)	Bits 7 to 0	Specify 8-bit compare values for channels 0 to 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channel 0.
	ITLEN01	Specify whether to start or stop counting in channel 1.
	ITLEN02	Specify whether to start or stop counting in channel 2.
	ITLEN03	Specify whether to start or stop counting in channel 3.
	ITLMD00	Set to 0.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV000 to FDIV002	Select the count clock for channel 0.
	FDIV010 to FDIV012	Select the count clock for channel 1.
	FDIV020 to FDIV022	Select the count clock for channel 2.
	FDIV030 to FDIV032	Select the count clock for channel 3.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the count clock for the interval timer.
	CSEL010 to CSEL012	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark mn = 00, 01, 12, 13

Table 9 - 4 Registers and Settings Used in 16-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1, and channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV000 to FDIV002	Select the count clock for channels 0 and 1.
	FDIV010 to FDIV012	Set to 000B.
	FDIV020 to FDIV022	Select the count clock for channels 2 and 3.
	FDIV030 to FDIV032	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the count clock for the interval timer.
	CSEL010 to CSEL012	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

Table 9 - 5 Registers and Settings Used in 32-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify a compare value in 32-bit counter mode. Specify the lower 16 bits of the compare value in channels 0 and 1 (ITLCMP00) and the upper 16 bits of the compare value in channels 2 and 3 (ITLCMP01).
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 to 3.
	ITLEN01	Set to 0.
	ITLEN02	Set to 0.
	ITLEN03	Set to 0.
	ITLMD00	Set to 0.
	ITLMD01	Set to 1.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV000 to FDIV002	Select the count clock for channels 0 to 3.
	FDIV010 to FDIV012	Set to 000B.
	FDIV020 to FDIV022	Set to 000B.
	FDIV030 to FDIV032	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the count clock for the interval timer.
	CSEL010 to CSEL012	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

9.3.2 Capture mode settings

When the 16-bit capture mode is to be used for channels 0 and 1, the counter value is stored in interval timer capture register 00 (ITLCAP00) in response to a selected capture trigger.

Table 9 - 6 Registers and Settings Used in 16-bit Capture Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1.
Interval timer compare register 01 (ITLCMP01) ^{Note}	Bits 15 to 0	Specify 16-bit compare values for channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV000 to FDIV002	Select the count clock for channel 0.
	FDIV010 to FDIV012	Set to 000B.
	FDIV020 to FDIV022	Set to 000B.
	FDIV030 to FDIV032	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the count clock for the interval timer in channels 0 and 1.
	CSEL010 to CSEL012	Select the count clock for the interval timer for capturing in channels 2 and 3.
Interval timer capture control register 0 (ITLCC0)	CAPEN0	Set to 1.
	CAPC0CR	Specify whether to clear or hold the counter value in channels 0 and 1 after the completion of capturing.
	CTRS00 and CTRS01	Select a capture trigger.

Note Channels 2 and 3 can only be used in 16-bit counter mode when an interrupt on compare match with ITLCMP01 is not to be used as a capture trigger.

Remark n = 0, 1

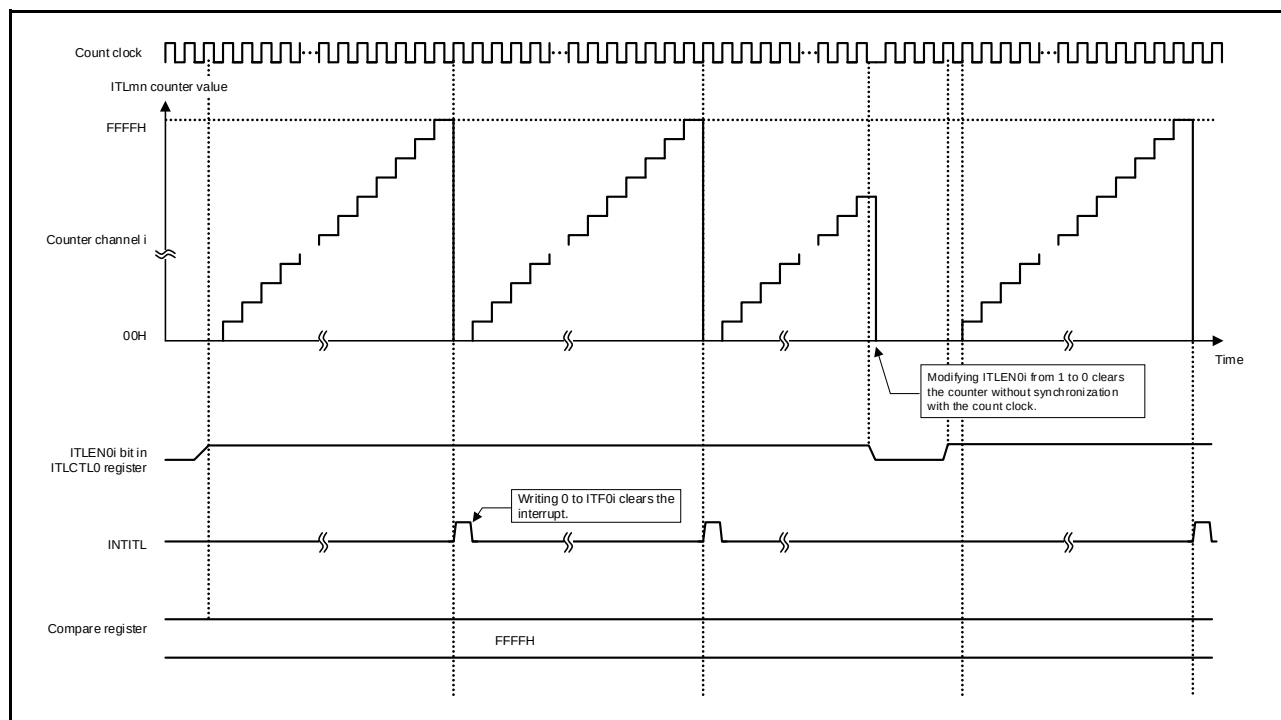
9.3.3 Timer operation

The ITL0mn counter counts up cycles of the counting clock specified in the interval timer frequency division registers (ITLFDIV00 and ITLFDIV01). An interrupt request signal (INTITL) is generated on the counting of the next clock cycle after the value of the counter matches the comparison value. The interrupt request signal (INTITL) remains high until the value of the ITLS0 register becomes 00H.

While the interrupt request signal (INTITL) is high, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) proceeds even if a compare match or capture completion is detected for an operating channel.

Clearing the ITLEN00 to ITLEN03 bits to 0 clears the counter value.

Figure 9 - 14 Example of Timer Operation



Remark mn = 00, 01, 12, 13 i = 0, 1, 2, 3

9.3.4 Capture operation

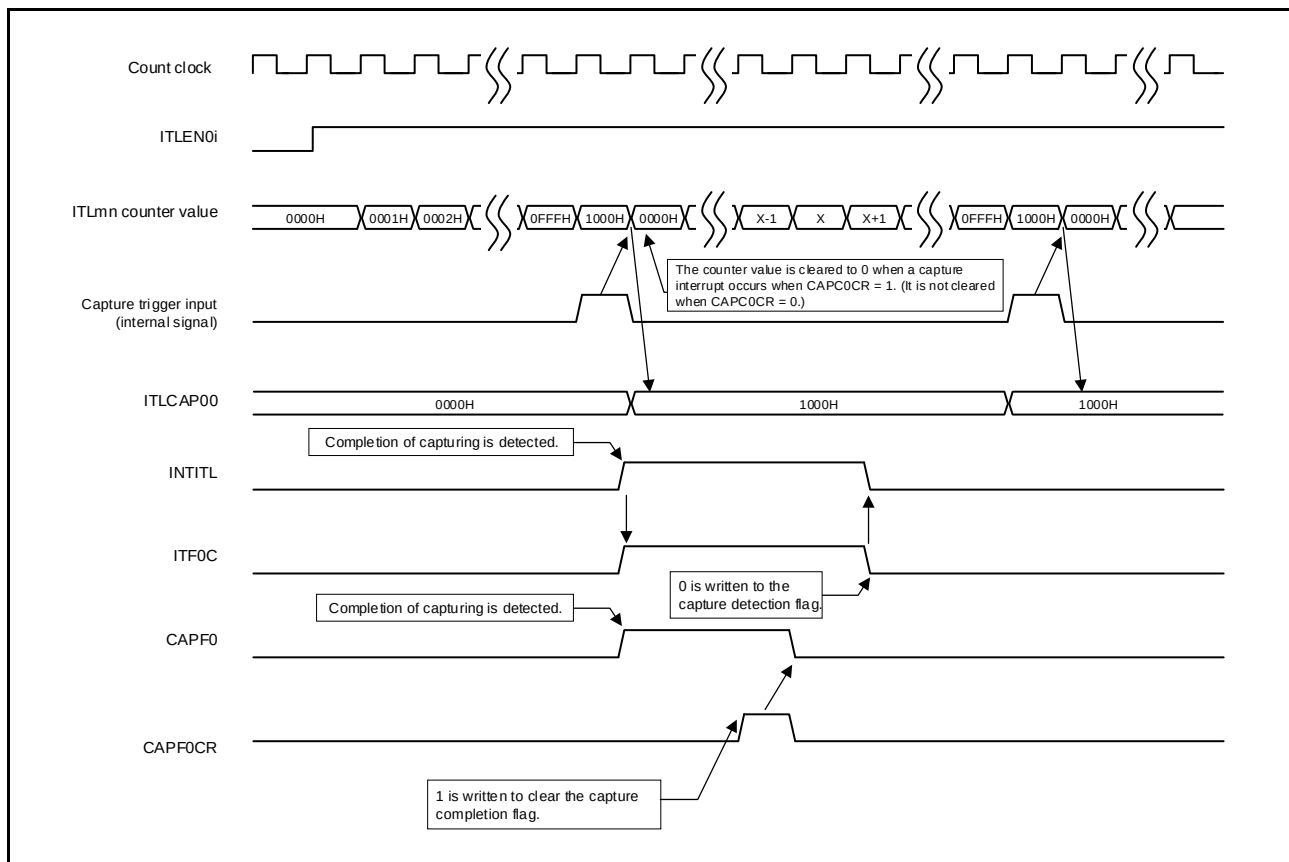
When the setting of the CAPEN0 bit in the interval timer capture control register 0 (ITLCC0) is 1, the values in the 16-bit counters (ITL000 and ITL001) are stored in interval timer capture register 00 (ITLCAP00) in response to the capture trigger specified in the ITLCC0 register.

The capture trigger is selectable from among the interrupt on compare match with the ITLCMP01 register, fsXP, an event signal input from the ELC, and a software trigger (setting the CAPR0 bit to 1). To use the interrupt on compare match with the ITLCMP01 register as the capture trigger, set interval timer clock select register 0 (ITLCSEL0) to select the clock for counting, and set interval timer compare register 01 (ITLCMP01) to specify the comparison value. When using fsXP, an event signal input from the ELC, or a software trigger (setting the CAPR0 bit to 1) as a capture trigger, channels 2 and 3 can be used in 16-bit counter mode.

After a capture trigger is input and the counter value is stored in the interval timer capture register, the interrupt request signal (INTITL) is output, the capture completion flag (CAPF0) and capture detection flag (ITF0C) are set to 1, and the flag values are retained until they are explicitly cleared **Note**. The CAPF0 flag can be cleared by setting the CAPF0CR bit to 1. The ITF0C flag in the ITLS0 register can be cleared by writing 0 to it. Since capture operations operate with the counter clock, the interval at which the capture trigger is generated should be at least five cycles of the counter clock. If a capture trigger is generated again within two cycles of the counter clock after an earlier capture trigger was generated, the CAPF0 bit may not be set.

Note If the value of the ITLS0 register is other than 00H, neither interrupt operation nor setting of the interrupt request flag (ITLIF) will proceed even when the capture detection flag (ITF0C) is set to 1 because the interrupt request signal (INTITL) is kept at the high level.

Figure 9 - 15 Example of Capture Operation



Remark mn = 00, 01, 12, 13 i = 0, 1, 2, 3

When the counter value matches the comparison value while the CAPC0CR bit in the ITLCC0 register is set to 1 (mode where the capture counter value is cleared following the completion of capture), counting of the next clock cycle clears the counter value. The counter value is not cleared in this way if the CAPC0CR bit is set to 0 (mode where the capture counter retains its value after the completion of capture).

9.3.5 Interrupt

Table 9 - 7 shows the interrupt sources in 8-bit, 16-bit, and 32-bit counter modes.

The ITF00 to ITF03 and ITF0C bits are interrupt status flags in the ITLS0 register. When any of the interrupt status flag is set, an interrupt request is output as the INTITL signal.

Table 9 - 7 Interrupt Sources in 8-bit, 16-bit, and 32-bit Counter Modes

Interrupt Source	Interrupt Condition in 8-bit Counter Mode	Interrupt Condition in 16-bit Counter Mode	Interrupt Condition in 32-bit Counter Mode
ITF00	Next rising edge of the counter clock after a compare match in channel 0	Next rising edge of the counter clock after a compare match in channels 0 and 1	Next rising edge of the counter clock after a compare match
ITF01	Next rising edge of the counter clock after a compare match in channel 1	Not generated	Not generated
ITF02	Next rising edge of the counter clock after a compare match in channel 2	Next rising edge of the counter clock after a compare match in channels 2 and 3	Not generated
ITF03	Next rising edge of the counter clock after a compare match in channel 3	Not generated	Not generated
ITF0C	Not generated; this is the case when the setting of ITLCC0 is 00H.	Timing of storing the counter value in the capture register after a capture trigger is input	Not generated; this is the case when the setting of ITLCC0 is 00H.

If the value of the ITLS0 register is other than 00H, the interrupt request signal (INTITL) is kept at the high level. Accordingly, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel. However, if the value of the ITLS0 register is not 00H after any bit in the ITLS0 register is set to 0 by an 8-bit memory manipulation instruction, a low-level pulse signal is output on the INTITL pin and the interrupt request flag (ITLIF) is set to 1. Accordingly, clearing a status flag in the ITLS0 register to 0 while ITLIF = 0 during vector interrupt processing or other processing enables the detection of an interrupt in response to another status bit having the setting 1. **Figure 9 - 16** shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

The following describes the operation shown in **Figure 9 - 16**.

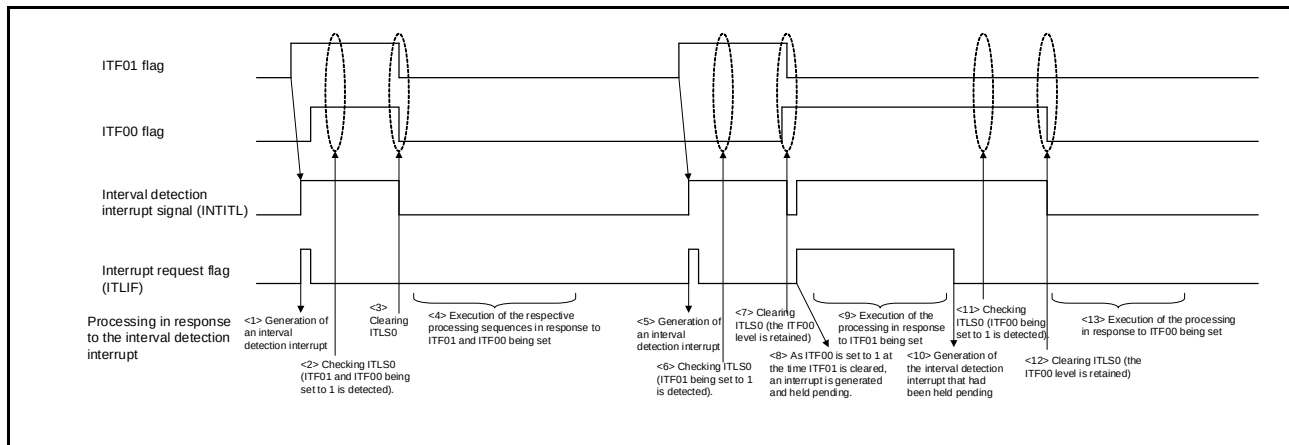
When a compare match in channel 1 is detected while the value of the ITLS0 register is 00H, the ITF01 flag is set to 1 and the interval detection interrupt signal (INTITL) is driven high. While the interval detection interrupt signal (INTITL) is kept at the high level, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel. Note that if another detection flag is set to 1 immediately before clearing the ITF0x (x = 0, 1, 2, 3, C) flag bit to 0, the output of the INTITL pin temporarily goes to the low level after clearing of the given ITF0x flag bit, leading to setting of the interrupt request flag (ITLIF) to 1.

- <1> The ITF01 flag is set to 1 in response to a compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.
- <2> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF01 and ITF00 flags being set to 1 can be confirmed.
- <3> Clear the ITF01 and ITF00 flags detected in step 2 by using an 8-bit memory manipulation instruction to write 00011100B to the ITLS0 register so that its value becomes 00H.**Note**
- <4> The respective processing sequences in response to the ITF01 and ITF00 flags being set to 1 are then executed.**Note**

Note Missing an interrupt source can also be prevented by repeating the processing for clearing an interrupt source per flag.

- <5> The ITF01 flag is set to 1 in response to a further compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.
- <6> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF01 flag being set to 1 can be confirmed.
- <7> Clear the ITF01 flag detected in step 6 by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H. Though the ITF00 flag is also set to 1 in response to the compare match in channel 0 at this time, the ITF00 flag is not cleared because the processing for the flag does not proceed.
- <8> As the ITF00 flag is set to 1 at the time the ITF01 flag is cleared to 0 in step 7, the INTITL signal is temporarily driven low and the interrupt request flag (ITLIF) is set to 1. If the interrupt enable flag (IE) is not cleared to 0 at this time, this interrupt request is held pending.
- <9> The processing in response to the ITF01 flag being set to 1 is then executed.
- <10> As the interrupt request flag (ITLIF) is still set to 1 after return from the processing in response to the ITF01 flag being set to 1, clear the interrupt request flag (ITLIF) to 0, after which the interval detection interrupt processing that was held pending proceeds.
- <11> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF00 flag being set to 1 can be confirmed.
- <12> Clear the ITF00 flag detected in step 11 by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H.
- <13> The processing in response to the ITF00 flag being set to 1 is then executed.

Figure 9 - 16 Example of Clearing the Detected Flags



9.3.6 Interval timer setting procedures

The following shows the procedures for setting up the 32-bit interval timer.

Figure 9 - 17 Procedure for Starting the 32-bit Interval Timer

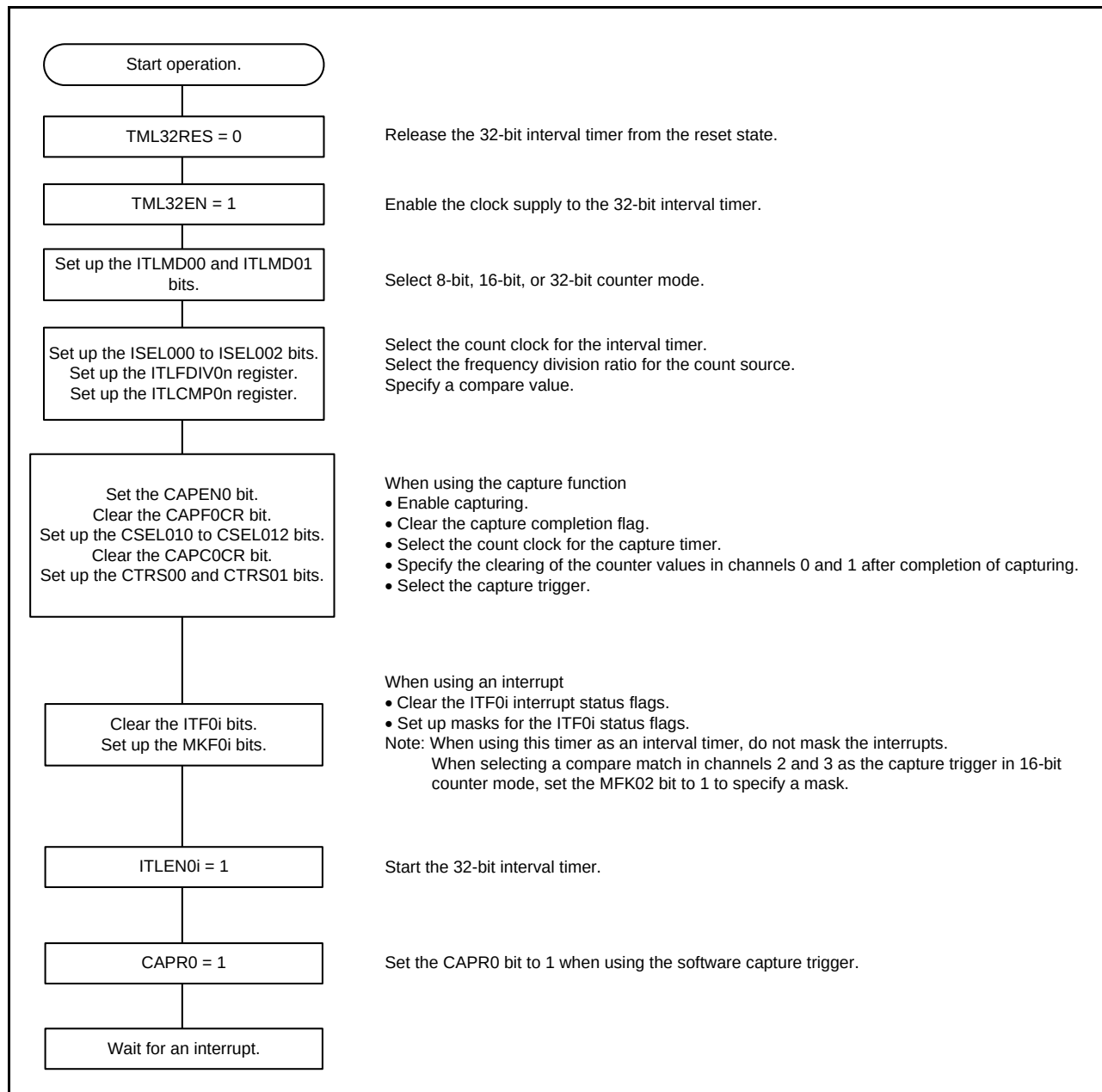


Figure 9 - 18 Procedure for Stopping the 32-bit Interval Timer

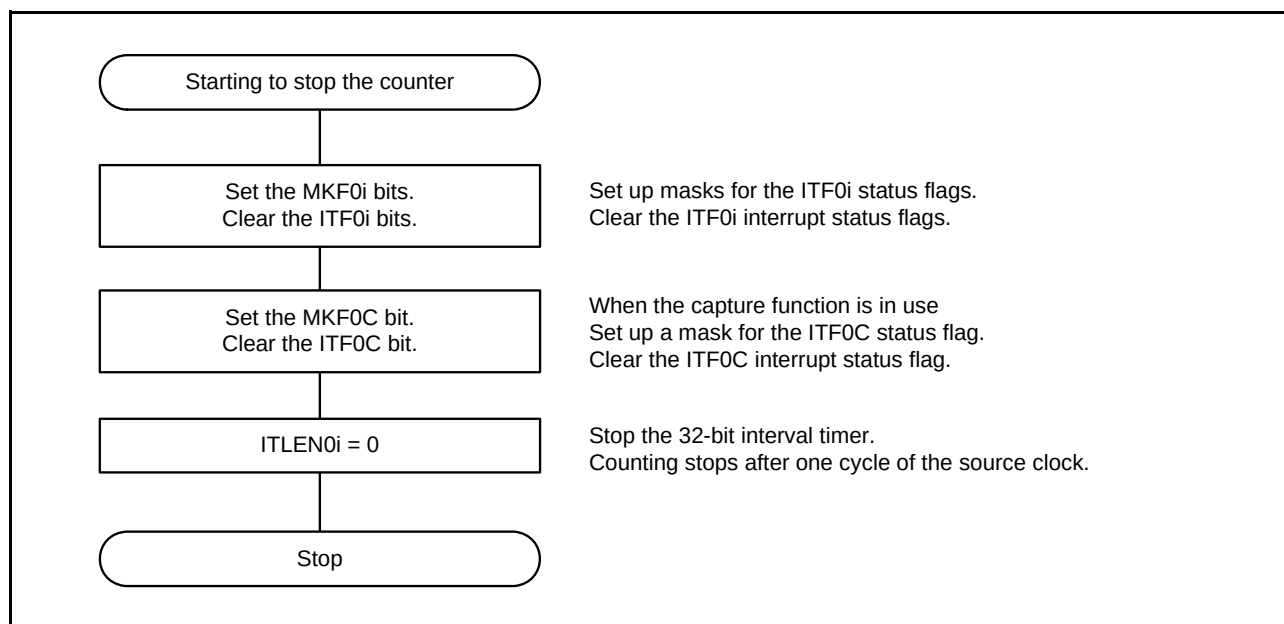


Figure 9 - 19 Procedure for Changing the Operating Mode of the 32-bit Interval Timer

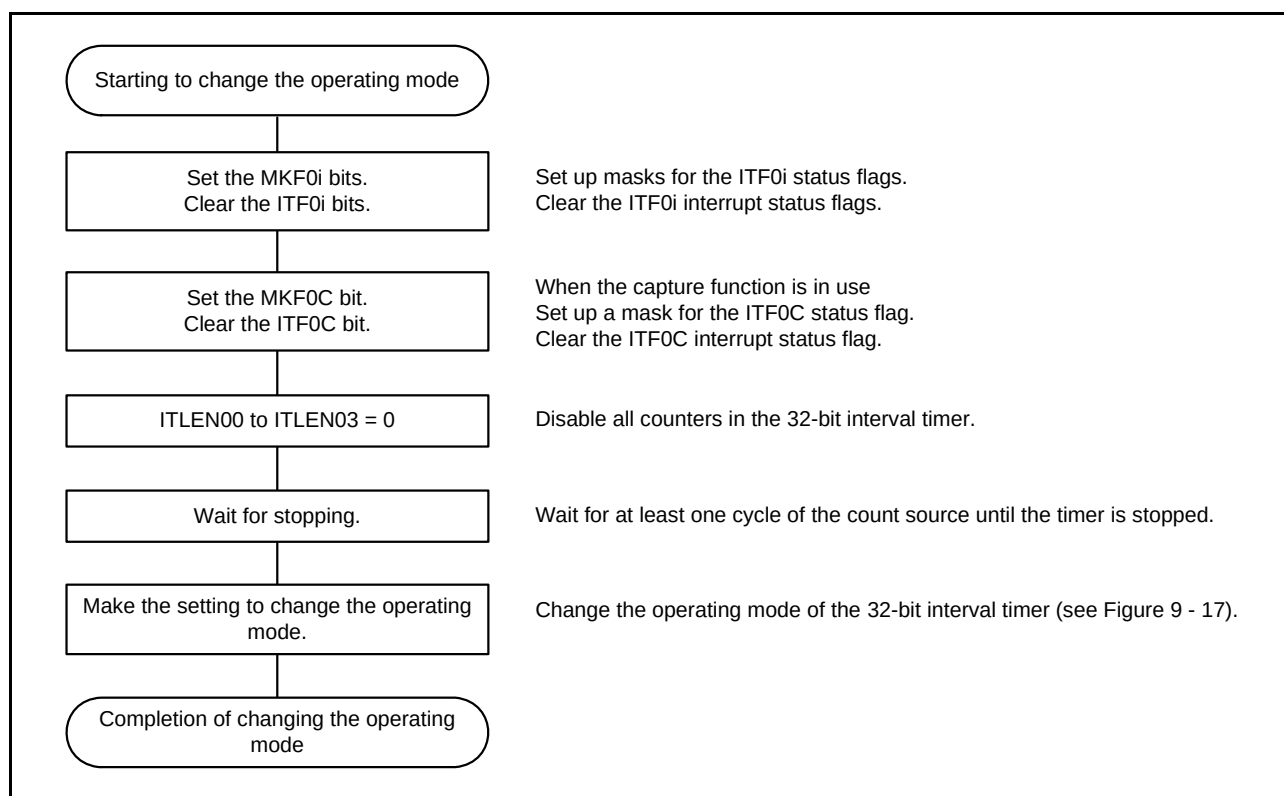


Figure 9 - 20 Procedure for Resetting the 32-bit Interval Timer

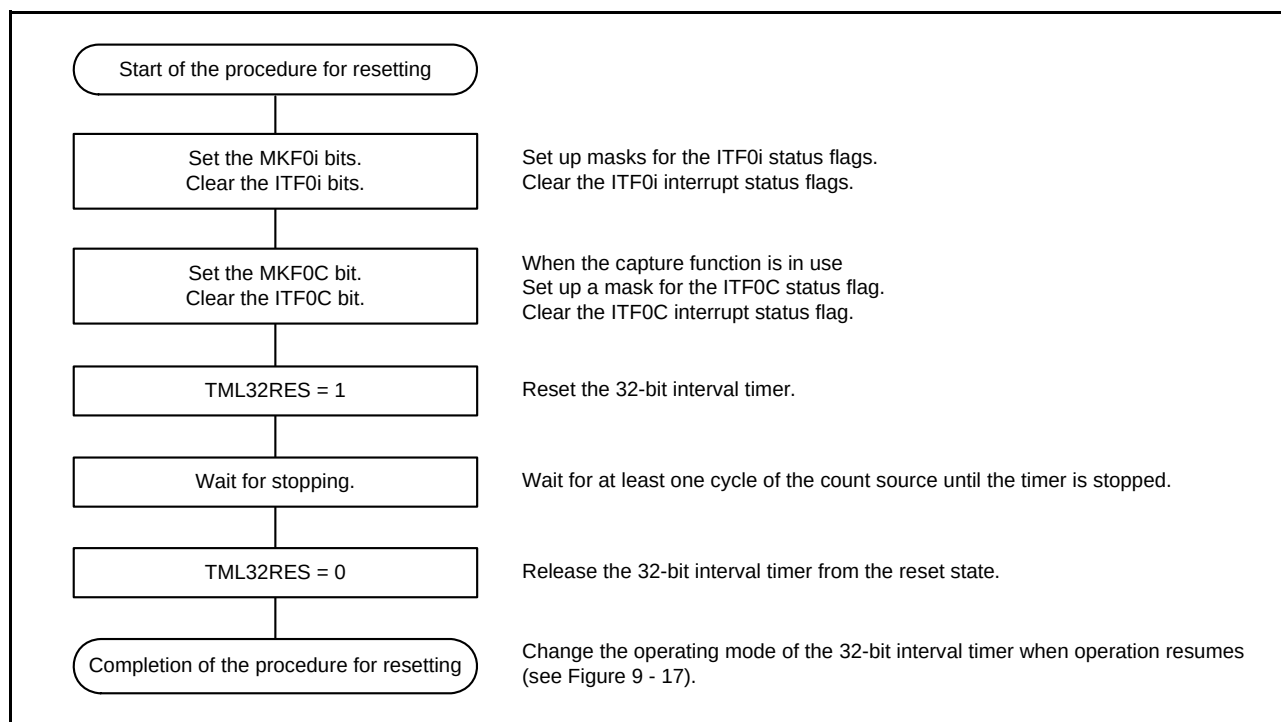


Figure 9 - 21 Procedure for Starting Event Input from the ELC

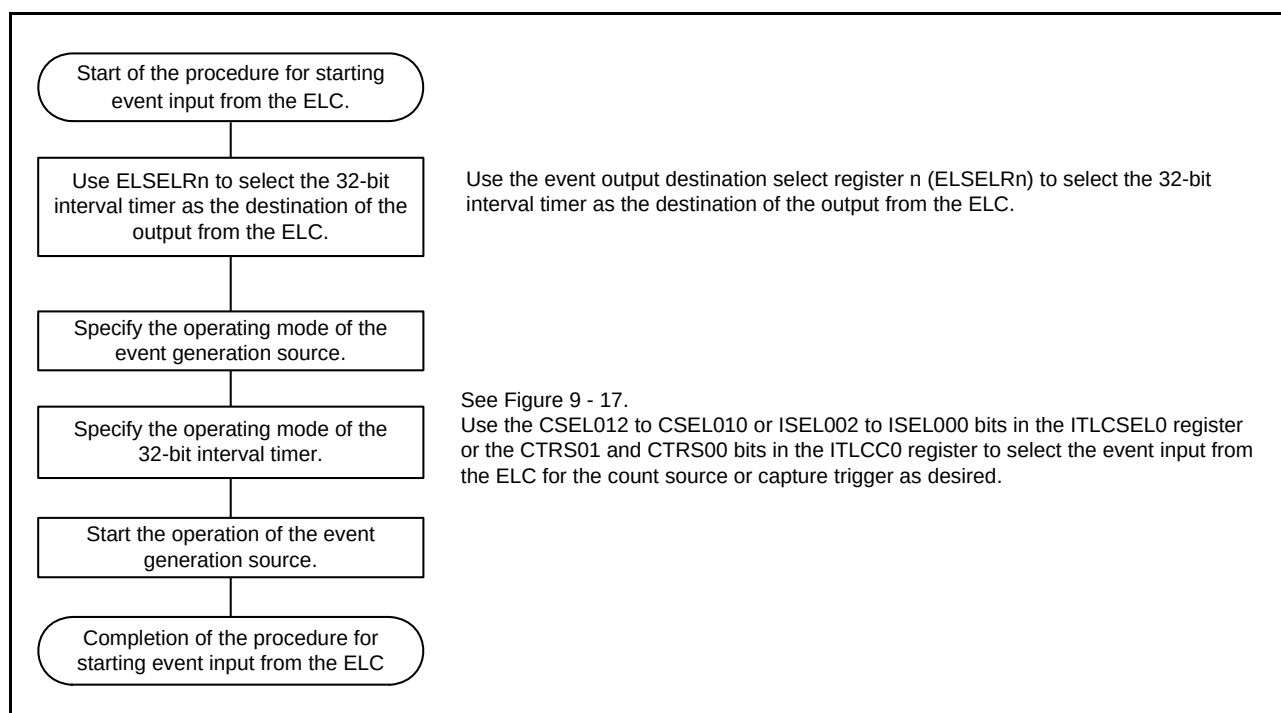
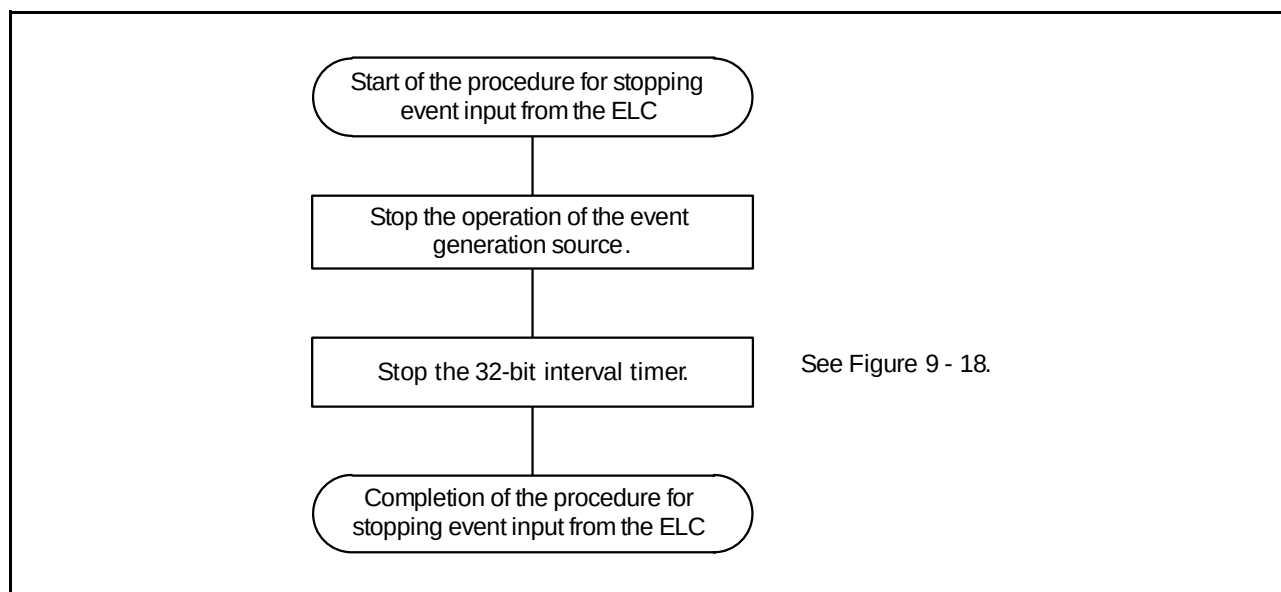


Figure 9 - 22 Procedure for Stopping Event Input from the ELC



Section 10 Clock Output/Buzzer Output Controller (PCLBUZ)

Caution Most of the following descriptions in this section use the 48-pin products as an example.

10.1 Functions of Clock Output/Buzzer Output Controller

In clock output, the controller outputs a clock signal for supply to peripheral ICs. In buzzer output, the controller outputs a square wave at the buzzer frequency. This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. Switching of the output is handled by clock output select registers n (CKSn).

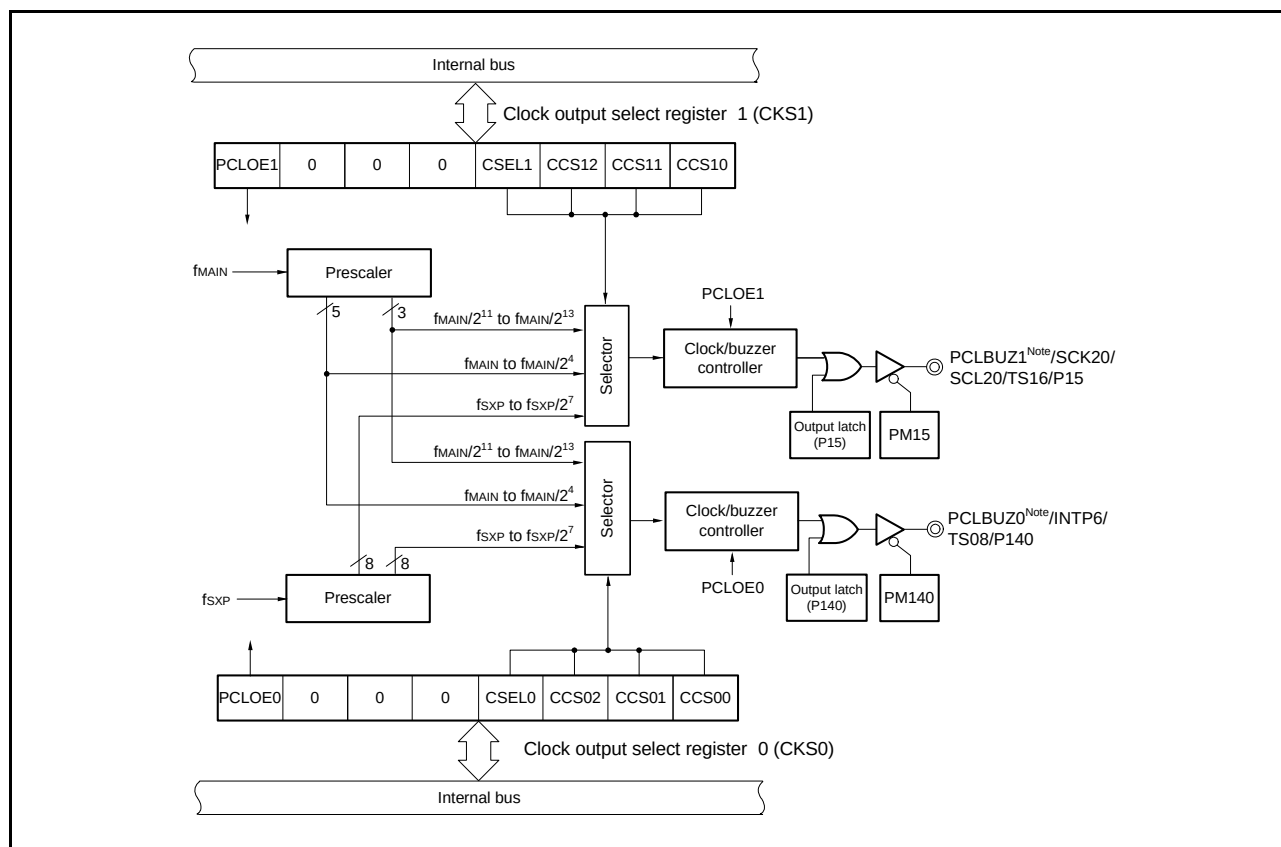
Figure 10 - 1 shows the block diagram of clock output/buzzer output controller.

Caution Output of the low-speed peripheral clock (fsxp) from the PCLBUZn pin is not possible when the following conditions are both satisfied:

- The setting of the RTCLPC bit in the subsystem clock supply mode control register (OSMC) is 1.
- Operation is in the HALT mode with the subsystem clock (fsub) selected as the CPU clock.

Remark n = 0, 1

Figure 10 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **34.4 AC Characteristics**.

Remark The clock output/buzzer output pins in above diagram show the information of 48-pin products with PIOR3 = 0.

10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 10 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode registers 1, 3, 14 (PM1, PM3, PM14) Port registers 1, 3, 14 (P1, P3, P14) Port mode control T registers 1, 3, 14 (PMCT1, PMCT3, PMCT14)

10.3 Registers for Controlling the Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port mode control T registers (PMCTxx)

10.3.1 Clock output select registers n (CKSn)

The CKSn registers enable or disable the output from the clock or buzzer frequency output pin (PCLBUZn), and set the output clock. Use the CKSn register to select the clock to be output from the PCLBUZn pin.

The CKSn registers are set by a 1-bit or 8-bit memory manipulation instruction. The value of each CKSn register following a reset is 00H.

Figure 10 - 2 Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1)
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification							
0	Output disabled (default)							
1	Output enabled							

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
					f _{MAIN} = 5 MHz	f _{MAIN} = 10 MHz	f _{MAIN} = 20 MHz	f _{MAIN} = 32 MHz
0	0	0	0	f _{MAIN}	5 MHz ^{Note}	10 MHz ^{Note}	Setting prohibited	Setting prohibited
0	0	0	1	f _{MAIN} /2	2.5 MHz	5 MHz ^{Note}	10 MHz ^{Note}	16 MHz ^{Note}
0	0	1	0	f _{MAIN} /2 ²	1.25 MHz	2.5 MHz	5 MHz ^{Note}	8 MHz ^{Note}
0	0	1	1	f _{MAIN} /2 ³	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	f _{MAIN} /2 ⁴	312.5 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	f _{MAIN} /2 ¹¹	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz
0	1	1	0	f _{MAIN} /2 ¹²	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
0	1	1	1	f _{MAIN} /2 ¹³	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	0	0	0	f _{SXP}	32.768 kHz			
1	0	0	1	f _{SXP} /2	16.384 kHz			
1	0	1	0	f _{SXP} /2 ²	8.192 kHz			
1	0	1	1	f _{SXP} /2 ³	4.096 kHz			
1	1	0	0	f _{SXP} /2 ⁴	2.048 kHz			
1	1	0	1	f _{SXP} /2 ⁵	1.024 kHz			
1	1	1	0	f _{SXP} /2 ⁶	512 Hz			
1	1	1	1	f _{SXP} /2 ⁷	256 Hz			

Note The selectable output clock frequency depends on the power supply voltage (V_{DD}). See **34.4 AC Characteristics** for details.

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).

Caution 2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register is set to 0 and moreover while STOP mode is set.

Caution 3. It is not possible to output the low-speed peripheral clock (f_{SXP}) from the PCLBUZn pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC) is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{SXP}: Low-speed peripheral clock frequency

10.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the following registers that control the port functions multiplexed on the target pin.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port mode control T registers (PMCTxx)

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.8 Port mode control T registers (PMCTxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P140/INTP6/TS08/PCLBUZ0, P15/PCLBUZ1/SCK20/SCL20/TS16) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx), port register (Pxx), and port mode control T register (PMCTxx) to 0.

Example: When P140/INTP6/TS08/PCLBUZ0 is to be used for clock or buzzer output
Set the PM140 bit of port mode register 14 to 0.
Set the P140 bit of port register 14 to 0.

10.4 Operations of the Clock Output/Buzzer Output Controller

This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. The PCLBUZ0 and PCLBUZ1 pins respectively output the clock specified by clock output select registers 0 and 1 (CKS0 and CKS1).

10.4.1 Operation of output pins

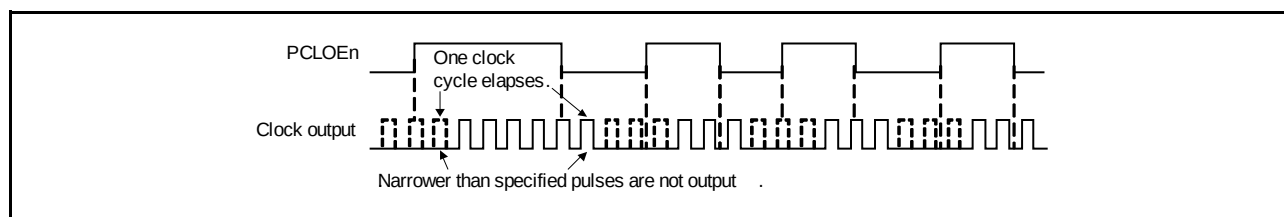
Follow the steps below to enable output from a PCLBUZn pin.

1. Set the corresponding bits of the following registers to 0 to select the port pin for use as a PCLBUZn pin.
 - Port mode register (PMxx)
 - Port register (Pxx)
 - Port mode control T register (PMCTxx)
2. Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) for a PCLBUZn pin (output is still disabled).
3. Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for the clock output starts or stops output one clock cycle after that in which the PCLOEn bit was set to enable or disable the output, respectively. Pulses that are narrower than the specified width are not output at those times. **Figure 10 - 3** shows the relationship between the setting of the PCLOEn bit and the timing of clock output.

Remark 2. n = 0, 1

Figure 10 - 3 Timing of the Clock Output from a PCLBUZn Pin



10.5 Point for Caution when the Clock Output/Buzzer Output Controller is to be Used

When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles of output from a PCLBUZn pin having been disabled (PCLOEn = 0), the width of a PCLBUZn pulse being output at that time becomes shorter.

Section 11 Watchdog Timer (WDT)

11.1 Functions of Watchdog Timer

Counting by the watchdog timer is set by the option byte (000C0H).

The watchdog timer operates with the low-speed on-chip oscillator clock (f_{IL}) divided by 2 ($1/2 f_{IL}$).

The watchdog timer is used to detect program malfunctions. If a malfunction is detected, an internal reset signal is generated. Any among the following cases is considered a program malfunction.

- The watchdog timer counter overflows
- A 1-bit manipulation instruction is used to write to the watchdog timer enable register (WDTE).
- A value other than ACH is written to the WDTE register.
- Writing to the WDTE register proceeds while the window is closed.

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **Section 21 Reset Function**.

When 75% of the overflow time + $1/4 f_{IL}$ is reached, an interval interrupt can be generated.

11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 11 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

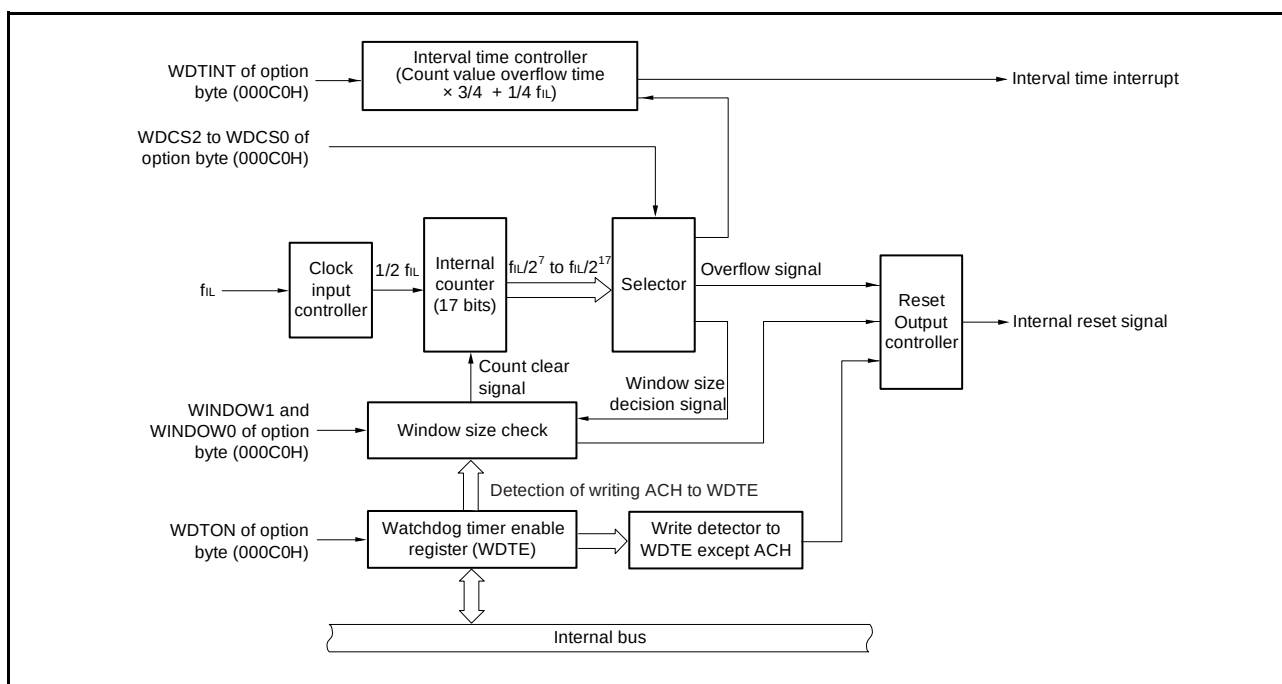
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 11 - 2 Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **Section 29 Option Bytes**.

Figure 11 - 1 Block Diagram of Watchdog Timer



Remark f_{IL}: Low-speed on-chip oscillator clock

11.3 Register for Controlling the Watchdog Timer

The following register is used to control the watchdog timer.

- Watchdog timer enable register (WDTE)

11.3.1 Watchdog timer enable register (WDTE)

Writing ACH to the WDTE register clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 9AH or 1AH^{Note}.

Figure 11 - 2 Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH
After reset: 9AH/1AH^{Note}
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The value of the WDTE register following a reset depends on the setting of the WDTON bit of the option byte (000C0H). To enable counting by the watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	Value of the WDTE Register following a Reset
0 (Watchdog timer counting disabled)	1AH
1 (Watchdog timer counting enabled)	9AH

Caution 1. If a value other than ACH is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 9AH or 1AH. Note that the written value (ACH) is not read.

11.4 Operation of Watchdog Timer

11.4.1 Controlling operation of watchdog timer

- When the watchdog timer is used, its operation is specified by the option byte (000C0H).
 - Enable counting by the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release). For details, see **Section 29 Option Bytes**.

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

- Set the time at which the counter is to overflow by using bits 3 to 1 (WDCS2 to WDCS0) of an option byte (000C0H). For details, see **11.4.2 Setting overflow time of watchdog timer** and **Section 29 Option Bytes**.
 - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H). For details, see **11.4.3 Setting window open period of watchdog timer** and **Section 29 Option Bytes**.
- After a reset release, the watchdog timer starts counting.
 - By writing ACH to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
 - After that, writing to the WDTE register the second and subsequent times must proceed while the window is open. If the WDTE register is written during a window close period, an internal reset signal is generated. If the time at which an overflow is to occur elapses without ACH having been written to the WDTE register, an internal reset signal is generated.
 - An internal reset signal is also generated in the following cases.
 - If a 1-bit manipulation instruction is executed on the WDTE register
 - If data other than ACH is written to the WDTE register

Caution 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Caution 2. After ACH is written to the WDTE register, an error of up to 4 cycles of the clock at f_{IL} may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the counter value overflows.

Caution 4. The operation of the watchdog timer in the HALT, STOP, and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.

When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.

Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.

Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

11.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing ACH to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 11 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f _{IL} = 37.683 kHz (max.))
0	0	0	2 ⁷ /f _{IL} (3.39 ms)
0	0	1	2 ⁸ /f _{IL} (6.79 ms)
0	1	0	2 ⁹ /f _{IL} (13.58 ms)
0	1	1	2 ¹⁰ /f _{IL} (27.17 ms)
1	0	0	2 ¹² /f _{IL} (108.69 ms)
1	0	1	2 ¹⁴ /f _{IL} (434.78 ms) ^{Note}
1	1	0	2 ¹⁵ /f _{IL} (869.56 ms) ^{Note}
1	1	1	2 ¹⁷ /f _{IL} (3478.26 ms) ^{Note}

Note Using the watchdog timer under the following conditions may lead to the generation of an interval interrupt (INTWDTI) after one cycle of the watchdog timer clock once the watchdog timer counter has been cleared.

Usage conditions that may lead to the generation of an interval interrupt:

- the watchdog timer interval interrupt is in use, and
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.

1. Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
2. Clear the watchdog timer counter.
3. Wait for at least 80 μs.
4. Clear the WDTIIF bit of the interrupt request flag register 0 (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

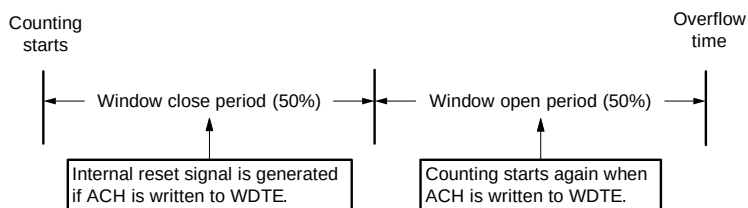
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

11.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If ACH is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if ACH is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 11 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	1	50%
1	1	100%
Others		Setting prohibited

Remark If the overflow time is set to $2^{10}/f_{IL}$, the times over which the window is open and closed are as follows.

	Setting of Window Open Period	
	50%	100%
Window close time	0 to 18.38 ms	None
Window open time	18.38 to 27.17 ms	0 to 27.17 ms

<When window open period is 50%>

- Overflow time:
 $2^{10}/f_{IL} \text{ (max.)} = 2^9/18.842 \text{ kHz} = 27.17 \text{ ms}$
- Window close time:
 $0 \text{ to } 2^{10}/f_{IL} \text{ (min.)} \times (1 - 0.5) = 0 \text{ to } 2^{10}/13.926 \text{ kHz} \times 0.5 = 0 \text{ to } 18.38 \text{ ms}$
- Window open time:
 $2^{10}/f_{IL} \text{ (min.)} \times (1 - 0.5) \text{ to } 2^{10}/f_{IL} \text{ (max.)} = 2^9/13.926 \text{ kHz} \times 0.5 \text{ to } 2^9/18.842 \text{ kHz} = 18.38 \text{ to } 27.17 \text{ ms}$

11.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% of the overflow time + 1/4 f_{IL} is reached.

Table 11 - 5 Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time + 1/4 f _{IL} is reached.

Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated until ACH is written to the watchdog timer enable register (WDTE). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

Section 12 A/D Converter (ADC)

The number of analog input channels of the A/D converter differs, depending on the product.

	16-pin	20-, 24-, and 25-pin	30- and 32-pin	36-pin	40-pin	44- and 48-pin
Number of the analog input channels	3 (ANI0 to ANI2)	6 (ANI0 to ANI2, ANI16 to ANI18)	8 (ANI0 to ANI3, ANI16 to ANI19)	8 (ANI0 to ANI5, ANI18, ANI19)	9 (ANI0 to ANI6, ANI18, ANI19)	10 (ANI0 to ANI7, ANI18, ANI19)

12.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control analog inputs, including up to 10 channels of A/D converter analog inputs (ANI0 to ANI7 and ANI16 to ANI19). 10-bit or 8-bit resolution can be selected by the ADTYP bit of A/D converter mode register 2 (ADM2). The A/D converter has the following function.

- 10-bit/8-bit resolution A/D conversion

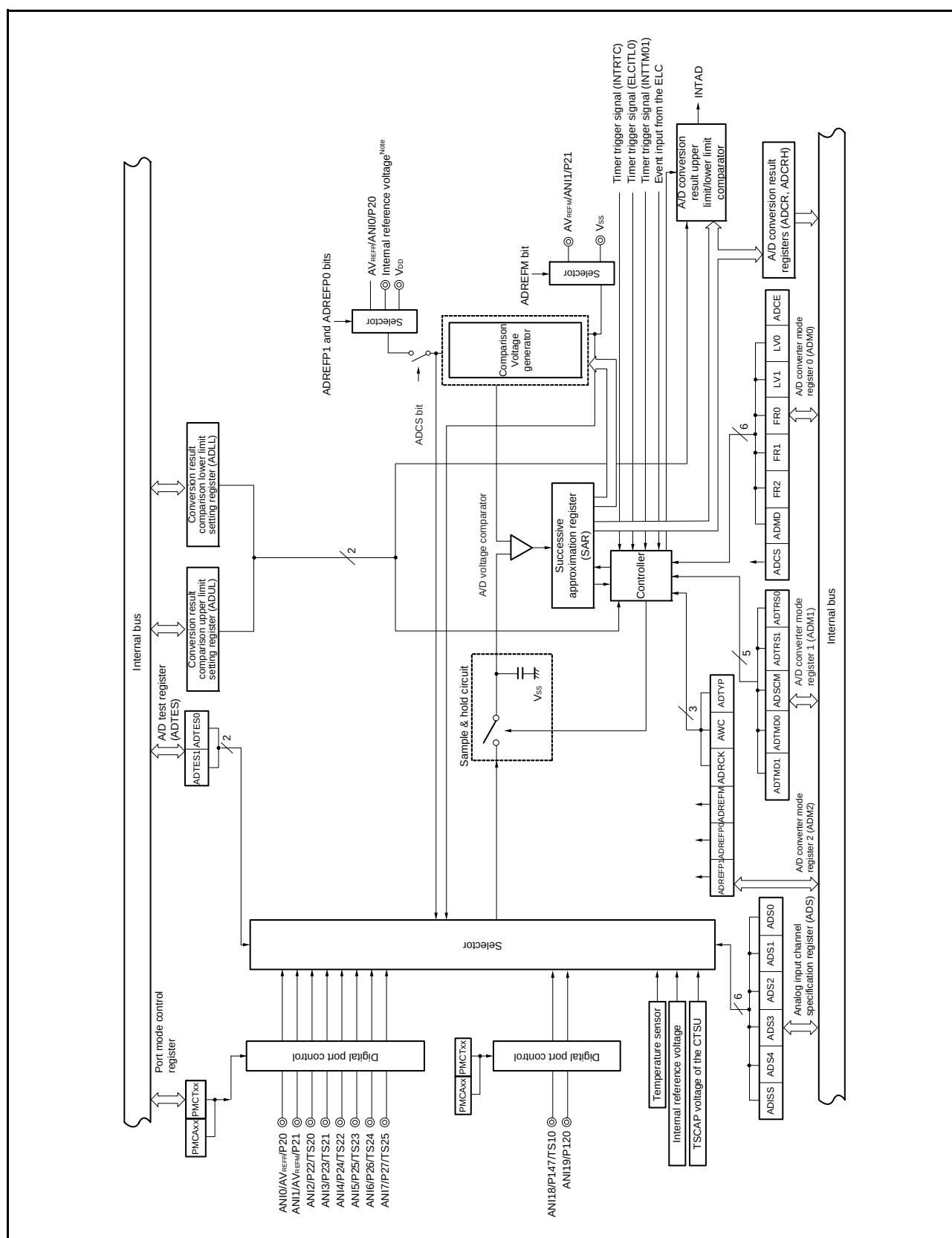
10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI7 and ANI16 to ANI19. Each time an A/D conversion operation ends, an interrupt request signal (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power to the A/D converter is turned on by detecting a hardware trigger while the power is off and the A/D converter is in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time has passed. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI7 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion operations are possible within the operating voltage range $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$.
	Low voltage 1 or low voltage 2 mode	Conversion operations are possible within the operating voltage range $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 or low voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Note The operation mode that can be selected differs depending on the analog input channel, V_{DD} voltage, AV_{REFP} voltage, trigger mode, and f_{CLK} . See **Table 12 - 3 A/D Conversion Time Selection (1/4)** for details.

Figure 12 - 1 Block Diagram of A/D Converter



Note For details about the internal reference voltage, see **Section 34 Electrical Characteristics** for $T_A = -40$ to $+105^{\circ}\text{C}$.

Remark Analog input pins in this figure are for a 48-pin product.

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 and ANI16 to ANI19 pins

These are the analog input pins of the 12 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage ($1/2 AVREF$) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage ($1/2 AVREF$), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: ($1/4 AVREF$)

Bit 9 = 1: ($3/4 AVREF$)

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage $\geq$ Voltage tap of comparison voltage generator: Bit 8 = 1

Analog input voltage $\leq$ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AVREF: The positive reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage^{Note}, and VDD.

Note For details about the internal reference voltage, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.

(9) AVREFP pin

This pin inputs an external reference voltage (AVREFP).

If using AVREFP as the positive reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI2 to ANI7 and ANI16 to ANI19 are converted to digital signals based on the voltage applied between AVREFP and the negative reference voltage (AVREFM/VSS).

In addition to AVREFP, it is possible to select VDD or the internal reference voltage as the positive reference voltage of the A/D converter.

Caution For details about the internal reference voltage, see Section 34 Electrical Characteristics for TA = -40 to +105°C.

(10) AVREFM pin

This pin inputs an external reference voltage (AVREFM). If using AVREFM as the negative reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AVREFM, it is possible to select VSS as the negative reference voltage of the A/D converter.

12.3 Registers for Controlling the A/D Converter

The following registers are used to control the A/D converter.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode registers (PMxx)

12.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the A/D converter is to be used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	Control of supply of an input clock to the A/D converter
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written. • When an SFR used by the A/D converter is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.

Caution 1. When setting the A/D converter, make sure that the setting of the ADCEN bit is 1 before setting the following registers. If ADCEN = 0, the values of the registers which control the A/D converter are cleared to 00H and writing to them is ignored (except for port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, and PM14), port mode control A registers 0, 2, 12, and 14 (PMCA0, PMCA2, PMCA12, and PMCA14), and port mode control T registers 0, 2, and 14 (PMCT0, PMCT2, and PMCT14)).

- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES).

Caution 2. Be sure to clear the following bits to 0.

For the presence of bits for each product, see Section 6 Clock Generator.

Bits 1, 3, 4, and 6 in the 16- and 20-pin products

Bits 1, 3, and 6 in the 24- and 25-pin products

Bits 1 and 6 in the 30-, 32-, 36-, 40-, and 48-pin products

12.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the A/D converter in the reset state, be sure to set bit 5 (ADCRES) of this register to 1.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 3 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

ADCRES	Control resetting of the A/D converter
0	The A/D converter is released from the reset state.
1	The A/D converter is in the reset state. • The SFRs for use with the A/D converter are initialized.

Caution 1. Be sure to clear bits 7, 6, and 1 to 0.

Caution 2. The functions that are mounted depend on the product. For details on the PRR0 register, see the description in Section 21 Reset Function.

12.3.3 A/D converter mode register 0 (ADM0)

The ADM0 register sets the time for converting analog input to digital data, and starts and stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 4 Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 ^{Note 1}	FR1 ^{Note 1}	FR0 ^{Note 1}	LV1 ^{Note 1}	LV0 ^{Note 1}	ADCE
ADCS		A/D conversion operation control						
0		Stops conversion operation [When read] Conversion is stopped or in standby.						
1		Enables conversion operation [When read] While in the software trigger mode: Conversion in progress While in the hardware trigger wait mode: A/D power supply stabilization wait state + conversion in progress						
ADMD		Specification of the A/D conversion channel selection mode						
0		Select mode						
1		Scan mode						
ADCE		A/D voltage comparator operation control ^{Note 2}						
0		Stops A/D voltage comparator operation						
1		Enables A/D voltage comparator operation						

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 12 - 3 A/D Conversion Time Selection (1/4)**.

Note 2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μ s from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μ s or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

Caution 3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 12.7 A/D Converter Setup Flowchart.

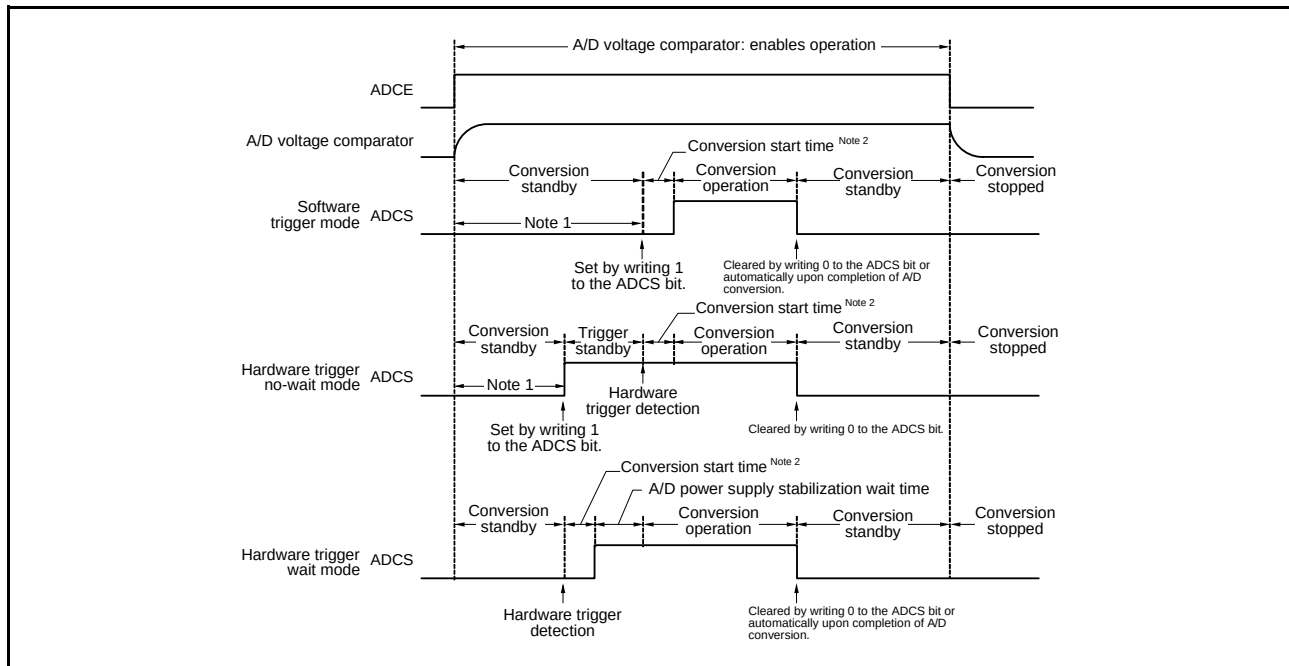
Table 12 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 12 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		- When 0 is written to ADCS - The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		- When 0 is written to ADCS - The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		- When 0 is written to ADCS - The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		- When 0 is written to ADCS - The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 12 - 5 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

Note 2. The following shows the maximum time to start conversion.

ADM0			Conversion Clock (f _{AD})	Conversion Start Time (Number of f _{CLK} Clock Cycles)	
FR2	FR1	FR0		Software Trigger Mode/ Hardware Trigger No-wait Mode	Hardware Trigger Wait Mode
0	0	0	f _{CLK} /64	63	1
0	0	1	f _{CLK} /32	31	
0	1	0	f _{CLK} /16	15	
0	1	1	f _{CLK} /8	7	
1	0	0	f _{CLK} /6	5	
1	0	1	f _{CLK} /5	4	
1	1	0	f _{CLK} /4	3	
1	1	1	f _{CLK} /2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Caution 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby state.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS bit is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby state).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 f_{CLK} clock + A/D conversion time

Hardware trigger wait mode: 2 f_{CLK} clock + stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (1/4)

(1) When there is no A/D power supply stabilization wait time
Normal modes 1 and 2 (in software trigger mode and hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Conversion Clock ^{Note}	Conversion Time	Conversion Time at 10-Bit Resolution						
FR2	FR1	FR0	LV1	LV0					2.7 V ≤ VDD ≤ 5.5 V						
									fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz		
0	0	0	0	0	Normal 1	fCLK/64	19 fAD (number of sampling clock: 7 fAD)	1216/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	38 μs		
0	0	1				fCLK/32		608/fCLK				76 μs	38 μs	19 μs	
0	1	0				fCLK/16		304/fCLK				76 μs	38 μs	19 μs	9.5 μs
0	1	1				fCLK/8		152/fCLK				38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				fCLK/6		114/fCLK				28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				fCLK/5		95/fCLK	95 μs	23.75 μs	11.875 μs	5.938 μs	2.9688 μs		
1	1	0				fCLK/4		76/fCLK	76 μs	19 μs	9.5 μs	4.75 μs	2.375 μs		
1	1	1				fCLK/2		38/fCLK	38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited		
0	0	0	0	1	Normal 2	fCLK/64	17 fAD (number of sampling clock: 5 fAD)	1088/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	34 μs		
0	0	1				fCLK/32		544/fCLK				68 μs	34 μs	17 μs	
0	1	0				fCLK/16		272/fCLK				68 μs	34 μs	17 μs	8.5 μs
0	1	1				fCLK/8		136/fCLK				34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				fCLK/6		102/fCLK				25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				fCLK/5		85/fCLK	85 μs	21.25 μs	10.625 μs	5.3125 μs	2.6563 μs		
1	1	0				fCLK/4		68/fCLK	68 μs	17 μs	8.5 μs	4.25 μs	2.125 μs		
1	1	1				fCLK/2		34/fCLK	34 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited		

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tconv) described in 34.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. Use normal mode 2 when the internal reference voltage, TSCAP voltage of the CTSU, or temperature sensor output voltage is selected as the target for A/D conversion.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.

Remark fCLK: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time

Low-voltage modes 1 and 2 (in software trigger mode and hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f _{AD})	Number of Conversion Clock ^{Note}	Conversion Time	Conversion Time at 10-Bit Resolution				
									1.6 V ≤ V _{DD} ≤ 5.5 V		1.8 V ≤ V _{DD} ≤ 5.5 V	2.4 V ≤ V _{DD} ≤ 5.5 V	2.7 V ≤ V _{DD} ≤ 5.5 V
									f _{CLK} = 1 MHz	f _{CLK} = 4 MHz	f _{CLK} = 8 MHz	f _{CLK} = 16 MHz	f _{CLK} = 32 MHz
FR2	FR1	FR0	LV1	LV0	Low voltage 1	f _{CLK} /64	19 f _{AD} (number of sampling clock: 7 f _{AD})	1216/f _{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	38 μs
0	0	0	1	0		f _{CLK} /32		608/f _{CLK}				76 μs	38 μs
0	0	1				f _{CLK} /16		304/f _{CLK}				76 μs	38 μs
0	1	0				f _{CLK} /8		152/f _{CLK}				38 μs	19 μs
0	1	1				f _{CLK} /6		114/f _{CLK}				38 μs	19 μs
1	0	0				f _{CLK} /5		95/f _{CLK}				28.5 μs	14.25 μs
1	0	1				f _{CLK} /4		76/f _{CLK}				19 μs	9.5 μs
1	1	0				f _{CLK} /2		38/f _{CLK}				9.5 μs	4.75 μs
0	0	0	1	1	Low voltage 2	f _{CLK} /64	17 f _{AD} (number of sampling clock: 5 f _{AD})	1088/f _{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	34 μs
0	0	1				f _{CLK} /32		544/f _{CLK}				68 μs	34 μs
0	1	0				f _{CLK} /16		272/f _{CLK}				68 μs	34 μs
0	1	1				f _{CLK} /8		136/f _{CLK}				34 μs	17 μs
1	0	0				f _{CLK} /6		102/f _{CLK}				34 μs	17 μs
1	0	1				f _{CLK} /5		85/f _{CLK}				25.5 μs	12.75 μs
1	1	0				f _{CLK} /4		68/f _{CLK}				12.75 μs	6.375 μs
1	1	1				f _{CLK} /2		34/f _{CLK}				10.625 μs	5.3125 μs
0	0	0	1	0	Low voltage 2	f _{CLK} /64	17 f _{AD} (number of sampling clock: 5 f _{AD})	1088/f _{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	34 μs
0	0	1				f _{CLK} /32		544/f _{CLK}				68 μs	34 μs
0	1	0				f _{CLK} /16		272/f _{CLK}				68 μs	34 μs
0	1	1				f _{CLK} /8		136/f _{CLK}				34 μs	17 μs
1	0	0				f _{CLK} /6		102/f _{CLK}				34 μs	17 μs
1	0	1				f _{CLK} /5		85/f _{CLK}				25.5 μs	12.75 μs
1	1	0				f _{CLK} /4		68/f _{CLK}				12.75 μs	6.375 μs
1	1	1				f _{CLK} /2		34/f _{CLK}				10.625 μs	5.3125 μs

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{conv}) described in 34.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When the internal reference voltage, TSCAP voltage of the CTSU, or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (3/4)

(3) When there is no A/D power supply stabilization wait time
Normal modes 1 and 2 (in hardware trigger wait mode^{Note 1})

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f _{CLK})	Number of Stabilization Wait Clock	Number of Conversion Clock ^{Note 2}	Stabilization Wait Time + Conversion Time	Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
FR2	FR1	FR0	LV1	LV0						2.7 V ≤ V _{DD} ≤ 5.5 V				
										f _{CLK} = 1 MHz	f _{CLK} = 4 MHz	f _{CLK} = 8 MHz	f _{CLK} = 16 MHz	f _{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f _{CLK} /64	8 f _{AD}	19 f _{AD} (number of sampling clock: 7 f _{AD})	1728/f _{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	108 μs	54 μs
0	0	1				f _{CLK} /32			864/f _{CLK}			108 μs	54 μs	27 μs
0	1	0				f _{CLK} /16			432/f _{CLK}		108 μs	54 μs	27 μs	13.5 μs
0	1	1				f _{CLK} /8			216/f _{CLK}		54 μs	27 μs	13.5 μs	6.75 μs
1	0	0				f _{CLK} /6			162/f _{CLK}		40.5 μs	20.25 μs	10.125 μs	5.0625 μs
1	0	1				f _{CLK} /5			135/f _{CLK}	135 μs	33.75 μs	16.875 μs	8.4375 μs	4.21875 μs
1	1	0				f _{CLK} /4			108/f _{CLK}	108 μs	27 μs	13.5 μs	6.75 μs	3.375 μs
1	1	1				f _{CLK} /2			54/f _{CLK}	54 μs	13.5 μs	6.75 μs	3.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f _{CLK} /64	8 f _{AD}	17 f _{AD} (number of sampling clock: 5 f _{AD})	1600/f _{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	100 μs	50 μs
0	0	1				f _{CLK} /32			800/f _{CLK}			100 μs	50 μs	25 μs
0	1	0				f _{CLK} /16			400/f _{CLK}		100 μs	50 μs	25 μs	12.5 μs
0	1	1				f _{CLK} /8			200/f _{CLK}		50 μs	25 μs	12.5 μs	6.25 μs
1	0	0				f _{CLK} /6			150/f _{CLK}		37.5 μs	18.75 μs	9.375 μs	4.6875 μs
1	0	1				f _{CLK} /5			125/f _{CLK}	125 μs	31.25 μs	15.625 μs	7.8125 μs	3.90625 μs
1	1	0				f _{CLK} /4			100/f _{CLK}	100 μs	25 μs	12.5 μs	6.25 μs	3.125 μs
1	1	1				f _{CLK} /2			50/f _{CLK}	50 μs	12.5 μs	6.25 μs	3.125 μs	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 3 A/D Conversion Time Selection (1/4)**).

Note 2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

Caution 5. Use normal mode 2 when the internal reference voltage, TSCAP voltage of the CTSU, or temperature sensor output voltage is selected as the target for A/D conversion.

Caution 6. When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (4/4)

(4) When there is no A/D power supply stabilization wait time
Low-voltage modes 1 and 2 (in hardware trigger wait mode^{Note 1})

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Stabilization Wait Clock	Number of Conversion Clock ^{Note 2}	Stabilization Wait Time + Conversion Time	Stabilization Wait Time + Conversion Time at 10-Bit Resolution							
										1.6 V ≤ VDD ≤ 5.5 V		1.8 V ≤ VDD ≤ 5.5 V	2.4 V ≤ VDD ≤ 5.5 V	2.7 V ≤ VDD ≤ 5.5 V			
FR2	FR1	FR0	LV1	LV0						fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz			
0	0	0	1	0	Low voltage 1	fCLK/64	2 fAD	19 fAD (number of sampling clock: 7 fAD)	1344/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	84 μs	42 μs			
0	0	1				fCLK/32			672/fCLK				84 μs	42 μs	21 μs		
0	1	0				fCLK/16			336/fCLK				84 μs	42 μs	21 μs	10.5 μs	
0	1	1				fCLK/8			168/fCLK				42 μs	21 μs	10.5 μs	5.25 μs	
1	0	0				fCLK/6			126/fCLK				31.5 μs	15.75 μs	7.875 μs	3.9375 μs	
1	0	1				fCLK/5			105/fCLK				105 μs	26.25 μs	13.125 μs	6.5625 μs	3.238125 μs
1	1	0				fCLK/4			84/fCLK				84 μs	21 μs	10.5 μs	5.25 μs	2.625 μs
1	1	1				fCLK/2			42/fCLK				42 μs	10.5 μs	5.25 μs	2.625 μs	Setting prohibited
0	0	0	1	1	Low voltage 2	fCLK/64	2 fAD	17 fAD (number of sampling clock: 5 fAD)	1216/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	38 μs			
0	0	1				fCLK/32			608/fCLK				76 μs	38 μs	19 μs		
0	1	0				fCLK/16			304/fCLK				76 μs	38 μs	19 μs	9.5 μs	
0	1	1				fCLK/8			152/fCLK				38 μs	19 μs	9.5 μs	4.75 μs	
1	0	0				fCLK/6			114/fCLK				28.5 μs	14.25 μs	7.125 μs	3.5625 μs	
1	0	1				fCLK/5			95/fCLK				96 μs	23.75 μs	11.88 μs	5.938 μs	2.9688 μs
1	1	0				fCLK/4			76/fCLK				76 μs	19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				fCLK/2			38/fCLK				38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 12 - 3 A/D Conversion Time Selection (2/4)**).

Note 2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

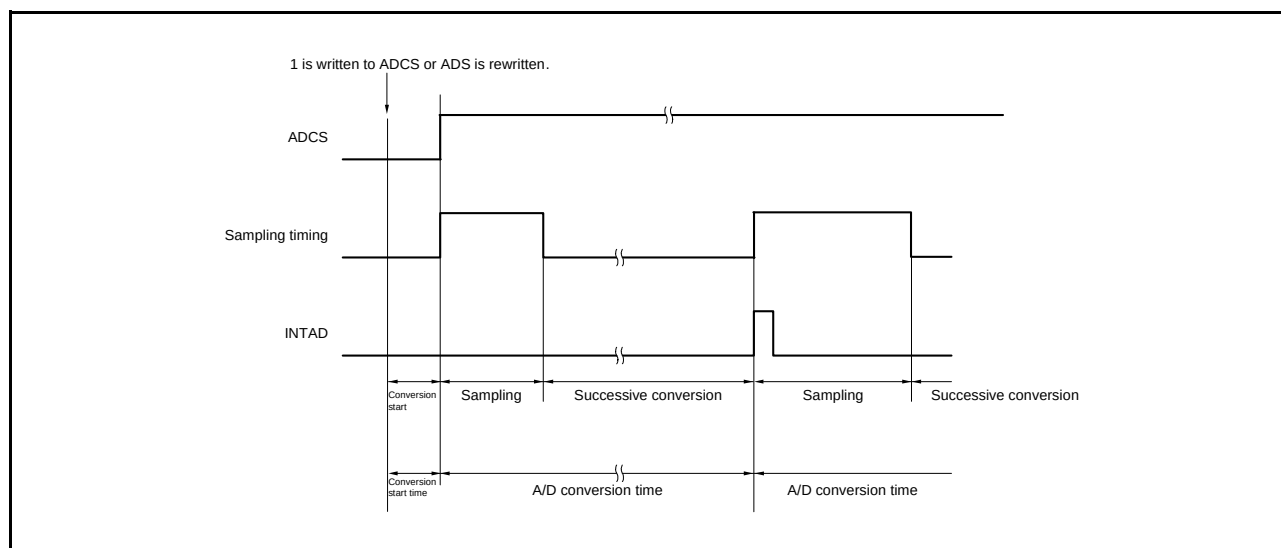
Caution 4. When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

Caution 5. When the internal reference voltage, TSCAP voltage of the CTSU, or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.

Caution 6. When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12 - 6 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



12.3.4 A/D converter mode register 1 (ADM1)

The ADM1 register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 7 Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 01 count or capture interrupt signal (INTTM01)
0	1	Event input from the ELC
1	0	Realtime clock interrupt signal (INTRTC)
1	1	32-bit interval timer interrupt signal (ELCITL0)

Caution 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 fCLK clock + conversion start time + A/D conversion time

Hardware trigger wait mode: 2 fCLK clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE mode, input of the next INTRTC or ELCITL0 will not be recognized as a valid hardware trigger for up to four fCLK cycles after the first INTRTC or ELCITL0 is input.

Remark 1. ×: Don't care

Remark 2. fCLK: CPU/peripheral hardware clock frequency

12.3.5 A/D converter mode register 2 (ADM2)

The ADM2 register is used to select the positive and negative reference voltages of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode. The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the positive reference voltage source of the A/D converter
0	0	Supplied from VDD
0	1	Supplied from P20/AVREFP/ANI0
1	0	Supplied from the internal reference voltage ^{Note 1}
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
 - Set ADCE = 0
 - Change the values of ADREFP1 and ADREFP0
 - Reference voltage stabilization wait time (A)
 - Set ADCE = 1
 - Reference voltage stabilization wait time (B)
 When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μ s, B = 1 μ s.
 When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μ s.
 After (5) stabilization time, start the A/D conversion.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage^{Note 1}.
 Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the negative reference voltage of the A/D converter
0	Supplied from Vss
1	Supplied from P21/AVREFM/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register $\leq$ the ADCR register $\leq$ the ADUL register (AREA 1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA 2) or the ADUL register < the ADCR register (AREA 3).

Figure 12 - 9 shows the generation range of the interrupt signal (INTAD) for AREA 1 to AREA 3.

(Notes and Cautions are listed on the next page.)

Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) (2/2)

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). - The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (fCLK). If any other clock is selected, specifying this mode is prohibited. - Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited. - Using the SNOOZE mode function in the sequential conversion mode is prohibited. - When using the SNOOZE mode function, specify a hardware trigger interval of at least "shift time to SNOOZE mode^{Note 2} + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 fCLK clock" - Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation and change it to 1 just before shifting to STOP mode. Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE mode or normal operation.	
ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note 1. For details about the internal reference voltage, see **Section 34 Electrical Characteristics** for $T_A = -40$ to $+105^{\circ}\text{C}$.

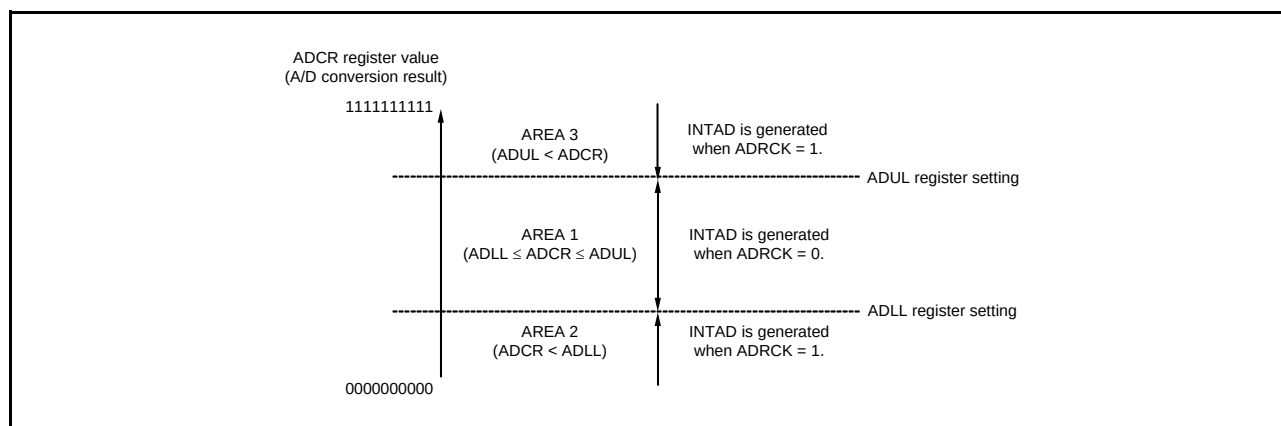
Note 2. Refer to "Transition time from STOP mode to SNOOZE mode" in **20.3.3 SNOOZE mode**.

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. When the internal reference voltage is selected (ADREFP1, ADREFP0 = 1, 0), the A/D converter reference voltage current (I_{ADREF}) indicated in 34.3.2 Characteristics of the supply current will be added.

Caution 3. When using AVREFP and AVREFM, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD is not generated, the result of A/D conversion is not stored in the ADCR and ADCRH registers.

12.3.6 10-bit A/D conversion result register (ADCR)

The ADCR register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH^{Note}.

The ADCR register can be read by a 16-bit memory manipulation instruction.

The value of this register following a reset is 0000H.

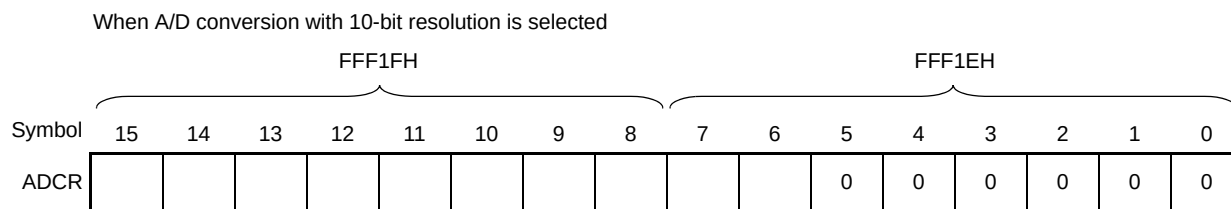
Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL and ADLL registers; see **Figure 12 - 9**), the result is not stored.

Figure 12 - 10 Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH

After reset: 0000H

R/W: R



Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the bits other than the higher 8 bits.

Caution 2. When the ADCR register is accessed in 16-bit units, and A/D conversion with 10-bit resolution is selected, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCR register.

12.3.7 8-bit A/D conversion result register (ADCRH)

The ADCRH register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL and ADLL registers; see **Figure 12 - 9**), the result is not stored.

Figure 12 - 11 Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to A/D converter mode register 0 (ADM0), the analog input channel specification register (ADS), port mode control A registers 0, 2, 12, and 14 (PMCA0, PMCA2, PMCA12, and PMCA14), and port mode control T registers 0, 2, and 14 (PMCT0, PMCT2, and PMCT14), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, PMCAxx, and PMCTxx registers. Using timing other than the above may cause an incorrect conversion result to be read.

12.3.8 Analog input channel specification register (ADS)

The ADS register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 12 Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

<Select mode (ADMD = 0)>

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFF pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	1	1	1	0	—	TSCAP voltage of the CTSU
1	0	0	0	0	0	—	Temperature sensor output voltage
1	0	0	0	0	1	—	Internal reference voltage ^{Note}
Other than the above						Setting prohibited	

Note For details about the internal reference voltage, see **Section 34 Electrical Characteristics** for $T_A = -40$ to $+105^\circ\text{C}$.

(Cautions are given on the next page.)

<Scan mode (ADMD = 1)>

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
Other than the above						Setting prohibited			

- Caution 1.** Be sure to clear bits 6 and 5 to 0.
- Caution 2.** Set the port that is specified as the analog input by a PMCAxx or PMCTxx register to the input mode by using port mode registers 0, 2, 12, or 14 (PM0, PM2, PM12, or PM14).
- Caution 3.** When specifying an input channel by the ADS register, do not select the pin that is specified as digital I/O by port mode control A register 0, 2, 12, or 14 (PMCA0, PMCA2, PMCA12, or PMCA14) or port mode control T register 0, 2, or 14 (PMCT0, PMCT2, or PMCT14).
- Caution 4.** Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).
- Caution 5.** If using AVREFP as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.
- Caution 6.** If using AVREFM as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.
- Caution 7.** If the ADISS bit is set to 1, the internal reference voltage cannot be used for the positive reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 12.7.4 Setup when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected (example for software trigger mode and one-shot conversion mode). For details about the internal reference voltage, see Section 34 Electrical Characteristics for TA = -40 to +105°C.
- Caution 8.** Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. When the ADISS bit is set to 1, the A/D converter reference voltage current (IADREF) indicated in 34.3.2 Characteristics of the supply current will be added.

12.3.9 Conversion result comparison upper limit setting register (ADUL)

The ADUL register is used to specify the setting for checking the upper limit of the A/D conversion results. The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range**).

The ADUL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Figure 12 - 13 Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

12.3.10 Conversion result comparison lower limit setting register (ADLL)

The ADLL register is used to specify the setting for checking the lower limit of the A/D conversion results. The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range**).

The ADLL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 14 Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When A/D conversion with 10-bit resolution is selected, the eight higher-order bits of the 10-bit A/D conversion result register (ADCR) are compared with the values in the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL registers must be greater than that of the ADLL register.

12.3.11 A/D test register (ADTES)

The ADTES register is used to select the positive or negative reference voltage for the converter, an analog input channel (ANlxx), TSCAP voltage of the CTSU, the temperature sensor output voltage, or the internal reference voltage^{Note} as the target for A/D conversion.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the negative reference voltage as the target for conversion.
- For full-scale measurement, select the positive reference voltage as the target for conversion.

The ADTES register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 12 - 15 Format of A/D Test Register (ADTES)

Address: F0013H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANlxx/TSCAP voltage of the CTSU/temperature sensor output voltage/internal reference voltage ^{Note} (This is specified using the analog input channel specification register (ADS).)
1	0	The negative reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The positive reference voltage (selected by the ADREFP1 and ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Note For details about the internal reference voltage, see **Section 34 Electrical Characteristics** for $T_A = -40$ to $+105^\circ\text{C}$.

12.3.12 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx), port mode control A registers (PMCAxx), and port mode control T registers (PMCTxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.7 Port mode control A registers (PMCAxx)**, and **4.3.8 Port mode control T registers (PMCTxx)**.

When using the ANI0 to ANI7 and ANI16 to ANI19 pins for analog input of the A/D converter, set the corresponding bits in the port mode register (PMxx) and port mode control A register (PMCAxx) to 1.

12.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

- <1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
- <2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.
- <3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to $(1/2)$ AVREF by the tap selector.
- <4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than $(1/2)$ AVREF, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than $(1/2)$ AVREF, the MSB bit is reset to 0.
- <5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
 - Bit 9 = 1: $(3/4)$ AVREF
 - Bit 9 = 0: $(1/4)$ AVREFThe voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.
 - Sampled voltage $\geq$ Voltage tap: Bit 8 = 1
 - Sampled voltage $<$ Voltage tap: Bit 8 = 0
- <6> Comparison is continued in this way up to bit 0 of the SAR register.
- <7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result registers (ADCR, ADCRH) and then latched^{Note 1}.
At the same time, the A/D conversion end interrupt request signal (INTAD) can also be generated^{Note 1}.
- <8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0^{Note 2}.
To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 12 - 9 ADRCK Bit Interrupt Signal Generation Range**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

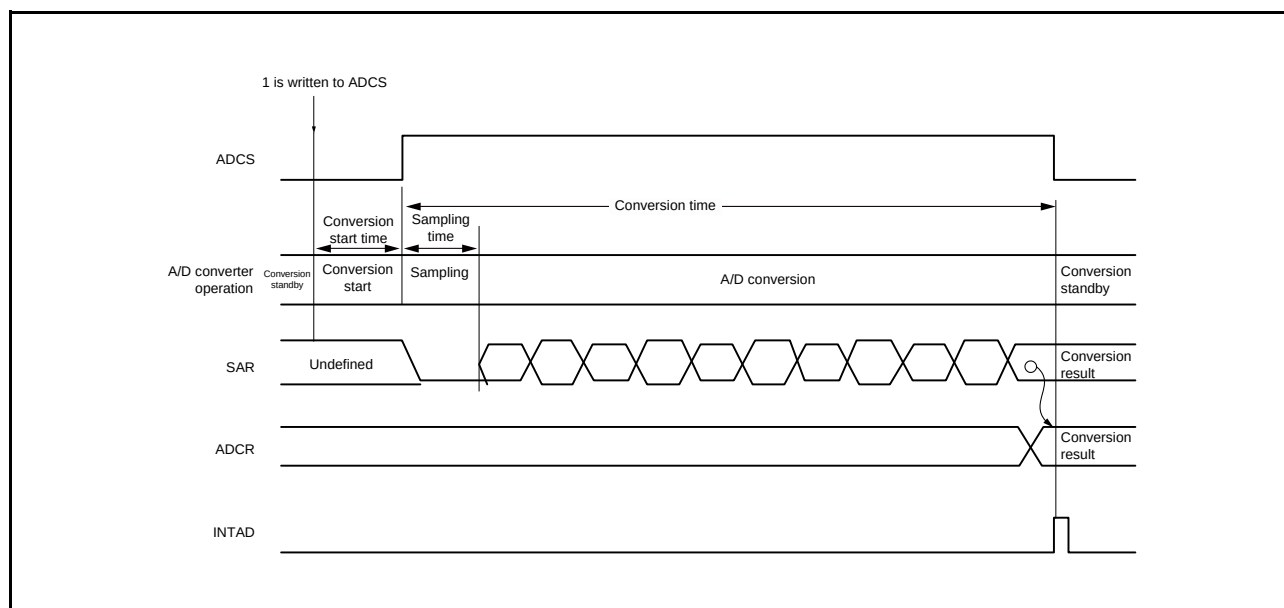
Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AVREF: The positive reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage, and VDD.

For details about the internal reference voltage, see **Section 34 Electrical Characteristics for Ta = -40 to +105°C**.

Figure 12 - 16 Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

The values of the A/D conversion result registers (ADCR and ADCRH) following a reset are 0000H and 00H, respectively.

12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7, ANI16 to ANI19) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\begin{aligned} \text{SAR} &= \text{INT} \left(\frac{V_{\text{AIN}}}{V_{\text{REF}}} \times 1024 + 0.5 \right) \\ \text{ADCR} &= \text{SAR} \times 64 \end{aligned}$$

or

$$(\text{ADCR}/64 - 0.5) \times \frac{V_{\text{REF}}}{1024} \leq V_{\text{AIN}} < (\text{ADCR}/64 + 0.5) \times \frac{V_{\text{REF}}}{1024}$$

where, INT(): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

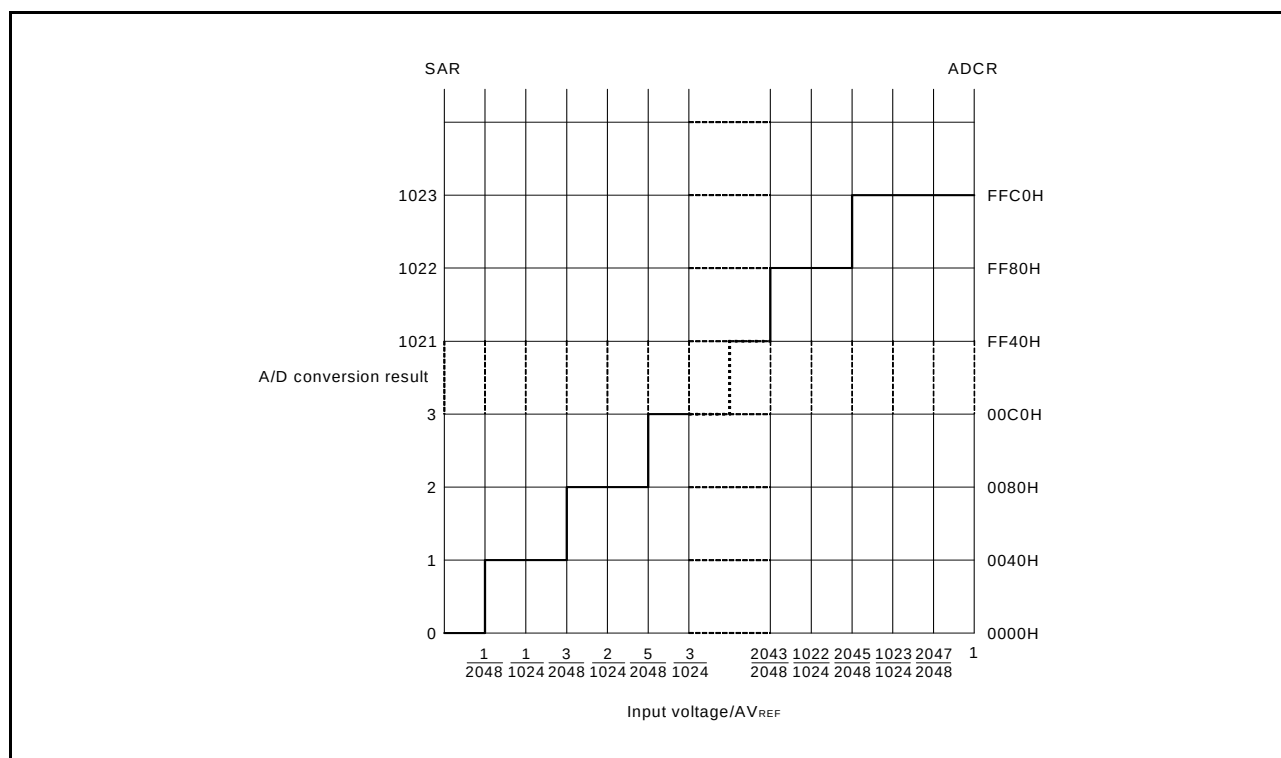
V_{REF}: V_{REF} pin voltage

ADCR: 10-bit A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12 - 17 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12 - 17 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark V_{REF}: The positive reference voltage of the A/D converter. This can be selected from V_{REFP}, the internal reference voltage^{Note}, and V_{DD}.

Note For details about the internal reference voltage, see **Section 34 Electrical Characteristics** for T_A = -40 to +105°C.

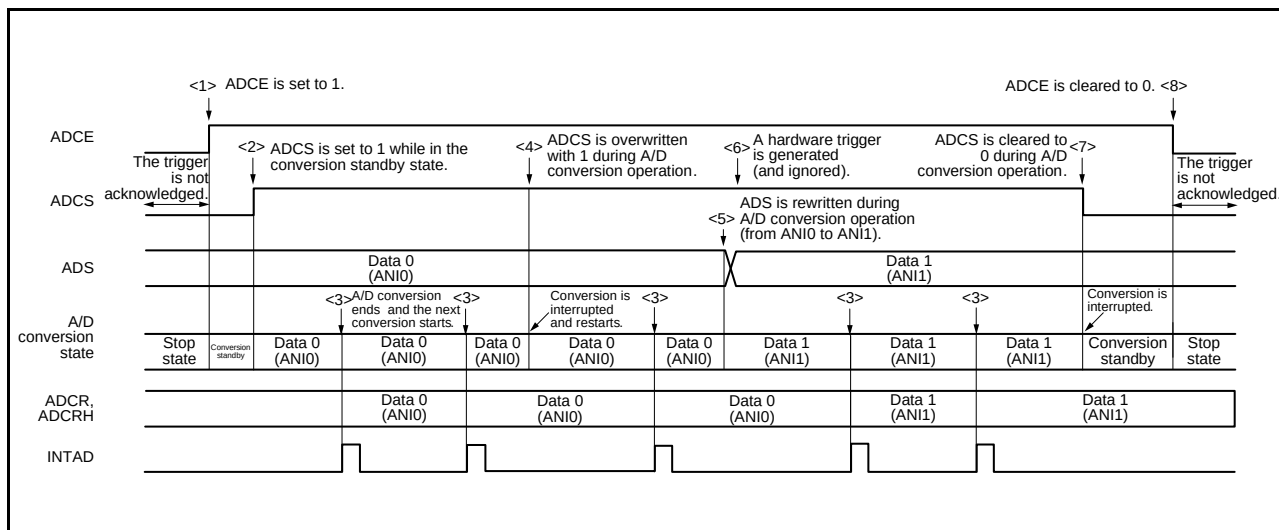
12.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **12.7 A/D Converter Setup Flowchart**.

12.6.1 Software trigger mode (select mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.
- <4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.
- <8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

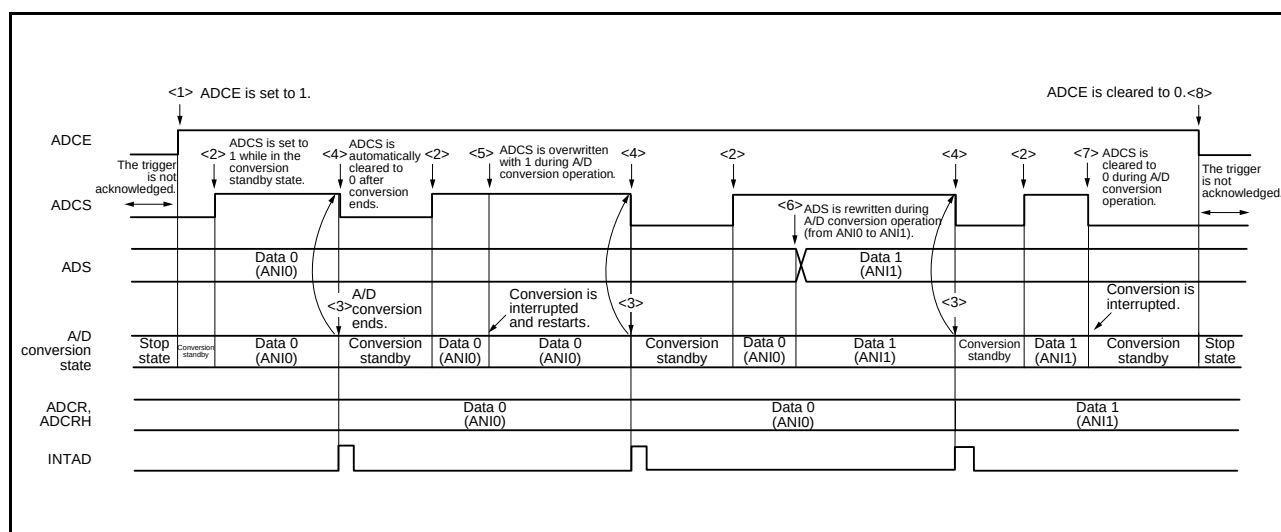
Figure 12 - 18 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.2 Software trigger mode (select mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- <4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the standby state.
- <5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.
- <8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

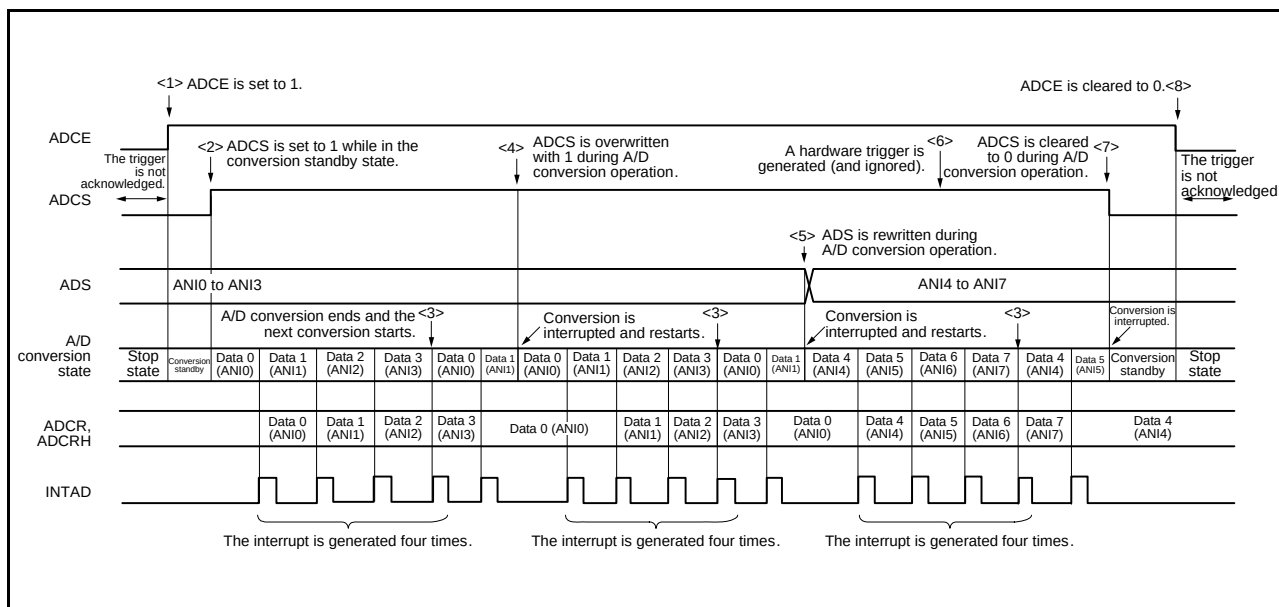
Figure 12 - 19 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.3 Software trigger mode (scan mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts (until all four channels are finished).
- <4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.
- <8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

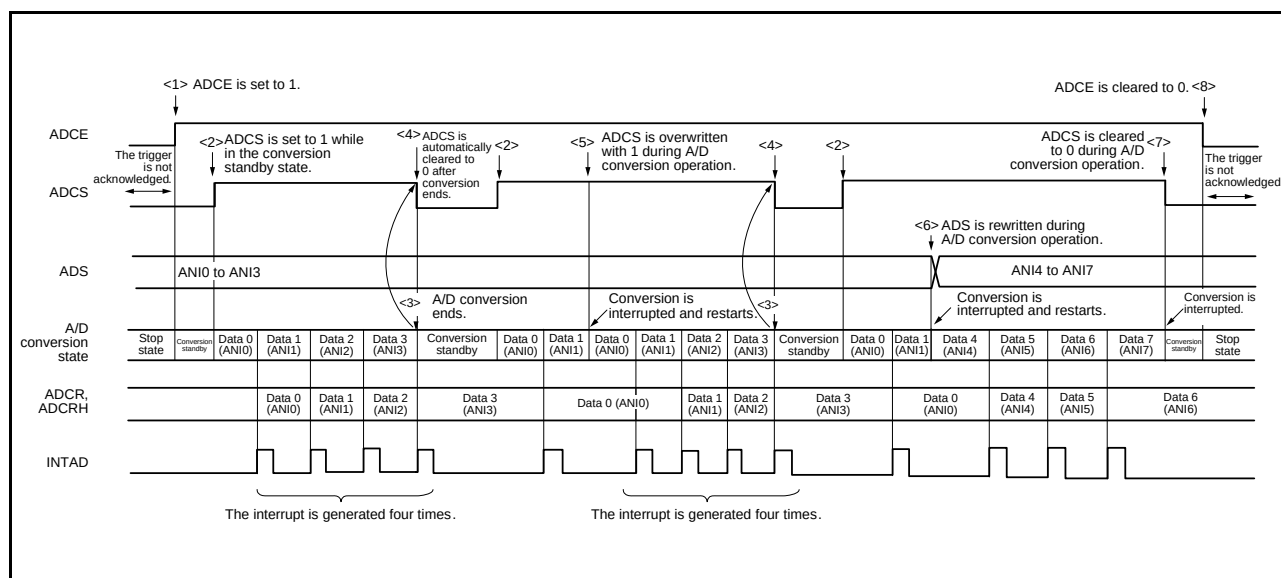
Figure 12 - 20 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.4 Software trigger mode (scan mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.
- <4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the standby state.
- <5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state.
- <8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

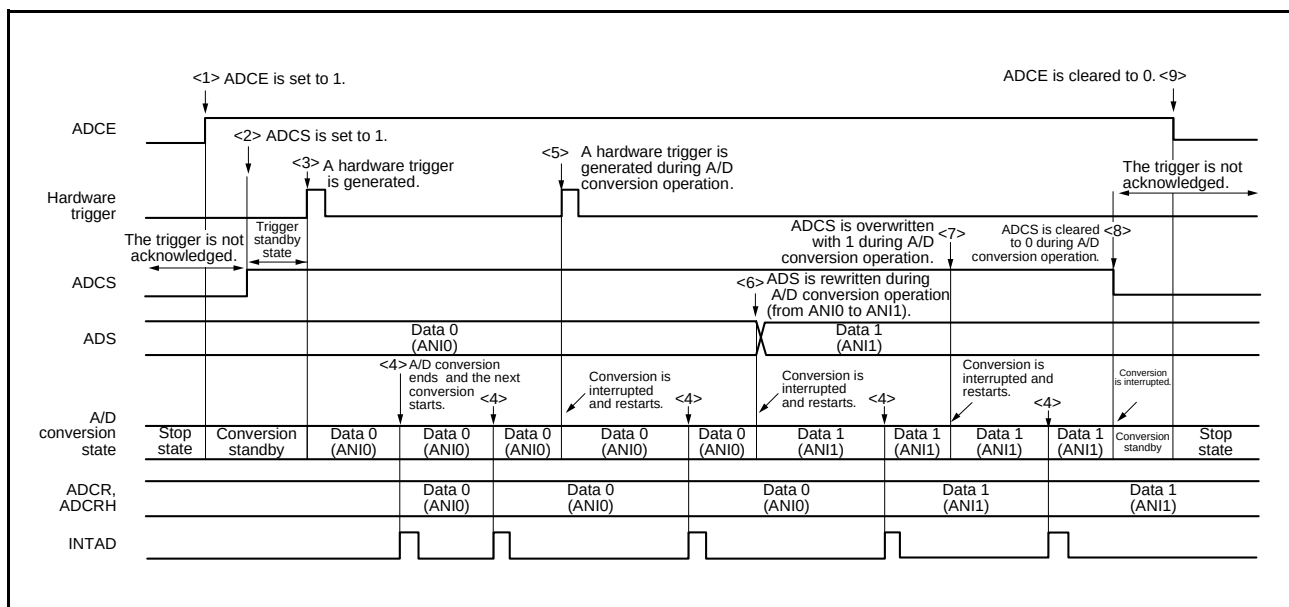
Figure 12 - 21 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.
- <3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).
- <4> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.
- <5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state. However, the A/D converter does not stop in this state.
- <9> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

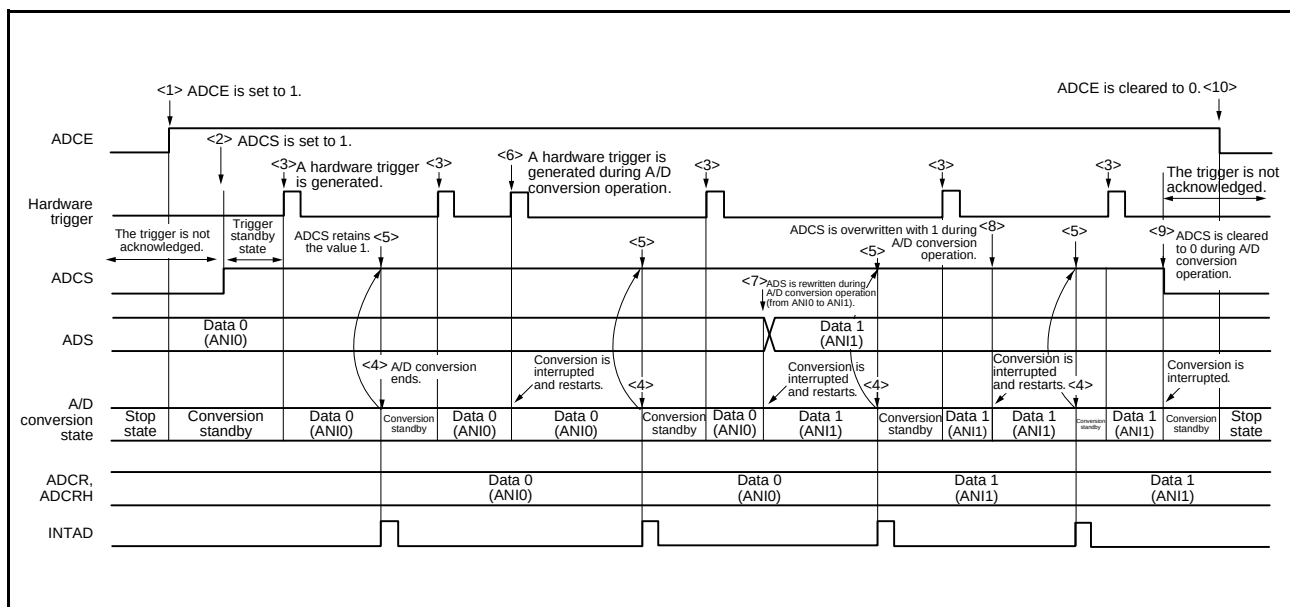
Figure 12 - 22 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.
- <3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).
- <4> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- <5> After A/D conversion ends, the ADCS bit remains set to 1, and the A/D converter enters the standby state.
- <6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state. However, the A/D converter does not stop in this state.
- <10> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

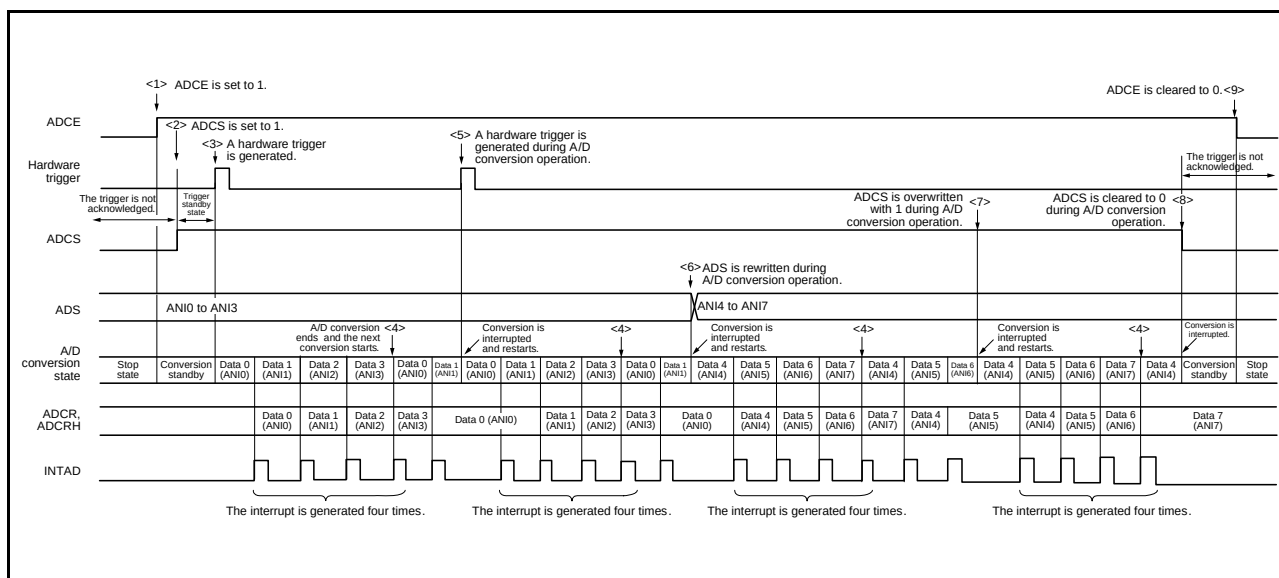
Figure 12 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.
- <3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.
- <5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state. However, the A/D converter does not stop in this state.
- <9> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

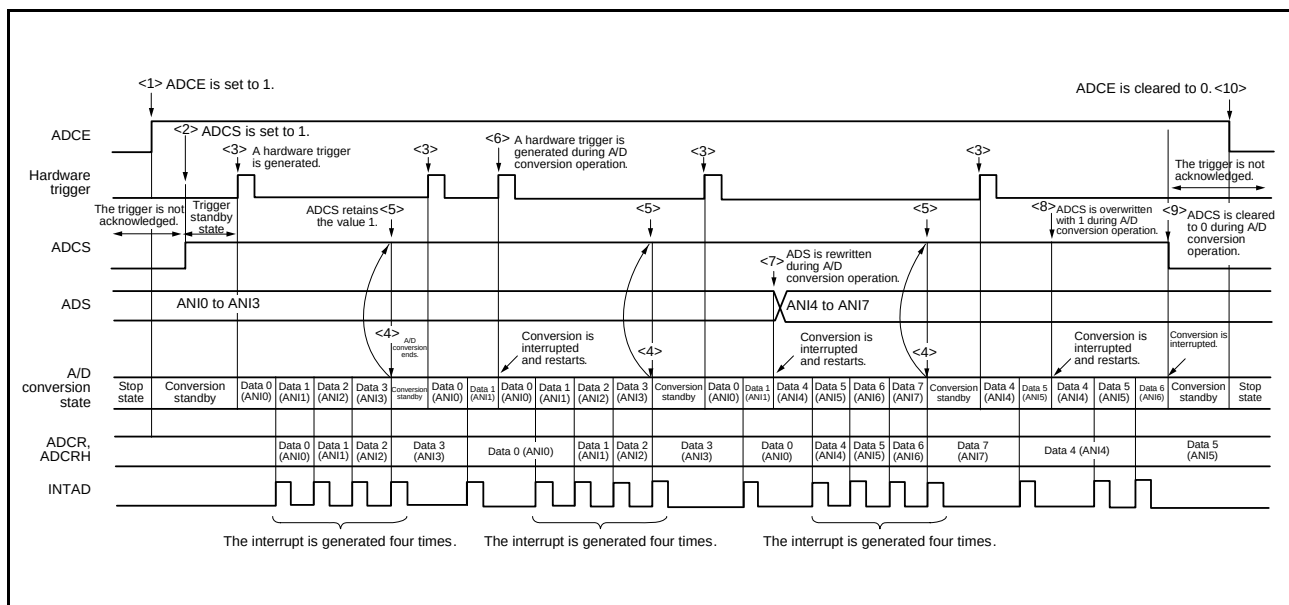
Figure 12 - 24 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> After the software counts up to the stabilization wait time (1 μ s), the ADCS bit of the ADM0 register is set to 1 to place the A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.
- <3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels ends.
- <5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the A/D converter enters the standby state.
- <6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the A/D converter enters the standby state. However, the A/D converter does not stop in this state.
- <10> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

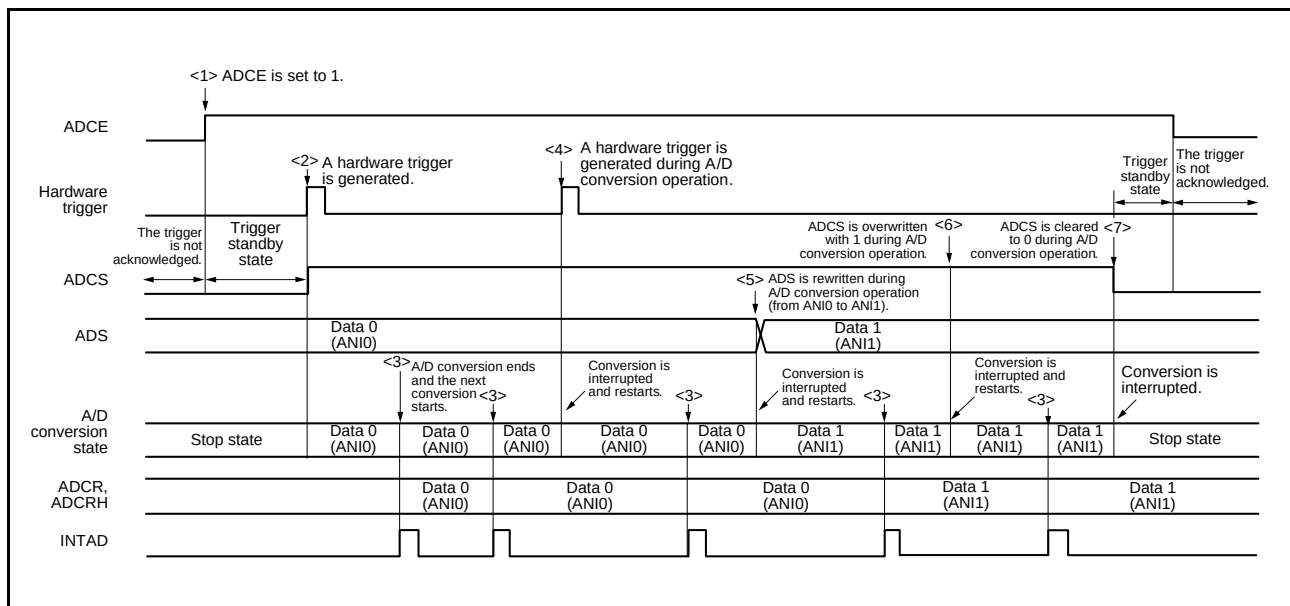
Figure 12 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the hardware trigger standby state.
- <2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)
- <4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

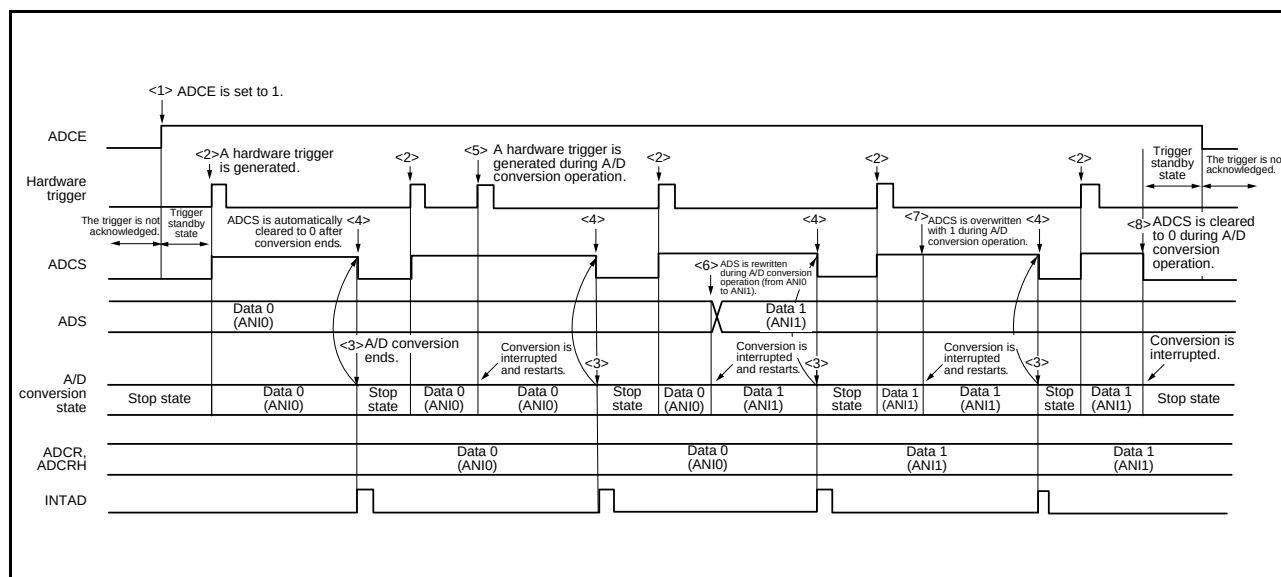
Figure 12 - 26 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the hardware trigger standby state.
- <2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.
- <4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.
- <5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.
- <8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 27 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing

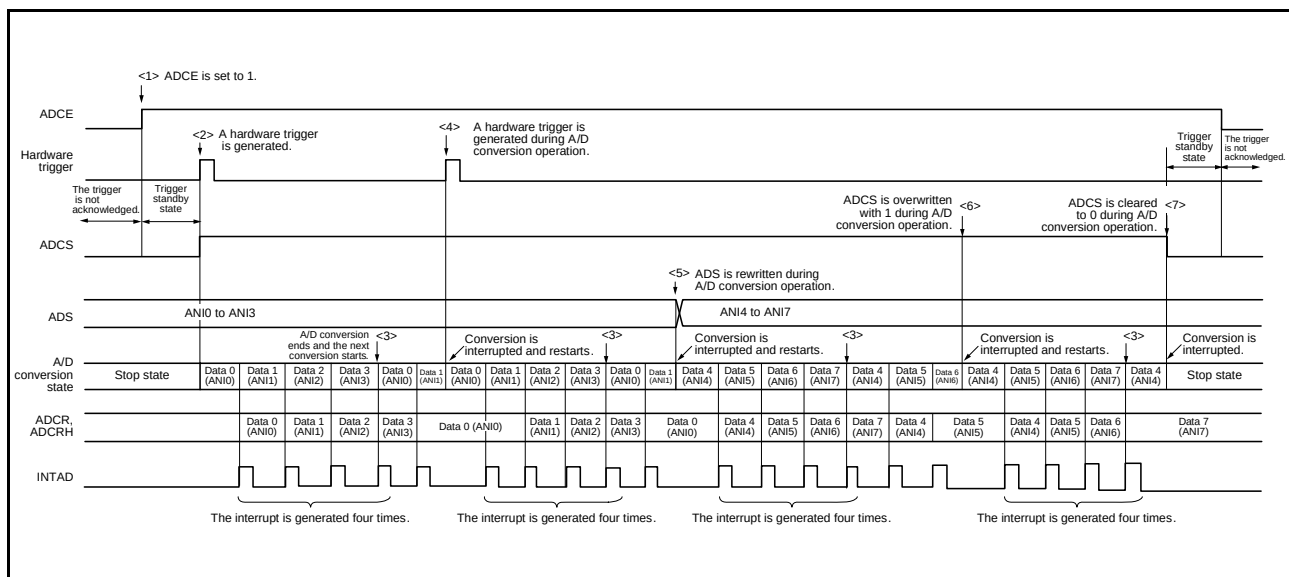


Caution In hardware trigger wait mode (select mode, one-shot conversion mode), setting the ADISS bit to 1 (selecting the temperature sensor output voltage and internal reference voltage as input sources) is not possible.

12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.
- <4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

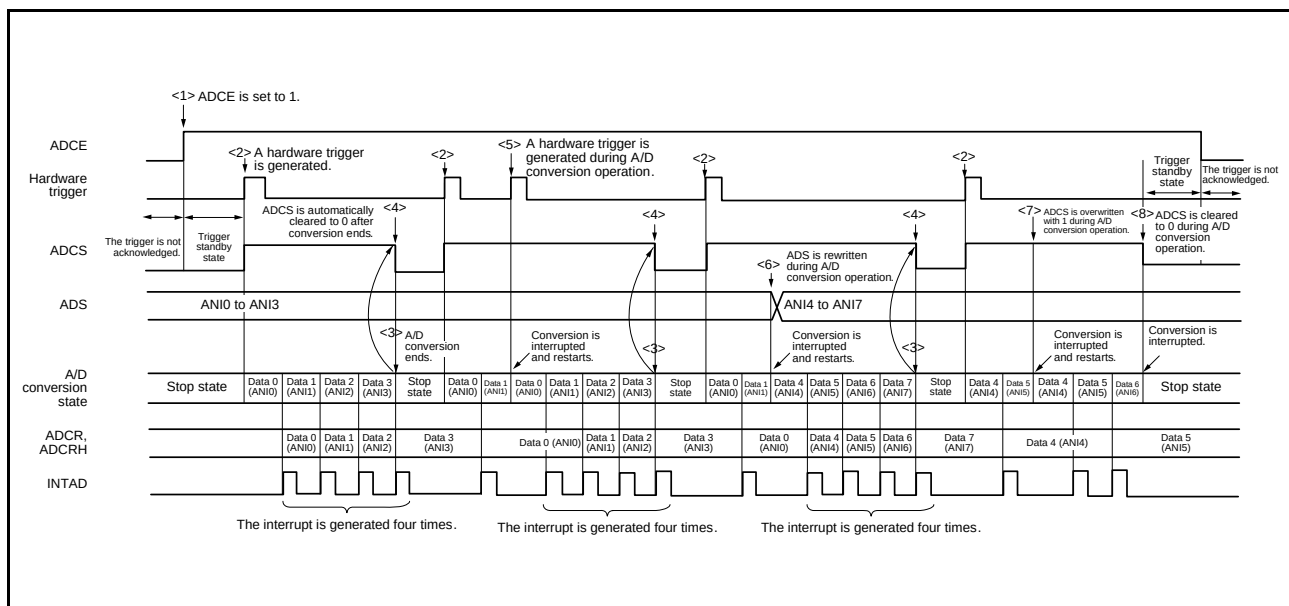
Figure 12 - 28 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

- <1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 in the stop state places the A/D converter in the standby state.
- <2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.
- <3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result registers (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.
- <4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop state.
- <5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.
- <6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.
- <7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.
- <8> When ADCS is cleared to 0 during conversion operation, the hardware trigger standby state is entered, and the A/D converter is placed in the stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 12 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

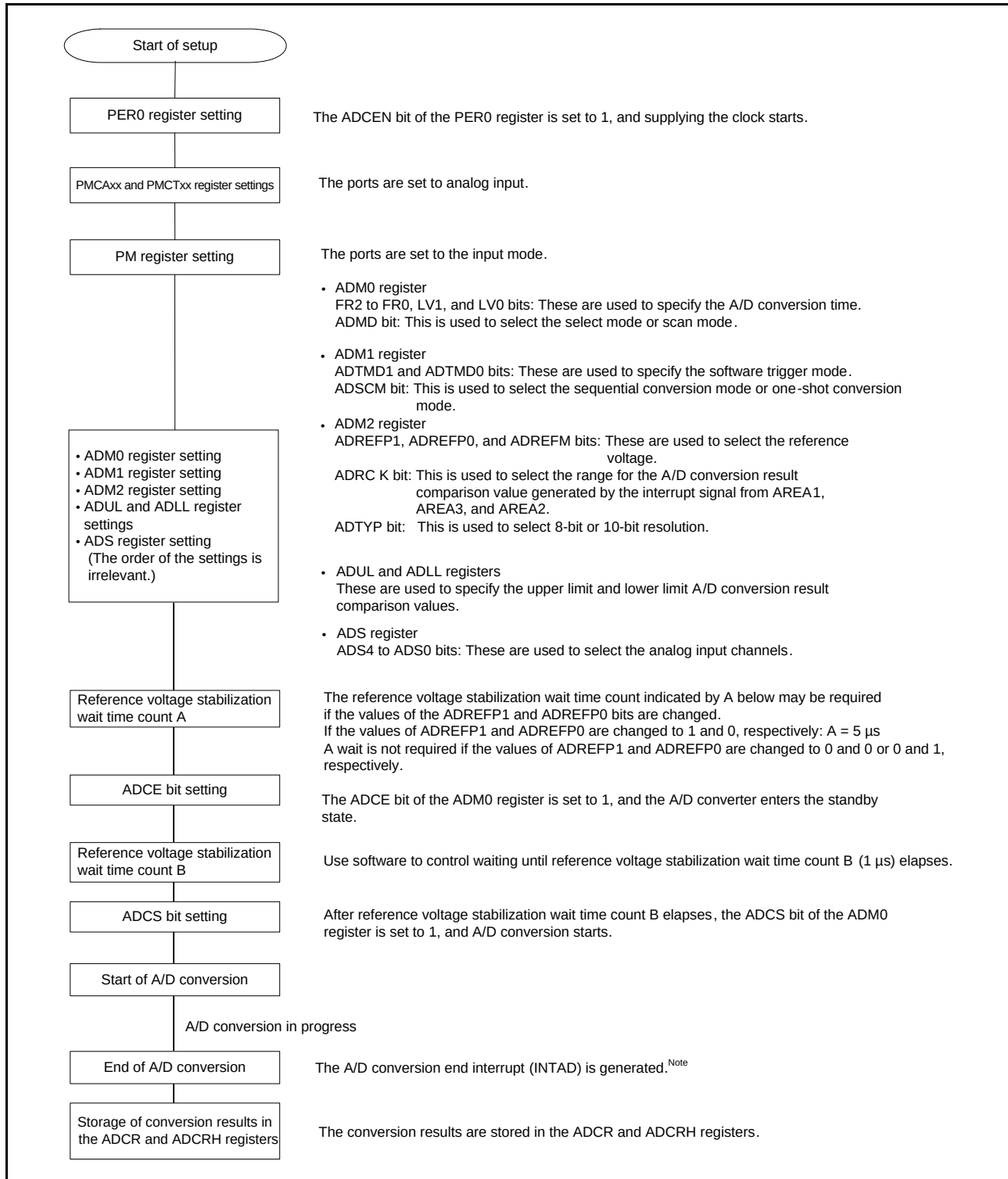


12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

12.7.1 Setting up software trigger mode

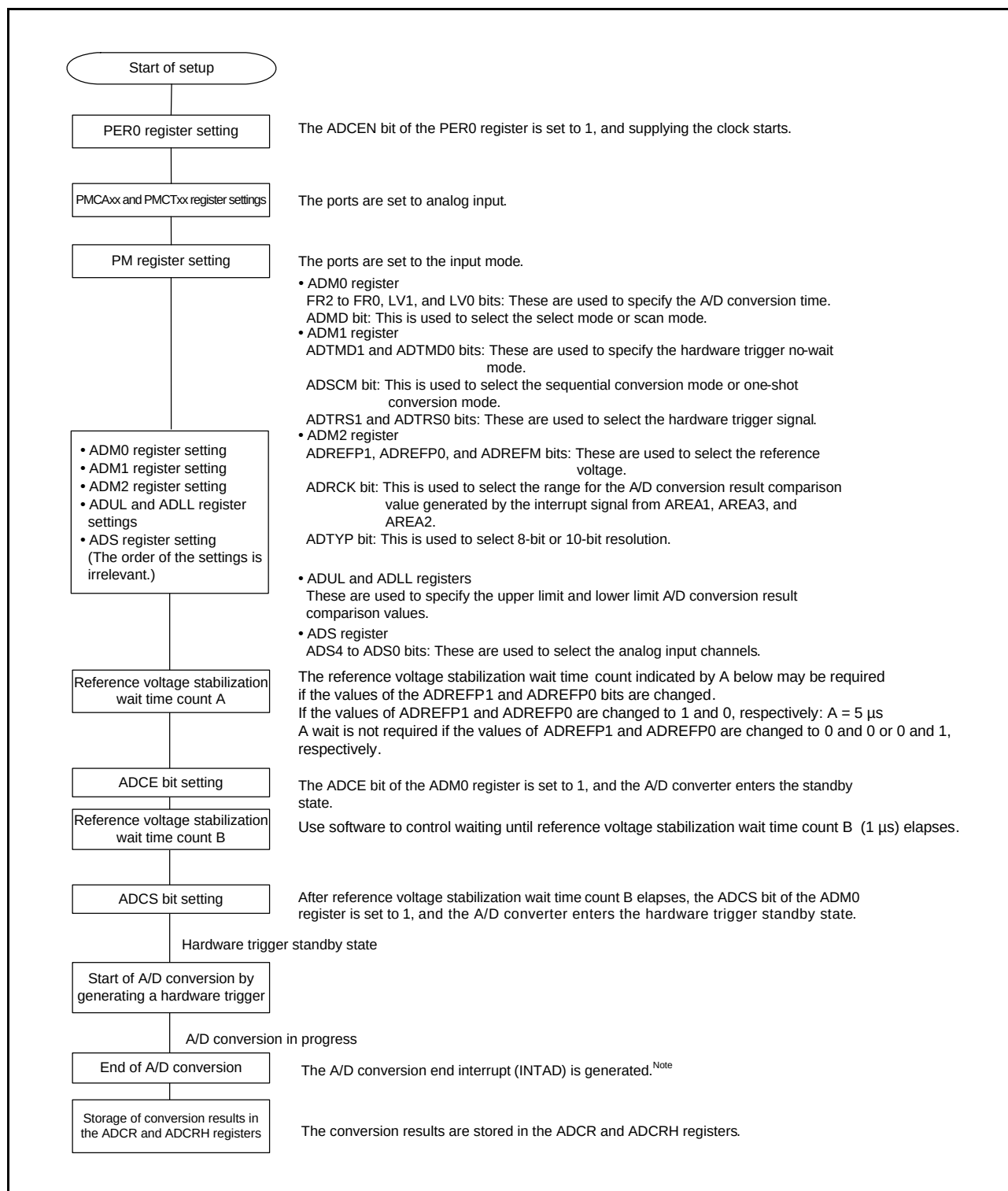
Figure 12 - 30 Setting up Software Trigger Mode



Note Depending on the settings of the ADRCCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

12.7.2 Setting up hardware trigger no-wait mode

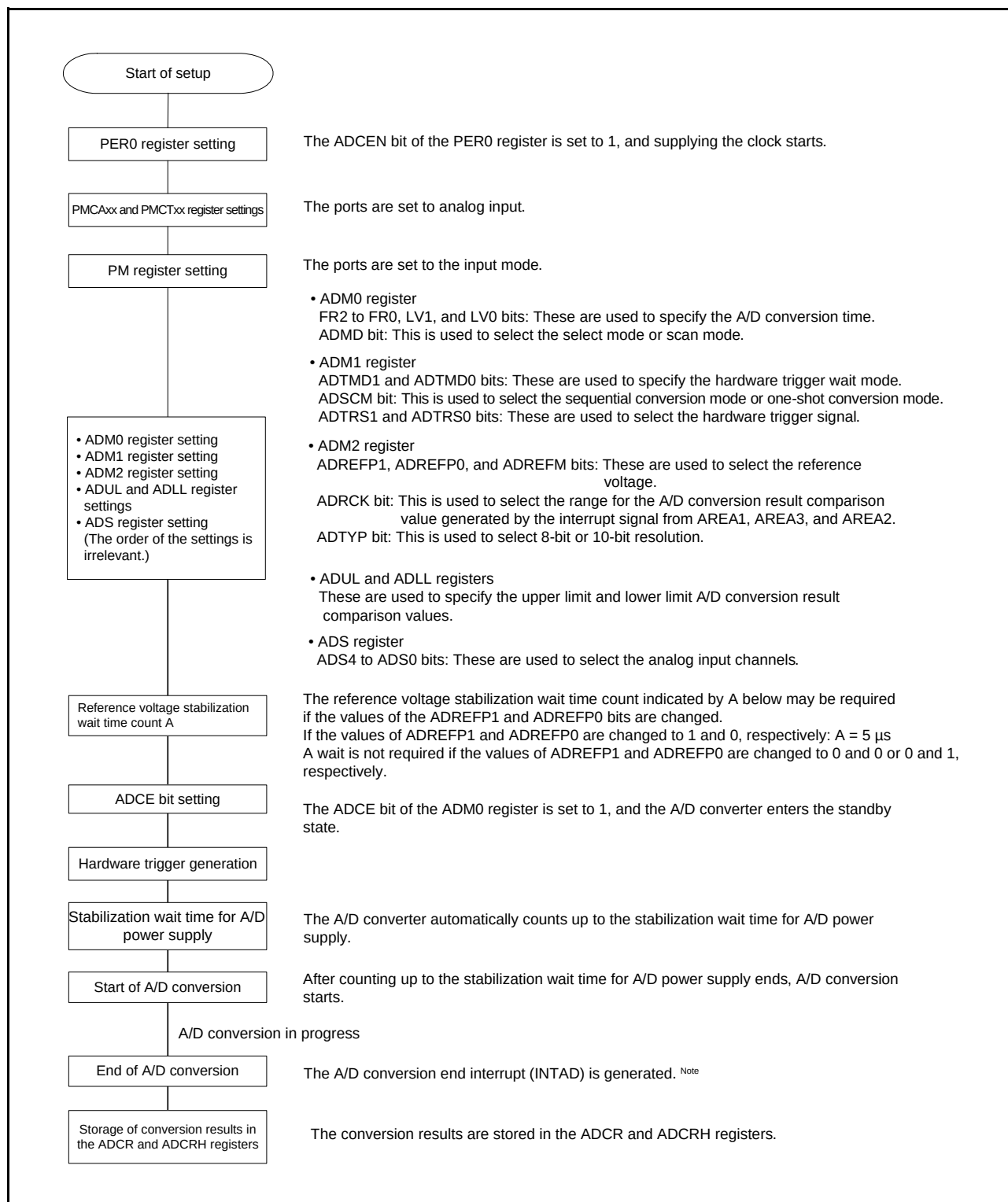
Figure 12 - 31 Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

12.7.3 Setting up hardware trigger wait mode

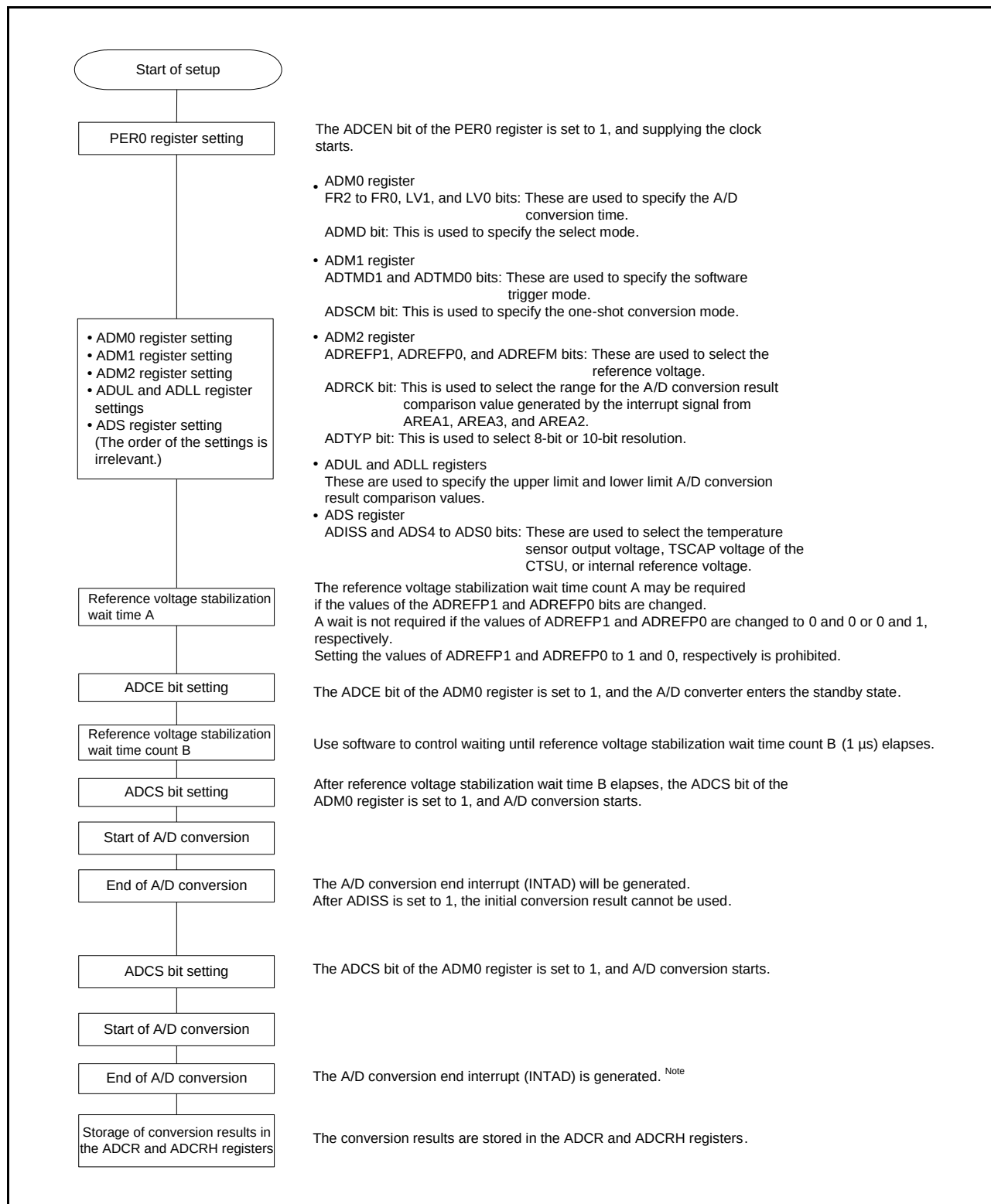
Figure 12 - 32 Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

12.7.4 Setup when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected (example for software trigger mode and one-shot conversion mode)

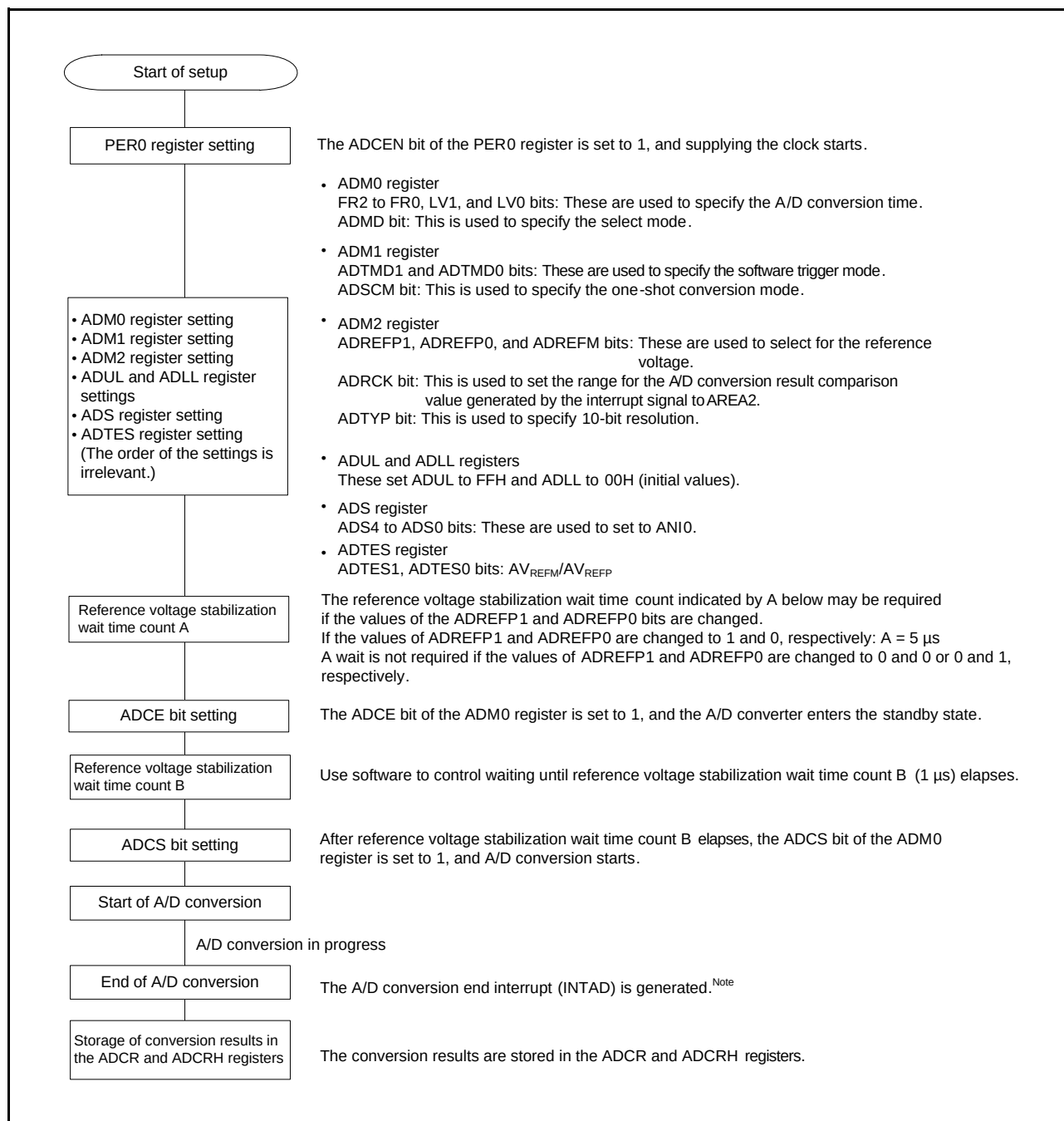
Figure 12 - 33 Setup When Temperature Sensor Output Voltage, Internal Reference Voltage, or TSCAP Voltage of the CTSU is Selected



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

12.7.5 Setting up test mode

Figure 12 - 34 Setting up Test Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

Caution For the procedure for testing the A/D converter, see 24.3.10 Testing of the A/D converter.

12.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode function, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.

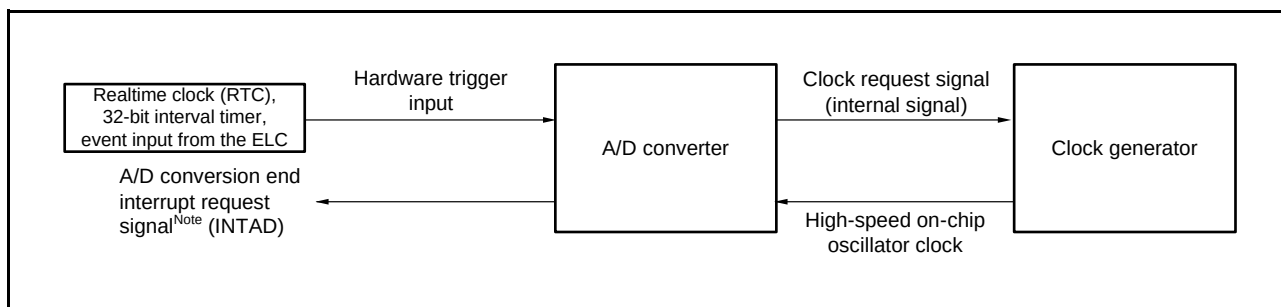
In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for fCLK.

Figure 12 - 35 Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. For details about these settings, see **Figure 12 - 38 Flowchart for Settings in SNOOZE Mode (Hardware Trigger)**. Just before move to STOP mode, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the A/D converter automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated^{Note}.

Note Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL and ADLL registers), there is a possibility of no interrupt signal being generated.

Caution Select the hardware trigger signal from among the realtime clock interrupt signal (INTRTC), 32-bit interval timer interrupt signal (ELCITL0), and event input from the ELC.

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD) is generated.

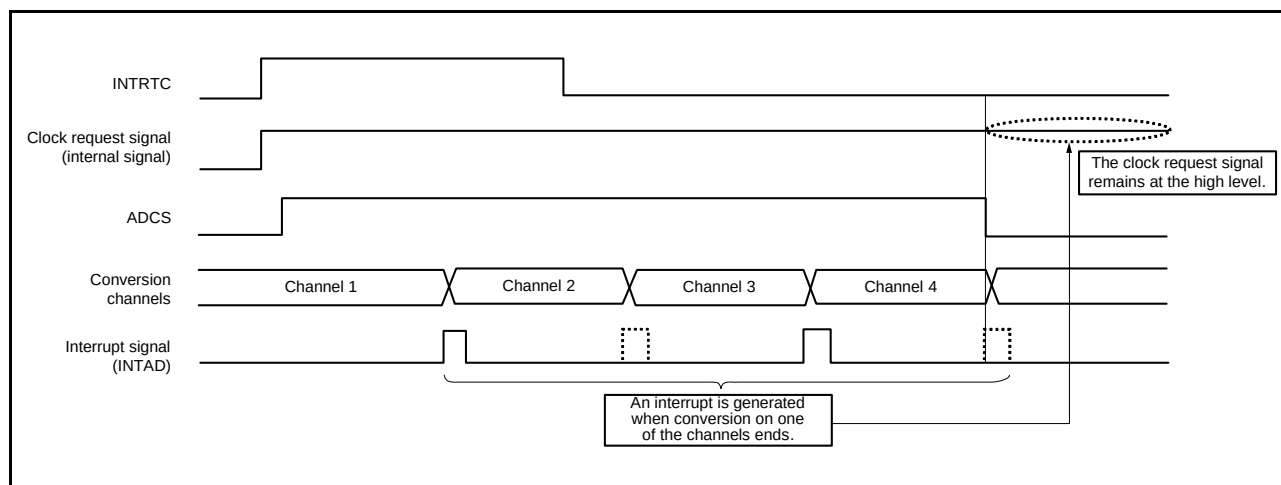
- While in the select mode

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

If even one value of the A/D conversion results of the four channels falls within the range specified by the A/D conversion result comparison function, and A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 12 - 36 Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode

If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

If the A/D conversion result values of the four channels do not fall within the range specified by the A/D conversion result comparison function even once, and the A/D conversion end interrupt request signal (INTAD) is not generated, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 12 - 37 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

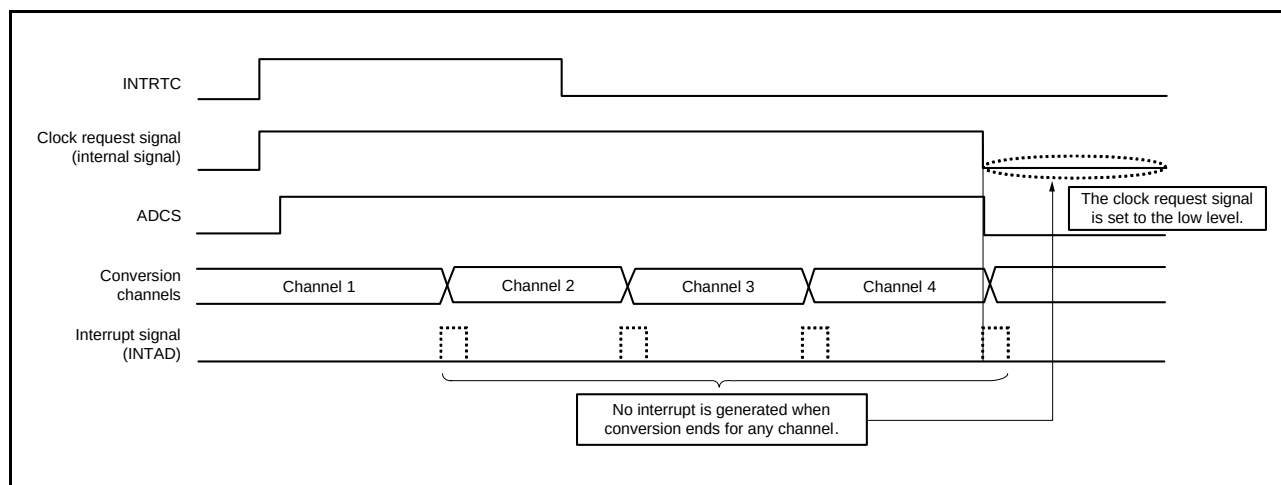
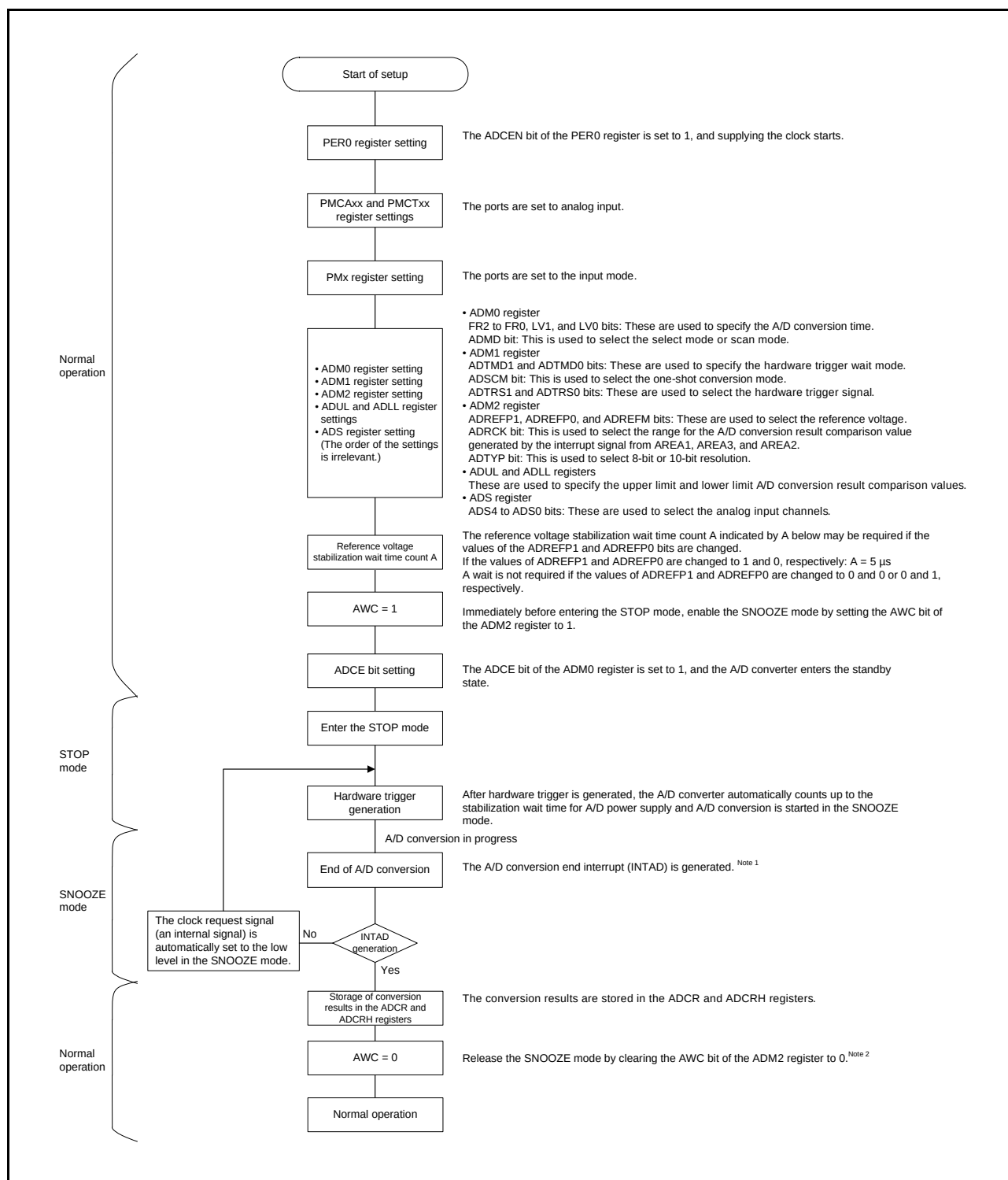


Figure 12 - 38 Flowchart for Settings in SNOOZE Mode (Hardware Trigger)



12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$1\text{LSB} = 1/2^{10} = 1/1024 \\ \approx 0.098\%\text{FSR}$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a $\pm 1/2\text{LSB}$ error naturally occurs. In an A/D converter, an analog input voltage in a range of $\pm 1/2\text{LSB}$ is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12 - 39 Overall Error

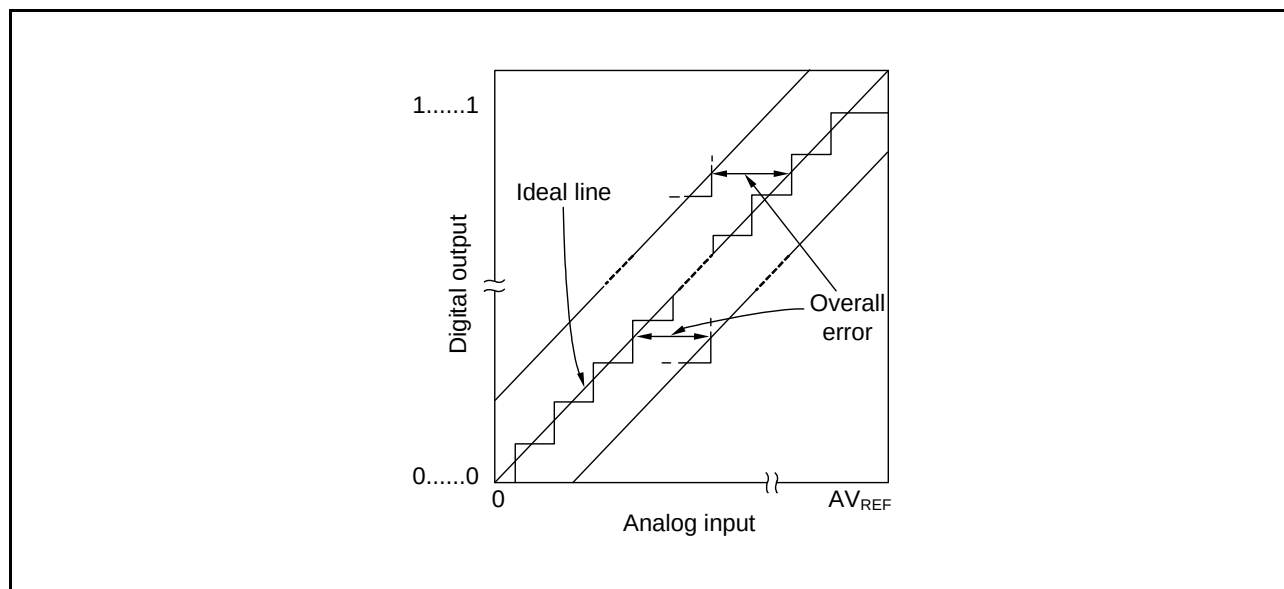
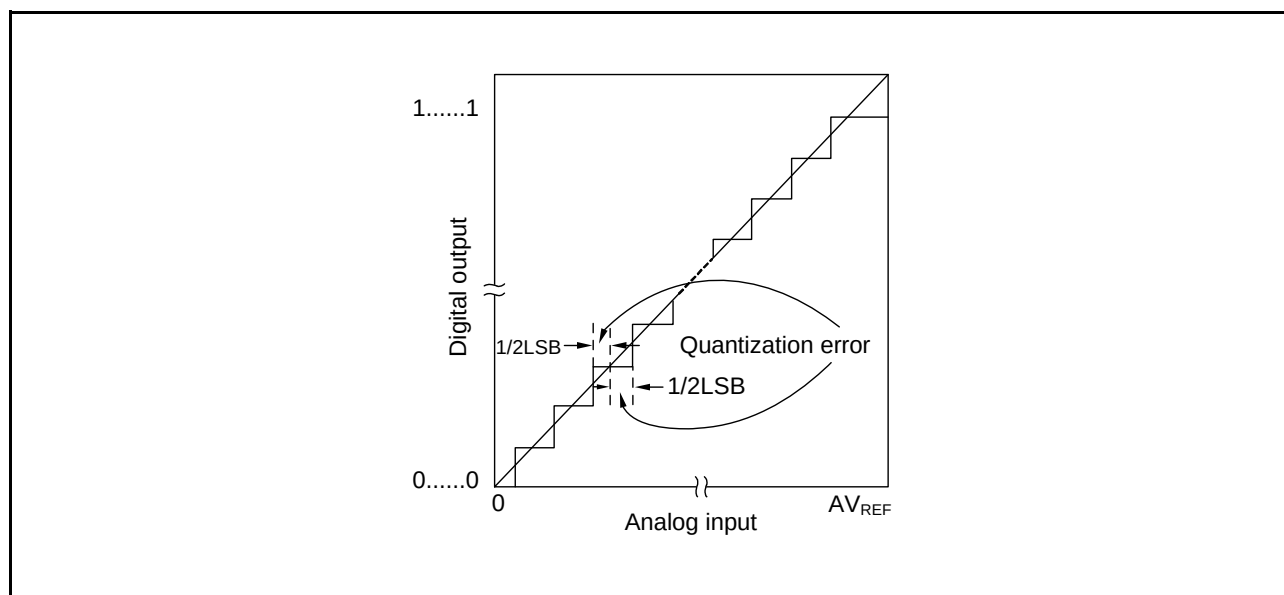


Figure 12 - 40 Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value ($1/2\text{LSB}$) when the digital output changes from $0\ldots\ldots000$ to $0\ldots\ldots001$.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value ($3/2\text{LSB}$) when the digital output changes from $0\ldots\ldots001$ to $0\ldots\ldots010$.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale - $3/2\text{LSB}$) when the digital output changes from $1\ldots\ldots110$ to $1\ldots\ldots111$.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 12 - 41 Zero-Scale Error

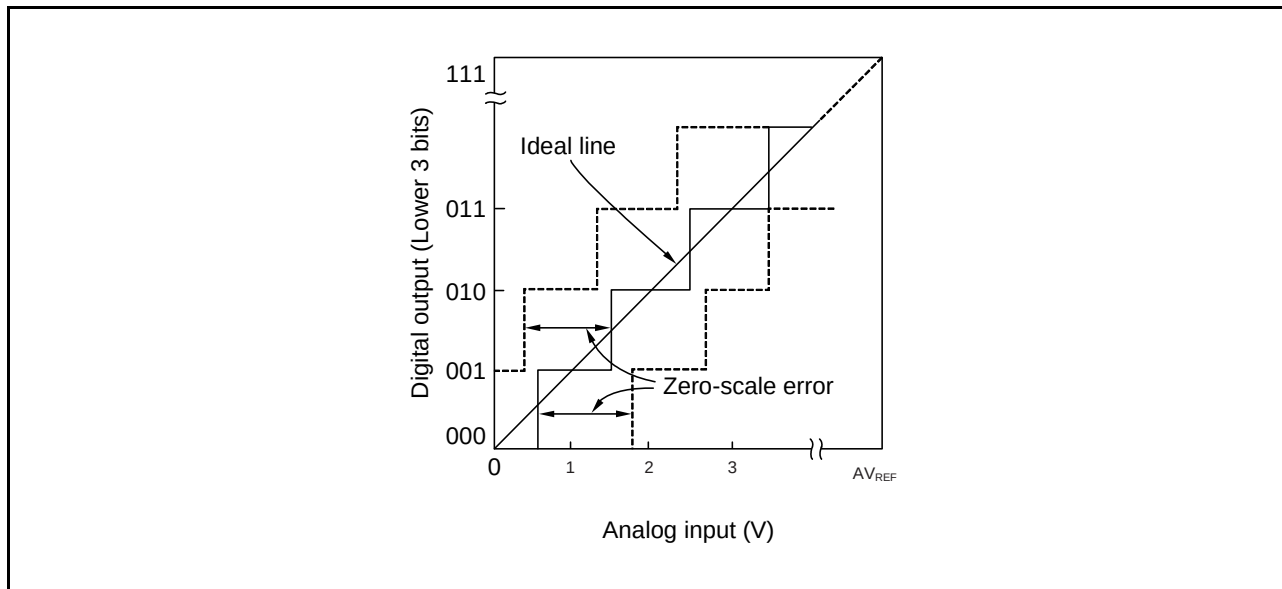


Figure 12 - 42 Full-Scale Error

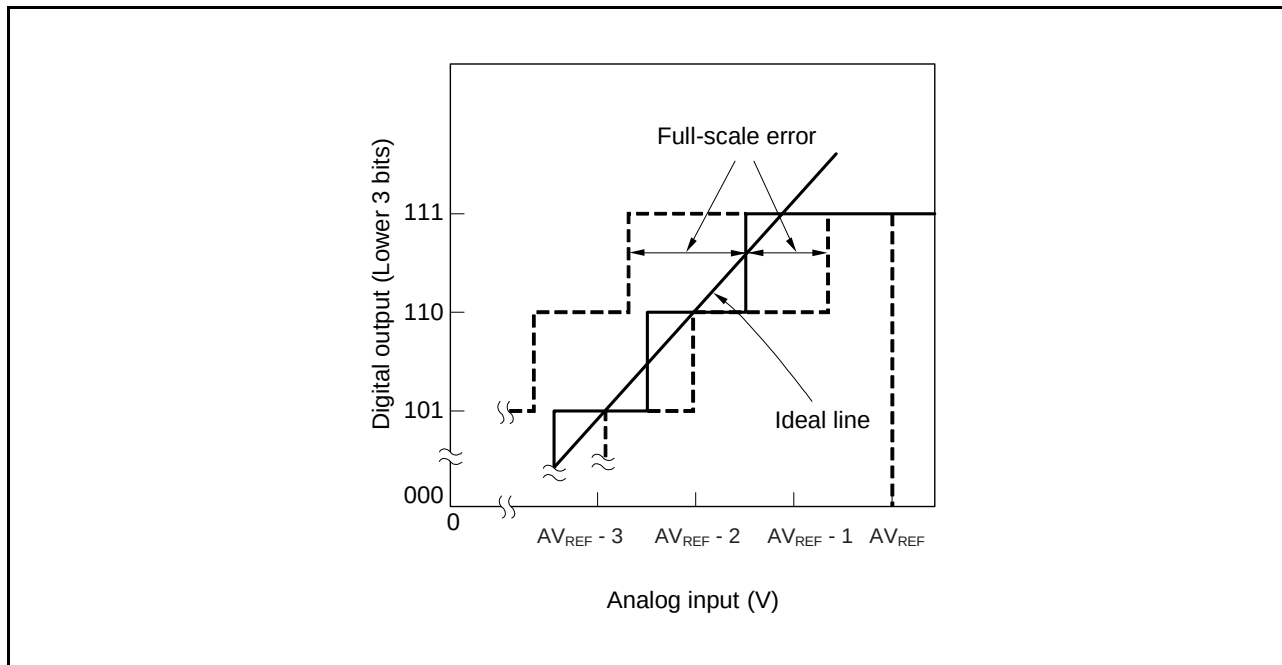


Figure 12 - 43 Integral Linearity Error

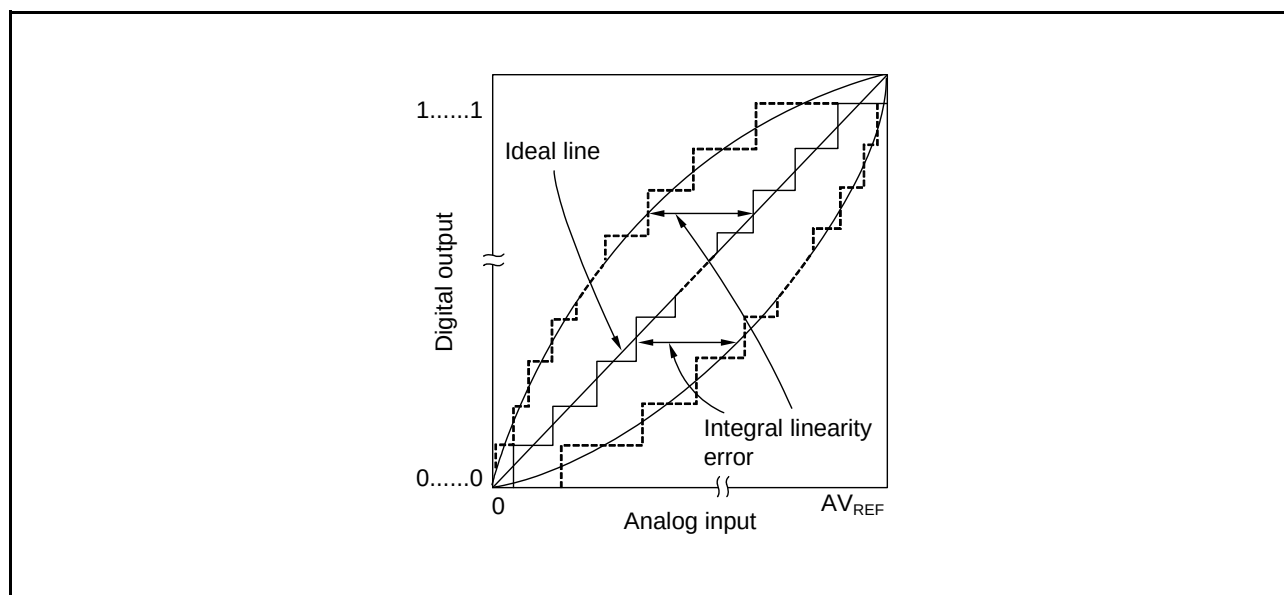
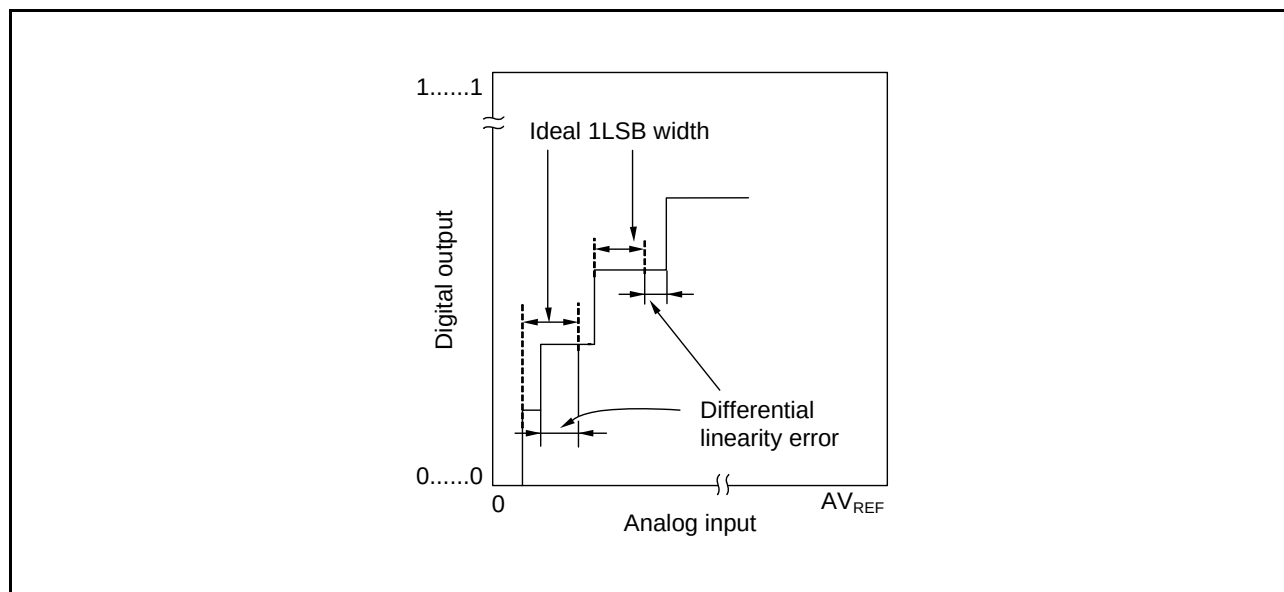


Figure 12 - 44 Differential Linearity Error

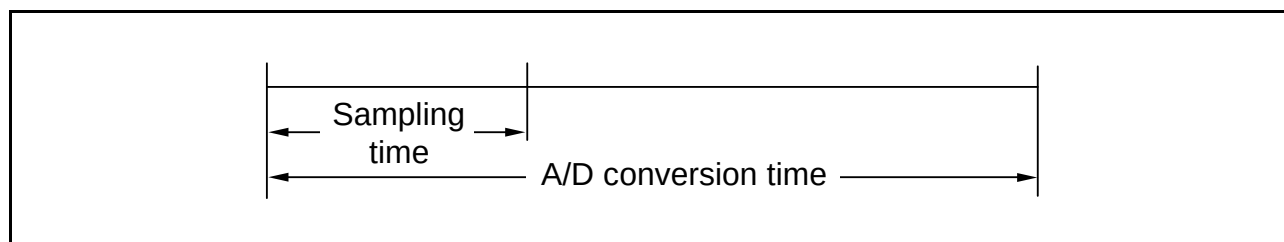


(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.
The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.10 Points for Caution when the A/D Converter is to be Used

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby state, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI7 and ANI16 to ANI19 pins

Observe the rated range of the ANI0 to ANI7 and ANI16 to ANI19 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or a voltage lower than V_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage is selected as the positive reference voltage of the A/D converter, do not input a voltage equal to or higher than the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a voltage equal to or higher than the internal reference voltage is input to a pin not selected by the ADS register.

Caution For details about the internal reference voltage, see Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$.

(3) Conflicting operations

<1> Conflict between the write access to the A/D conversion result register (ADCR or ADCRH) upon the end of conversion and the read access to the ADCR or ADCRH register with an instruction

Reading from the ADCR or ADCRH register has the higher priority. After the read operation, the new conversion result is written to the ADCR or ADCRH register.

<2> Conflict between the write access to the ADCR or ADCRH register for storage of the result of A/D conversion upon the end of conversion, the write access to A/D converter mode register 0 (ADM0) with an instruction, and the write access to the analog input channel specification register (ADS)

Writing to the ADM0 and ADS registers has the higher priority. Writing to the ADCR and ADCRH registers is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP} , V_{DD} , ANI0 to ANI7, and ANI16 to ANI19 pins.

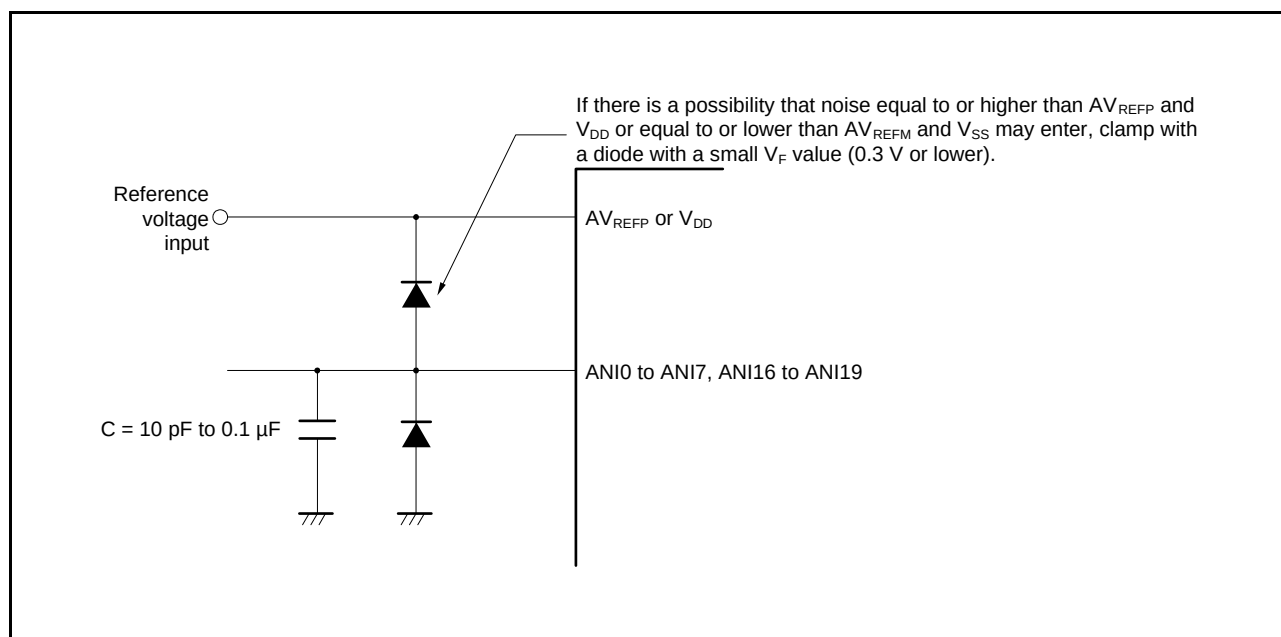
<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about $0.01\ \mu\text{F}$) via the shortest possible run of relatively thick wiring to the V_{DD} and AV_{REFP} pins.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting an external capacitor as shown in **Figure 12 - 45** is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 12 - 45 Analog Input Pin Connection



(5) Analog input (ANLxx) pins

- <1> The analog input pins (ANI0 to ANI7 and ANI16 to ANI19) are also used as input port pins (P00, P01, P20 to P27, P120, and P147). Do not change the output values for the port-pin functions P00, P01, P20 to P27, P120, and P147 while A/D conversion of the signals on the ANI0 to ANI7 or ANI16 to ANI19 pins is selected and conversion is in progress, since doing so may lower the precision of the results of conversion.
- <2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output. Be sure to avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.

(6) Input impedance of analog input (ANLxx) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI7 and ANI16 to ANI19 to which the source is connected (see **Figure 12 - 45**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

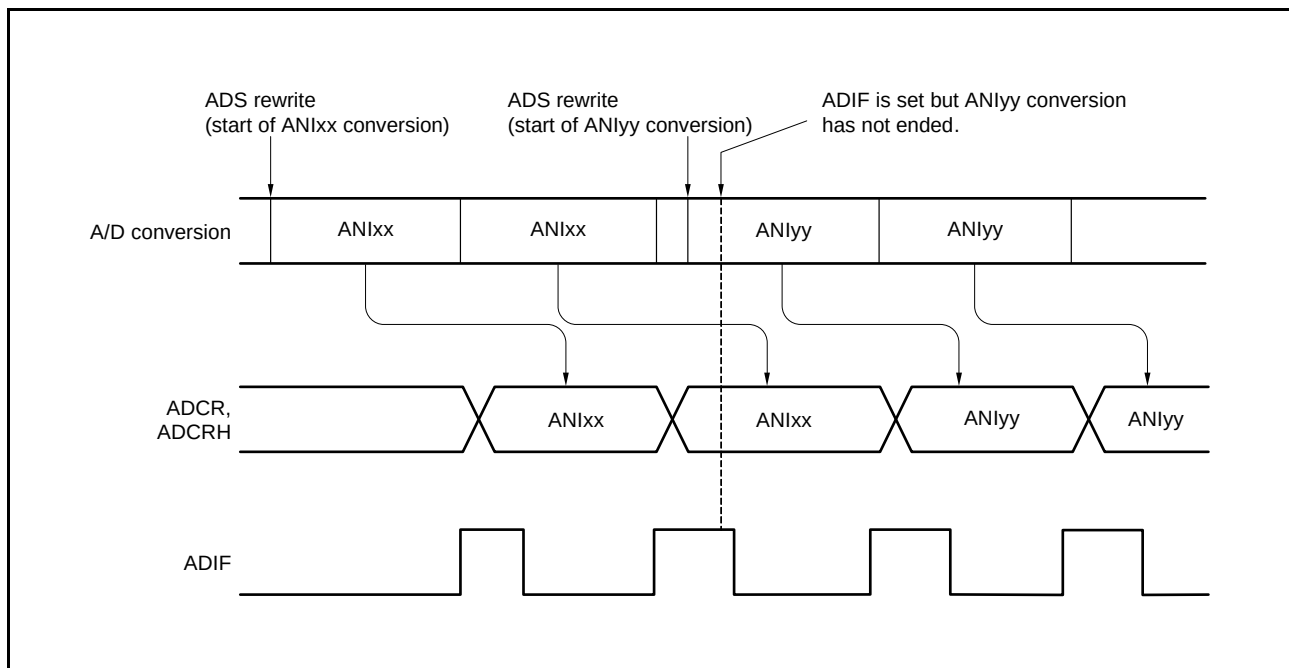
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 12 - 46 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μ s after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), port mode control A register (PMCAxx), or port mode control T register (PMCTxx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, PMCAxx, or PMCTxx register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12 - 47 Internal Equivalent Circuit of ANIxx Pin

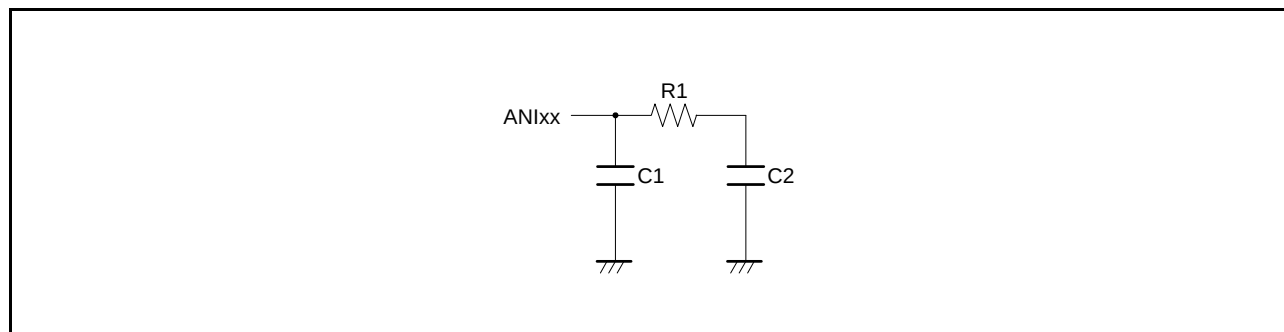


Table 12 - 4 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AVREFP, VDD	ANIxx Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ VDD ≤ 5.5 V	ANI0 to ANI7	14	8	2.5
	ANI16 to ANI19	18	8	7
2.7 V ≤ VDD ≤ 3.6 V	ANI0 to ANI7	39	8	2.5
	ANI16 to ANI19	53	8	7
1.8 V ≤ VDD ≤ 2.7 V	ANI0 to ANI7	231	8	2.5
	ANI16 to ANI19	321	8	7
1.6 V ≤ VDD < 2.7 V	ANI0 to ANI7	632	8	2.5
	ANI16 to ANI19	902	8	7

Remark The resistance and capacitance values shown in **Table 12 - 4** are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AVREFP and VDD voltages stabilize.

Section 13 Serial Array Unit (SAU)

A single serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (simplified SPI or CSI^{Note}), UART, and simplified I²C communication. Function assignment of each channel supported by the RL78/G22 is as shown below.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

<16-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		IIC11

<20-, 24-, and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	—		—

<36-, 40-, and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

<48-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

When UART0 is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used, but CSI11, UART1, and IIC11 in channel 3 can be used.

Caution Most of the following descriptions in this section use the units and channels of the 48-pin products as an example.

13.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G22 has the following features.

13.1.1 Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **13.5 Operation of Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) Communication**.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate^{Note} During master communication: Max. $f_{CLK}/2$ (CSI00 only)

Max. $f_{CLK}/4$

During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$** .

13.1.2 UART (UART0 to UART2)

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **13.6 Operation of UART (UART0 to UART2) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits **Note**
- MSB/LSB first selectable
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

UART2 (channels 0 and 1 of unit 1) in 30-pin to 48-pin products only supports the LIN-bus.

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

Note UART0 only supports the 9-bit data length.

13.1.3 Simplified I²C (IIC00, IIC01, IIC11, IIC20, IIC21)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **13.8 Operation of Simplified I²C (IIC00, IIC01, IIC11, IIC20, and IIC21) Communication.**

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function **Note** and ACK detection function
- Data length of 8 bits
(When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See **13.8.3 (2) Processing flow** for details.

Remark 1. To use an I²C bus of full function, see **Section 14 Serial Interface IICA (IICA).**

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3)

13.2 Configuration of Serial Array Unit

The serial array unit includes the following registers, and input and output pins.

Table 13 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 or 9 bits ^{Note 1}
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) ^{Notes 1, 2}
Serial clock I/O	SCK00, SCK01, SCK11, SCK20, and SCK21 pins (for simplified SPI), SCL00, SCL01, SCL11, SCL20, and SCL21 pins (for simplified I ² C)
Serial data input	SI00, SI01, SI11, SI20, and SI21 pins (for simplified SPI), RxD0 and RxD1 pins (for UART), RxD2 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO01, SO11, SO20, and SO21 pins (for simplified SPI), TxD0 and TxD1 pins (for UART), TxD2 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA01, SDA11, SDA20, and SDA21 pins (for simplified I ² C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOM) • Serial output level register m (SOLm) • Serial standby control register 0 (SSC0) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0) <Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode registers 0, 1, 7 (PIM0, PIM1, PIM7) • Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7) • Port mode control A register 0 (PMCA0) • Port mode control T registers 0, 1, 3, 5, 7 (PMCT0, PMCT1, PMCT3, PMCT5, PMCT7) • Port mode registers 0, 1, 3, 5, 7 (PM0, PM1, PM3, PM5, PM7) • Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7) • Port function output enable register (PFOE1) • UART loopback select register (ULBS)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.

- 16- to 48-pin products and mn = 00, 01: Lower 9 bits
- Other than above: Lower 8 bits

Note 2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- SIOp (CSIp data register) in the CSIp communications
- RxDq (UARTq receive data register) in the UARTq reception
- TxDq (UARTq transmit data register) in the UARTq transmission
- SIOr (IICr data register) in the IICr communication

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20, 21)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20, 21)

Figure 13 - 1 shows the block diagram of serial array unit 0.

Figure 13 - 1 Block Diagram of Serial Array Unit 0

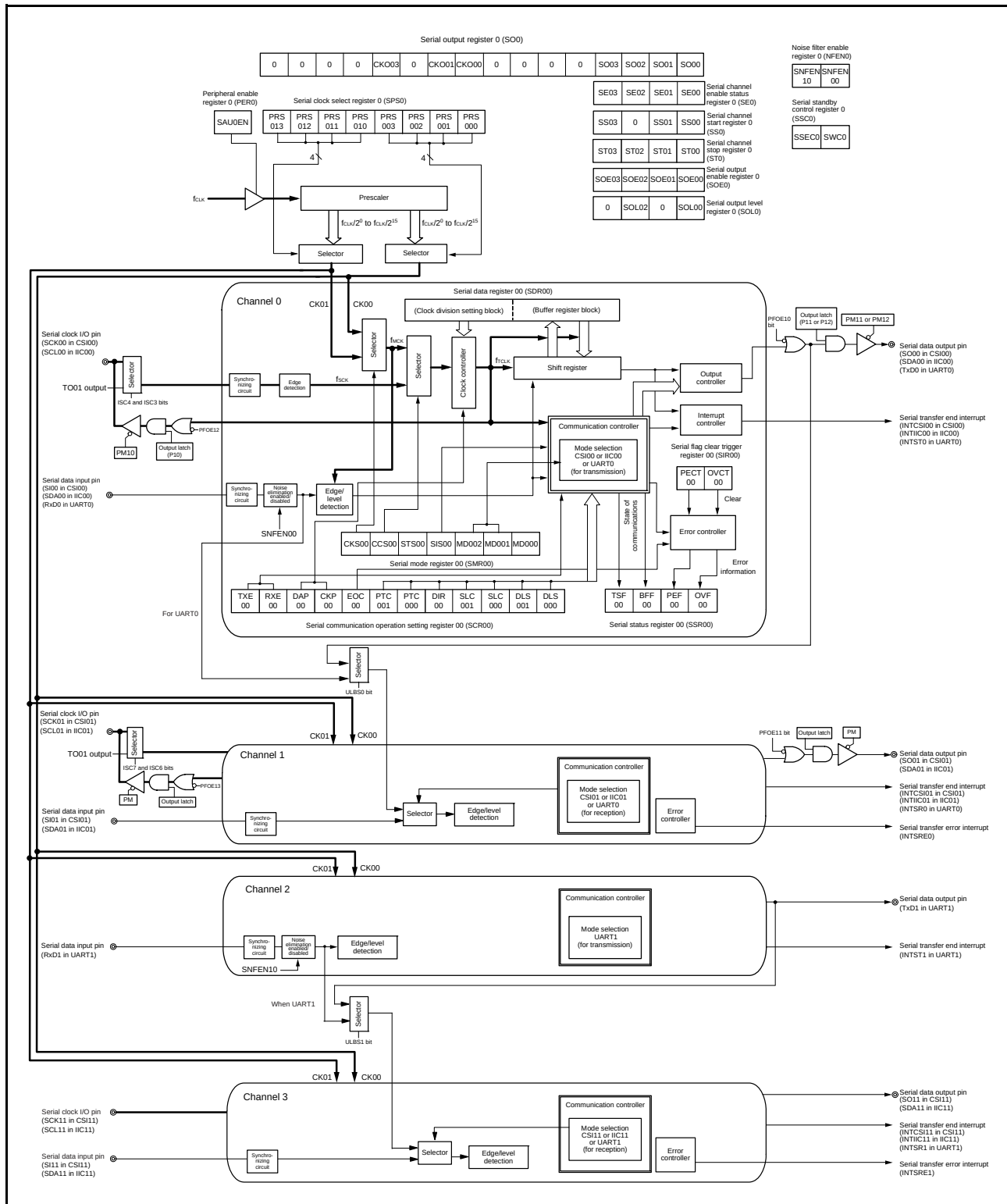
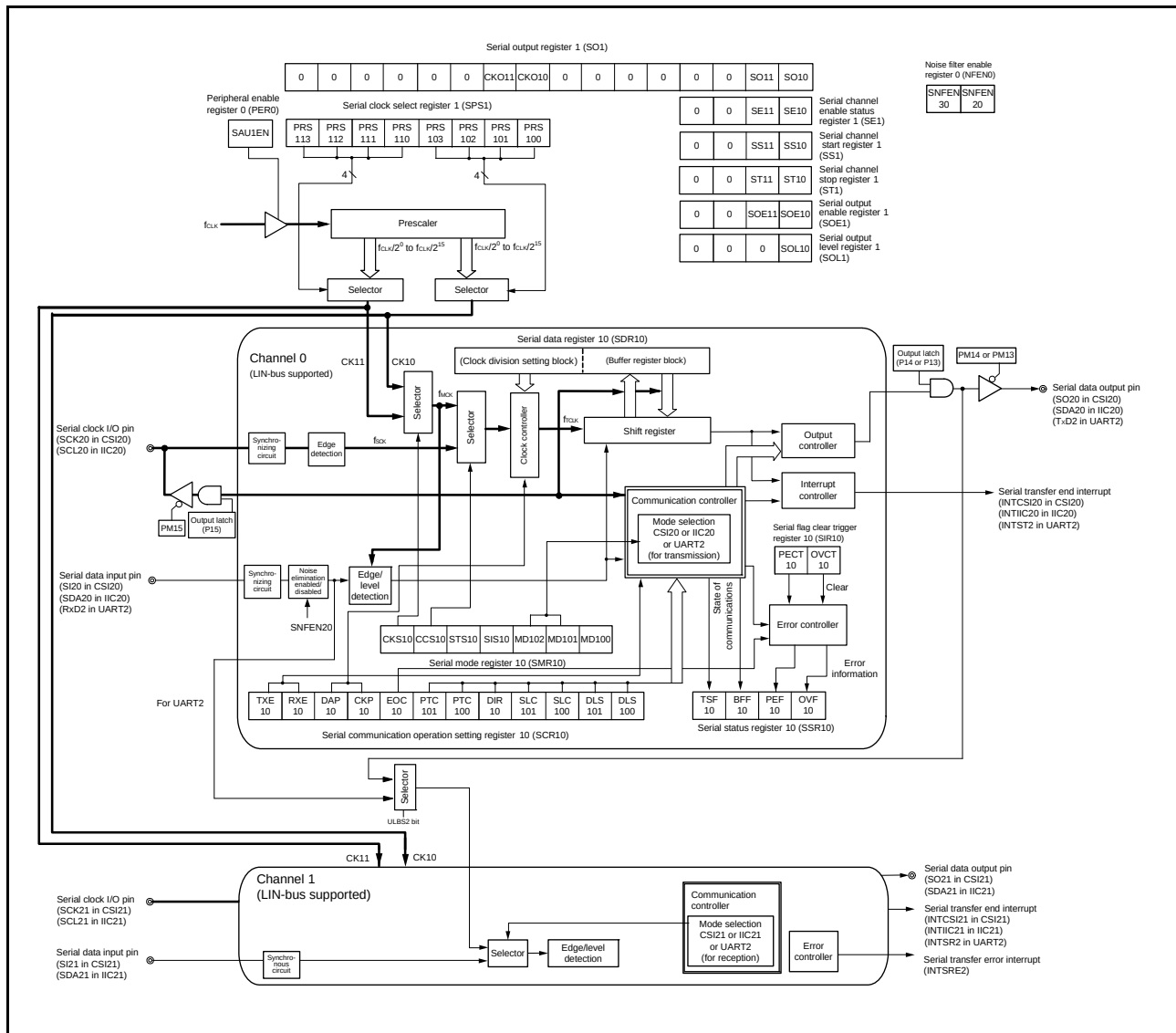


Figure 13 - 2 shows the block diagram of serial array unit 1.

Figure 13 - 2 Block Diagram of Serial Array Unit 1



13.2.1 Shift Register

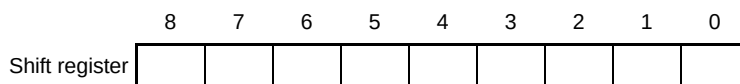
This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used **Note 1**.

During reception, it converts data input to the serial pin into parallel data. When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 or 9 bits of serial data register mn (SDRmn).



13.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn)

The SDRmn is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits)**Note 1** or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operating clock (fmck).

When data is received, parallel data converted by the shift register is stored in the lower 8 or 9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8 or 9 bits.

The data stored in the lower 8 or 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of the SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of the SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of the SDRmn register)**Note 1**

The SDRmn register can be read or written in 16-bit units.

The lower 8 or 9 bits of the SDRmn register can be read or written**Note 2** as the following SFR, depending on the communication mode.

- CSIp communication ... SIOp (CSIp data register)
- UARTq reception ... RxDq (UARTq receive data register)
- UARTq transmission ... TxDq (UARTq transmit data register)
- IICr communication ... SIOr (IICr data register)

The value of each SDRmn register following a reset is 0000H.

Note 1. UART0 only supports the 9-bit data length.

Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, 0 is stored in bits 0 to 8 in bit portions that exceed the data length.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20, 21)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20, 21)

Figure 13 - 3 Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

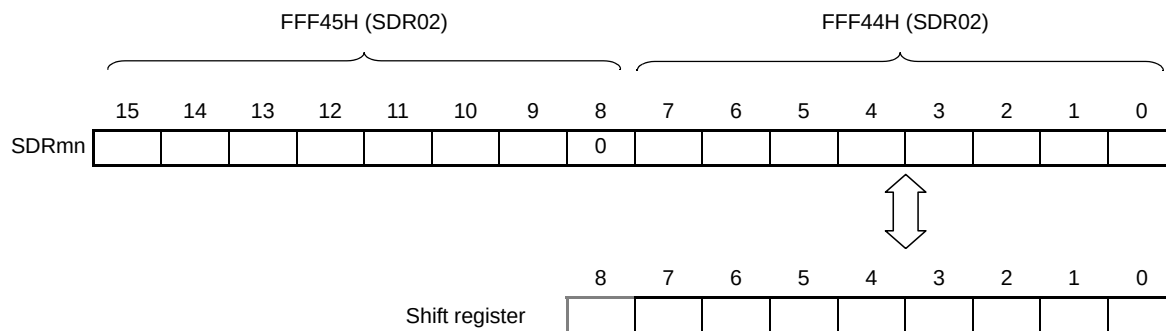
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
After reset: 0000H
R/W: R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers for Controlling the Serial Array Unit**.

Figure 13 - 4 Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 10, 11)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)
After reset: 0000H
R/W: R/W



Caution Be sure to clear bit 8 to 0.

Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers for Controlling the Serial Array Unit**.

13.3 Registers for Controlling the Serial Array Unit

The following registers are used to control the serial array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output register m (SOM)
- Serial output level register m (SOLm)
- Serial standby control register 0 (SSC0)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port function output enable register 1 (PFOE1)
- Port mode registers (PMxx)
- Port registers (Pxx)
- UART loopback select register (ULBS)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If serial array unit 0 is to be used, be sure to set bit 2 (SAU0EN) of this register to 1.

If serial array unit 1 is to be used, be sure to set bit 3 (SAU1EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the PER0 register following a reset is 00H.

Figure 13 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H

After reset: 00H

R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

SAUmEN	Control of supply of an input clock to serial array unit m
0	Stops supply of an input clock. - The SFRs used by serial array unit m cannot be written. - When an SFR used by serial array unit m is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by serial array unit m can be read and written.

(Cautions are listed on the next page.)

Caution 1. When setting serial array unit m, start by setting the following registers while the setting of the SAUmEN bit is 1.

- Port function output enable register 1 (PFOE1)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOM)
- Serial standby control register 0 (SSC0)

If the setting of the SAUmEN bit is 0, writing to any of the registers which control serial array unit m is ignored. Note, however, writing to the following registers is valid.

- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 7 (PIM0, PIM1, PIM7)
- Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)
- Port mode control A register 0 (PMCA0)
- Port mode control T register 0 (PMCT0, PMCT1, PMCT3, PMCT5, PMCT7)
- Port mode registers 0, 1, 3, 5, 7 (PM0, PM1, PM3, PM5, PM7)
- Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)

Caution 2. Be sure to clear the following bits to 0.

Bits 1, 3, 4, and 6 in the 16- and 20-pin products

Bits 1, 3, and 6 in the 24- and 25-pin products

Bits 1 and 6 in the 30-, 32-, 36-, 40-, and 48-pin products

13.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

Set bits 2 (SAU0RES) and 3 (SAU1RES) to 1 to reset serial array units 0 and 1, respectively.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the PRR0 register following a reset is 00H.

Figure 13 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H

After reset: 00H

R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

SAUmRES	Control resetting of serial array unit m
0	Serial array unit m is released from the reset state.
1	Serial array unit m is in the reset state. • The SFRs for use with serial array unit m are initialized.

Caution 1. Be sure to set bits 7, 6, and 1 to 0.

Caution 2. The functions mounted depend on the product. For details on the PRR0 register, see Section 21 Reset Function.

13.3.3 Serial clock select register m (SPSm)

The SPSm is a 16-bit register that is used to select two types of operating clocks (CKm0 and CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEMn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

The value of each SPSm register following a reset is 0000H.

Figure 13 - 7 Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1)^{Note}

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SPSm	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	Selection of operating clock (CKmk) ^{Note}					
					fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz
0	0	0	0	fCLK	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	fCLK/2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	fCLK/2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	fCLK/2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	fCLK/2 ⁵	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	fCLK/2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	fCLK/2 ⁷	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	fCLK/2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	fCLK/2 ⁹	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	fCLK/2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	fCLK/2 ¹¹	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	fCLK/2 ¹²	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	fCLK/2 ¹³	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	fCLK/2 ¹⁴	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	fCLK/2 ¹⁵	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Caution Be sure to clear bits 15 to 8 to 0.

Remark 1. fCLK: CPU/peripheral hardware clock frequency

Remark 2. m: Unit number (m = 0, 1)

Remark 3. k = 0, 1

13.3.4 Serial mode register mn (SMRmn)

The SMRmn register is used to set an operation mode of channel n. It is also used to select an operating clock (fmck), specify whether the serial clock (fsck) may be input or not, set a start trigger, the operating mode (as simplified SPI or CSI, UART, or simplified I²C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEMn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

The value of each SMRmn register following a reset is 0020H.

Figure 13 - 8 Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03),
F0150H, F0151H (SMR10), F0152H, F0153H (SMR11)^{Note 1}
After reset: 0020H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn

7	6	5	4	3	2	1	0
0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operating clock (fmck) of channel n
0	Operating clock CKm0 set by the SPSm register
1	Operating clock CKm1 set by the SPSm register

Operating clock (fmck) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (fTCLK) is generated.

CCSmn	Selection of transfer clock (fTCLK) of channel n
0	Divided operating clock fmck specified by the CKSmn bit
1	Clock input fsck from the SCKp pin (slave transfer in simplified SPI or CSI mode)

Transfer clock fTCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operating clock (fmck) is set by the higher 7 bits of the SDRmn register.

STSmn Note 2	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI or CSI, UART transmission, and simplified I ² C).
1	Valid edge of the RxDq pin (selected for UART reception)

Transfer is started when the above source is satisfied after 1 is set to the SSm register.

Figure 13 - 8 Format of Serial Mode Register mn (SMRmn) (2/2)

SISmn0 Note 2	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MDmn2	MDmn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I ² C mode
1	1	Setting prohibited

MDmn0 Note 2	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.	

Note 1. SMR00 and SMR01 are present in all products.
SMR02 and SMR03 are only present in 20- to 48-pin products.
SMR10 and SMR11 are only present in 30- to 48-pin products.

Note 2. This bit is only present in the SMR01, SMR03, and SMR11 registers.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, SMR02, and SMR10 registers) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20, 21),
q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 11, 20, 21)

13.3.5 Serial communication operation setting register mn (SCRmn)

The SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEMn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

The value of each SCRmn register following a reset is 0087H.

Figure 13 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03),
F0158H, F0159H (SCR10), F015AH, F015BH (SCR11)^{Note 1}

After reset: 0087H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0

7	6	5	4	3	2	1	0
DIRmn	0	SLCmn1	SLCmn0	0	1	DLSmn1	DLSmn0

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0	SCKp SOp Slp input timing	1
0	1	SCKp SOp Slp input timing	2
1	0	SCKp SOp Slp input timing	3
1	1	SCKp SOp Slp input timing	4

Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I²C mode.

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 2))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).

Set EOCmn = 0 in the simplified SPI (CSI) mode, simplified I²C mode, and during UART transmission^{Note 4}.

Figure 13 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

PTCmn1	PTCmn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity ^{Note 5} .	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the simplified SPI (CSI) mode and simplified I ² C mode.			

DIRmn	Selection of data transfer sequence in simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I ² C mode.	

SLCmn1 Note 2	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits; this setting is only possible for mn = 00, 10.
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I²C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1 Note 3	DLSmn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) This setting is only possible in the UART mode.
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I ² C mode.		

- Note 1.** SCR00 and SCR01 are present in all products.
SCR02 and SCR03 are only present in 20- to 48-pin products.
SCR10 and SCR11 are only present in 30- to 48-pin products.
- Note 2.** This bit is only present in the SCR00, SCR02, and SCR10 registers.
- Note 3.** This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.
- Note 4.** Using CSImn with EOCmn being set to 1 may trigger an INTSREn error interrupt.
- Note 5.** 0 is always appended to the data value regardless of the contents.

Caution Be sure to clear bits 11, 6, and 3 to 0. Also clear bit 5 of the SCR01, SCR03, or SCR11 register to 0.
Be sure to set bit 2 to 1.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 11, 20, 21)

13.3.6 Serial data register mn (SDRmn)

The SDRmn is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR10, and SDR11 function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operating clock (fMCK).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00, SDR01, SDR10, and SDR11 to 0000000B. The input clock fSCK (slave transfer in simplified SPI or CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8 or 9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 or 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 or 9 bits.

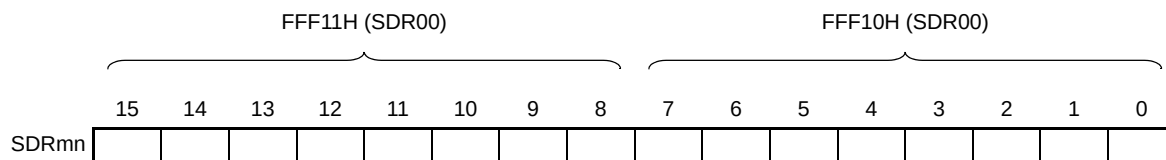
The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can only be written or read when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 or 9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

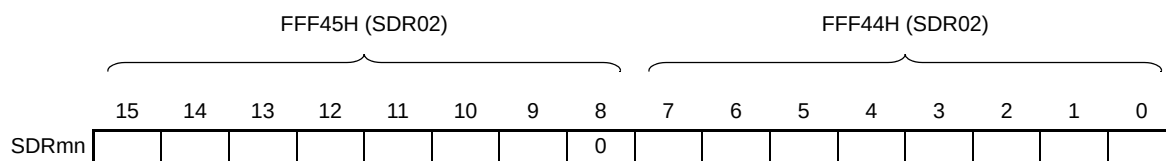
The value of each SDRmn register following a reset is 0000H.

Figure 13 - 10 Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
After reset: 0000H
R/W: R/W



Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)
After reset: 0000H
R/W: R/W



SDRmn[15:9]							Transfer clock setting by dividing the operating clock
0	0	0	0	0	0	0	fMCK/2
0	0	0	0	0	0	1	fMCK/4
0	0	0	0	0	1	0	fMCK/6
0	0	0	0	0	1	1	fMCK/8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	fMCK/254
1	1	1	1	1	1	1	fMCK/256

Caution 1. Be sure to clear bit 8 of the SDR02, SDR03, SDR10, and SDR11 to 0.

Caution 2. Setting SDRmn[15:9] to 0000000B or 0000001B is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] to 0000000B is prohibited when simplified I²C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. For the function of the lower 8 or 9 bits of the SDRmn register, see 13.2 Configuration of Serial Array Unit.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.7 Serial flag clear trigger register mn (SIRmn)

The SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFMn) of serial status register mn is cleared to 0. Because the SIRmn is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

The value of each SIRmn register following a reset is 0000H.

Figure 13 - 11 Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03),
F0148H, F0149H (SIR10), F014AH, F014BH (SIR11)^{Note 1}
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SIRmn	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	FECTmn	PECTmn	OVCTmn
FECTmn	Clear trigger of framing error flag of channel n							
0	Not cleared							
1	Clears the FEFmn bit of the SSRmn register to 0.							
PECTmn	Clear trigger of parity error flag of channel n							
0	Not cleared							
1	Clears the PEFmn bit of the SSRmn register to 0.							
OVCTmn	Clear trigger of overrun error flag of channel n							
0	Not cleared							
1	Clears the OVFMn bit of the SSRmn register to 0.							

Note 1. SIR00 and SIR01 are present in all products.
SIR02 and SIR03 are only present in 20- to 48-pin products.
SIR10 and SIR11 are only present in 30- to 48-pin products.

Note 2. This bit is only present in the SIR01, SIR03, and SIR11 registers.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, and SIR10 register) to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

13.3.8 Serial status register mn (SSRmn)

The SSRmn register indicates the state of communications and occurrence of errors for channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with an 8-bit memory manipulation instruction with SSRmnL.

The value of each SSRmn register following a reset is 0000H.

Figure 13 - 12 Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03),
F0140H, F0141H (SSR10) to F0142H, F0143H (SSR11)^{Note 1}

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
SSRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	TSFmn	BFFmn	0	0	FEFmn	PEFmn	OVFmn

TSFmn	Flag indicating the state of communications for channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> - The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). - Communication ends. <Set condition> - Communication starts.	

BFFmn	Flag indicating the state of the buffer register for channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> - Transferring transmit data from the SDRmn register to the shift register ends during transmission. - Reading receive data from the SDRmn register ends during reception. - The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set condition> - Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission/reception mode in each communication mode). - Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission/reception mode in each communication mode). - A reception error occurs.	

Figure 13 - 12 Format of Serial Status Register mn (SSRmn) (2/2)

FEFmn Note 2	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear conditions> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I ² C transmission).
<Clear conditions> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave at the ACK reception timing during I²C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear conditions> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission/reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission/reception in simplified SPI (CSI) mode.	

Note 1. SSR00 and SSR01 are present in all products.
SSR02 and SSR03 are only present in 20- to 48-pin products.
SSR10 and SSR11 are only present in 30- to 48-pin products.

Note 2. This bit is only present in the SSR01, SSR03, and SSR11 registers.

Caution 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVFmn = 1) is detected.

Caution 2. When the simplified SPI (CSI) is handling reception in the SNOOZE mode (SWC0 = 1), the BFFmn flag will not change.

Caution 3. When the simplified SPI (CSI) is handling reception in the SNOOZE mode (SWC0 = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.9 Serial channel start register m (SSm)

The SSm is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written to a bit (SSmn) of this register, the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with a 1-bit or 8-bit memory manipulation instruction with SSmL.

The value of each SSm register following a reset is 0000H.

Figure 13 - 13 Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1)^{Note}

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Set the SEmn bit to 1 to place the channel in the communications waiting state. ^{Note}

Note Setting an SSmn bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKmn and SOMn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution 1. Be sure to clear bits 15 to 4 of the SS0 register and bits 15 to 2 of the SS1 register to 0.

Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fmck clock cycles have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SSm register is read, 0000H is always read.

13.3.10 Serial channel stop register m (STm)

The STm is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written to a bit (STmn) of this register, the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

The value of each STm register following a reset is 0000H.

Figure 13 - 14 Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1)^{Note}

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation ^{Note} .

Note The values of the control registers and shift register, the states of the SCKmn and SOMn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution Be sure to clear bits 15 to 4 of the ST0 register and bits 15 to 2 of the ST1 register to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the STm register is read, 0000H is always read.

13.3.11 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written to a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

For channel n whose operation is enabled, the value of the CKOm_n bit of serial output register m (SOM) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin.

For channel n whose operation is stopped, the value of the CKOm_n bit of the SOM register can be set by software and is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be read with a 1-bit or 8-bit memory manipulation instruction with SEmL.

The value of each SEm register following a reset is 0000H.

Figure 13 - 15 Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
SE0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
SE1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SE11	SE10

SEm _n	Indication of whether operation of channel n is enabled or stopped.
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.12 Serial output enable register m (SOEm)

The SOEm register is used to enable or stop output of the serial communication operation of each channel.
For channel n whose serial output is enabled, the value of the SOMn bit of serial output register m (SOM) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOMn bit value of the SOM register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

The value of each SOEm register following a reset is 0000H.

Figure 13 - 16 Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	SOE03	SOE02	SOE01	SOE00

Address: F016AH, F016BH (SOE1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SOE11	SOE10

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 4 of the SOE0 register and bits 15 to 2 of the SOE1 register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.13 Serial output register m (SOM)

The SOM is a buffer register for serial output of each channel.

The value of the SOMn bit of this register is output from the serial data output pin of channel n.

The value of the CKOm bit of this register is output from the serial clock output pin of channel n.

The SOMn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOm bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0).

While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOm bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOm and SOMn bits to 1.

The SOM register can be set by a 16-bit memory manipulation instruction.

The value of each SOM register following a reset is 0F0FH.

Figure 13 - 17 Format of Serial Output Register m (SOM)

Address: F0128H, F0129H (SO0)

After reset: 0B0BH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO0	0	0	0	0	CKO03	0	CKO01	CKO00
	7	6	5	4	3	2	1	0
	0	0	0	0	SO03	SO02	SO01	SO00

Address: F0168H, F0169H (SO1)

After reset: 0303H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO1	0	0	0	0	0	0	CKO11	CKO10
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SO11	SO10

CKOm	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SOMn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Caution Be sure to clear bits 15 to 12, 10, and 7 to 4 of the SO0 register to 0.
Be sure to clear bits 15 to 10 and 7 to 2 of the SO1 register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3)

13.3.14 Serial output level register m (SOLm)

The SOLm register is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for the bit corresponding the channel used in the simplified SPI (CSI) mode or simplified I²C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1).

When serial output is disabled (SOEmn = 0), the value of the SOMn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEMn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

The value of each SOLm register following a reset is 0000H.

Figure 13 - 18 Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	SOL02	0	SOL00

Address: F0174H, F0175H (SOL1)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	SOL10

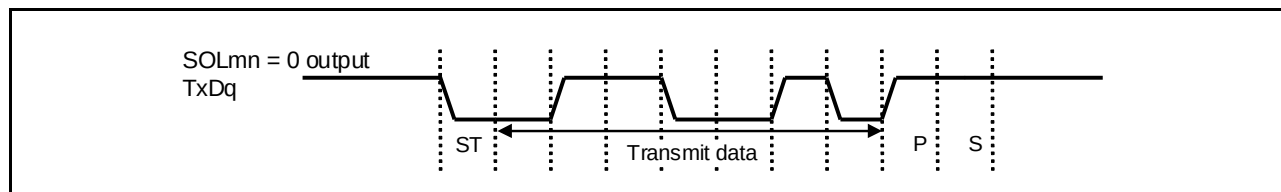
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 3, and 1 of the SOL0 register and bits 15 to 1 of the SOL1 register to 0.

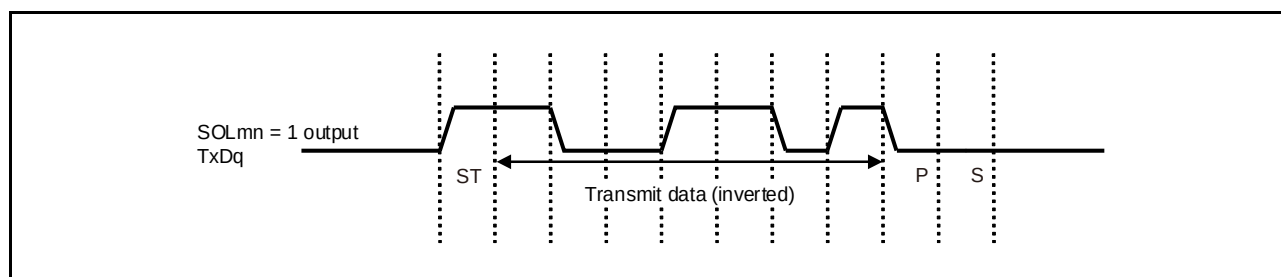
Figure 13 - 19 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 13 - 19 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

13.3.15 Serial standby control register 0 (SSC0)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC0 register can be set with an 8-bit memory manipulation instruction with SSC0L.

The value of each SSC0 register following a reset is 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- For CSI00: Up to 1 Mbps
- For UART0: Up to 115.2 kbps (when FWKUP = 1, fCLK = fIH (32 MHz))

Figure 13 - 20 Format of Serial Standby Control Register 0 (SSC0)

Address: F0138H, F0139H (SSC0)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SSC0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SSEC0	SWC0

SSEC0	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Disable the generation of error interrupts (INTSRE0).
• The SSEC0 bit can only be set to 1 or 0 when the setting of both the SWC0 and EOC00 bits are 1 during UART reception in the SNOOZE mode. In other cases, clear the SSEC0 bit to 0. • Setting SSEC0 and SWC0 to 1 and 0, respectively, is prohibited.	

SWC0	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and simplified SPI (CSI) or UART reception is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (fCLK). If any other clock is selected, specifying this mode is prohibited. • Even when using SNOOZE mode, be sure to set the SWC0 bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWC0 bit to 0 after having returned from STOP mode to normal operation mode.	

Figure 13 - 21 Interrupt in UART Reception Operation in SNOOZE Mode

EOC00 Bit	SSEC0 Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSR0 is generated.	INTSR0 is generated.
0	1	INTSR0 is generated.	INTSR0 is generated.
1	0	INTSR0 is generated.	INTSRE0 is generated.
1	1	INTSR0 is generated.	No interrupt is generated.

13.3.16 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus communication operation by UART2 in coordination with an external interrupt and the timer array unit.

The ISC4 to ISC2 bits are used to select the serial data input source and serial clock input source of CSI00.

The ISC7 to ISC5 bits are used to select the serial data input source and serial clock input source of CSI01.

When bit 0 is set to 1, the input signal of the serial data input (RxD2) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD2) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

Bits 3 and 4 can be used to select the SCK00 pin input or TO01 output signal as the serial clock input source of CSI00.

Bits 6 and 7 can be used to select the SCK01 pin input or TO01 output signal as the serial clock input source of CSI01.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the ISC register following a reset is 00H.

Figure 13 - 22 Format of Input Switch Control Register (ISC)

Address: F0073H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	0	ISC4	ISC3	0	ISC1	ISC0

ISC7	ISC6	Switch of the serial clock input source of CSI01 ^{Note 1}
0	0	Input signal of the SCK01 pin (normal operation)
0	1	Setting prohibited
1	0	TO01 output signal
1	1	Setting prohibited

ISC4	ISC3	Switch of the serial clock input source of CSI00 ^{Note 2}
0	0	Input signal of the SCK00 pin (normal operation)
0	1	Setting prohibited
1	0	TO01 output signal
1	1	Setting prohibited

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD2 pin as an external interrupt (wakeup signal detection).

Note 1. When UART mode or simplified I²C mode is to be selected for channel 1, set the ISC7 and ISC6 bits to 0.

Note 2. When UART mode or simplified I²C mode is to be selected for channel 0, set the ISC4 and ISC3 bits to 0.

13.3.17 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for simplified SPI (CSI) or simplified I²C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1. When the noise filter is enabled, after synchronization is performed with the operating clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operating clock (f_{MCK}) of the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the NFEN0 register following a reset is 00H.

Figure 13 - 23 Format of Noise Filter Enable Register 0 (NFEN0) (1/2)

Address: F0070H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	Use of noise filter of RxD2 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 5, 3, and 1 to 0.

13.3.18 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel. For the related registers, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.4 Port input mode registers (PIMxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.10 Port function output enable register 1 (PFOE1)**

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P13/TxD2/SO20/TS14/(SDAA0)/(TI04)/(TO04)) for serial data or serial clock output, requires setting the corresponding bits in the port mode control A register (PMCAxx), port mode control T register (PMCTxx), and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1. Set the corresponding bit in the port function output enable register 1 (PFOE1) to 1. When using the port pin in N-ch open-drain output (withstand voltage of VDD) mode, set the corresponding bit in the port output mode register (POMxx) to 1. To communicate with an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Example: When P13/TxD2/SO20/TS14/(SDAA0)/(TI04)/(TO04) is to be used for serial data output

Set the PMCT13 bit of port mode control T register 1 to 0.

Set the PM13 bit of port mode register 1 to 0.

Set the P13 bit of port register 1 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P11/SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06)) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bits in the port mode control A register (PMCAxx) and port mode control T register (PMCTxx) to 0. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. To communicate with an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Example: When P11/SI00/RxD0/TOOLRxD/SDA00/TS12/(TI06)/(TO06) is to be used for serial data input

Set the PMCT11 bit of port mode control T register 1 to 0.

Set the PM11 bit of port mode register 1 to 1.

Set the P11 bit of port register 1 to 0 or 1.

13.3.19 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback function. This register has bits to individually control UART channels. When the bit corresponding to each channel is set to 1, the UART loopback function is selected, and output from the transmission shift register is looped back to the reception shift register.

The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the ULBS register following a reset is 00H.

Figure 13 - 24 Format of UART Loopback Select Register (ULBS)

Address: F0079H

After reset: 00H

R/W: R/W

Symbol	7	6	5	<4>	3	<2>	<1>	<0>
ULBS	0	0	0	ULBS4	0	ULBS2	ULBS1	ULBS0

ULBSn	Selection of the UART loopback function
0	Inputs the states of the RxD0 to RxD2 pins of serial array unit UART0 to UART2 to the reception shift register.
1	Loops back output from the transmission shift register to the reception shift register.

Caution Be sure to set bits 7 to 5 and 3 to 0.

Remark n = 0 to 2

13.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the pin for serial interface can be used as port function pins in this mode.

13.4.1 Stopping the Operation by Units

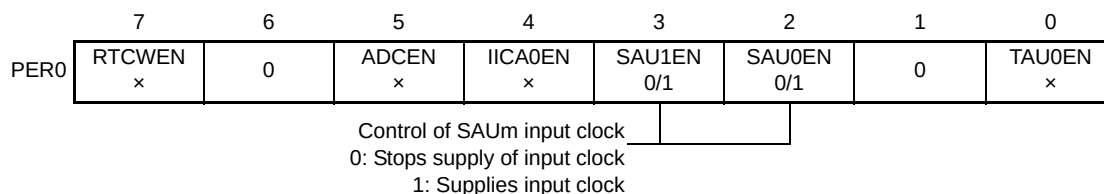
The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0. To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 13 - 25 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) ... Set only the bit of SAUm to be stopped to 0.



Caution 1. If SAUmEN = 0, writing to the registers which control serial array unit m is ignored, and, even if the register is read, only the initial value is read

Note that this does not apply to the following registers.

- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 7 (PIM0, PIM1, PIM7)
- Port output mode registers 0, 1, 5, 7 (POM0, POM1, POM5, POM7)
- Port mode control A register 0 (PMCA0)
- Port mode control T registers 0, 3 (PMCT0, PMCT3)
- Port mode registers 0, 1, 3, 5, 7 (PM0, PM1, PM3, PM5, PM7)
- Port registers 0, 1, 3, 5, 7 (P0, P1, P3, P5, P7)

Caution 1. Be sure to clear the bits 6 and 1 to 0.

Caution 2. The functions mounted depend on the product. For details on the PER0 register, see Section 21 Reset Function.

Remark x: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user.

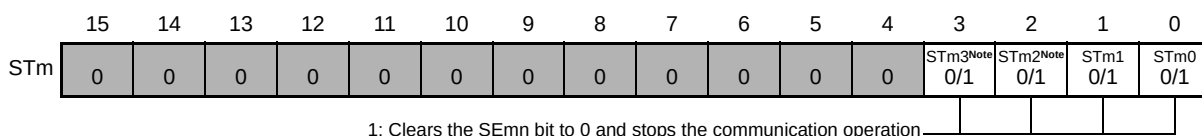
13.4.2 Stopping the Operation by Channels

The stopping of the operation by channels is set using each of the following registers.

Figure 13 - 26 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)

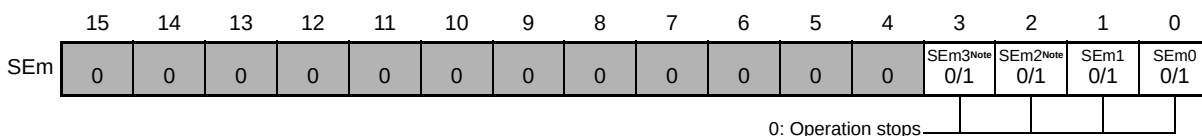
... The STm is a trigger register that is used to enable stopping communication/count by each channel.



* Because the STmn bit is a trigger bit, it is cleared immediately when SEMn = 0.

(b) Serial channel enable status register m (SEm)

... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.



* The SEM is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOm bit of the SOM register can be set by software.

(c) Serial output enable register m (SOEm)

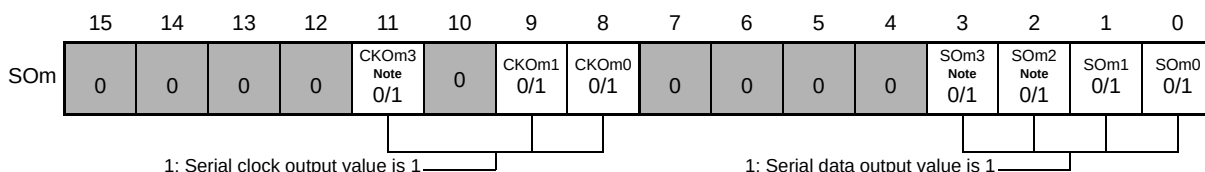
... This register is used to enable or stop output of the serial communication operation of each channel.



* For channel n, whose serial output is stopped, the SOMn bit value of the SOM register can be set by software.

(d) Serial output register m (SOM)

...The SOM is a buffer register for serial output of each channel.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOm, SOMn bits to 1.

Note This bit is only present in serial array unit 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user.

13.5 Operation of Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate^{Note}

During master communication:	Max. $f_{CLK}/2$ (CSI00 only)
	Max. $f_{CLK}/4$
During slave communication:	Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$.**

Caution Use a general-purpose port pin to send a chip select signal when required.

The channels supporting simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) are channels 0, 1, and 3 of SAU0, and channels 0 and 1 of SAU1.

<16-pin products>

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		IIC11

<20-, 24-, and 25-pin products>

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

<30- and 32-pin products>

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	—		—

<36-, 40-, and 44-pin products>

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

<48-pin products>

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

Simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) handles the following seven types of communications.

- Master transmission (See **13.5.1.**)
- Master reception (See **13.5.2.**)
- Master transmission/reception (See **13.5.3.**)
- Slave transmission (See **13.5.4.**)
- Slave reception (See **13.5.5.**)
- Slave transmission/reception (See **13.5.6.**)
- SNOOZE mode function (See **13.5.7.**)

13.5.1 Master Transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.				
Error detection flag	None				
Transfer data length	7 or 8 bits				
Transfer rate ^{Note}	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK} (2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

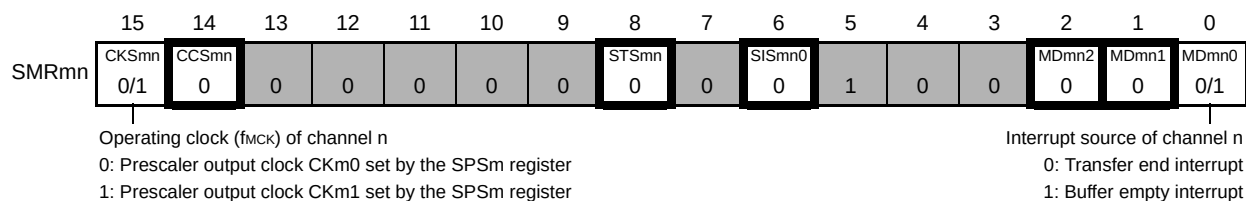
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

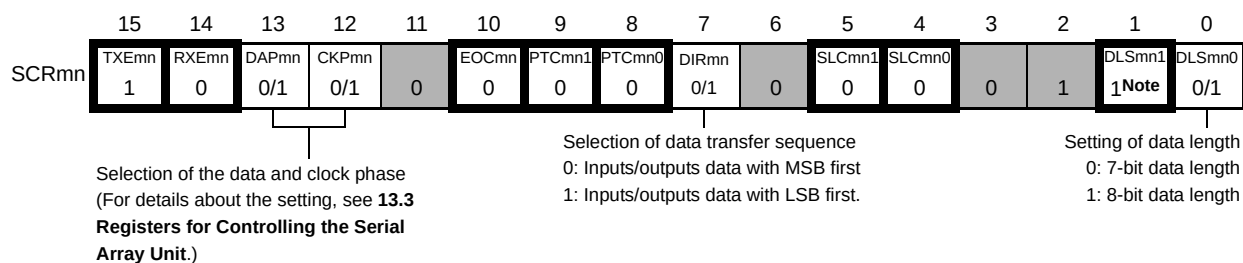
(1) Register setting

Figure 13 - 27 Example of Contents of Registers for Master Transmission of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

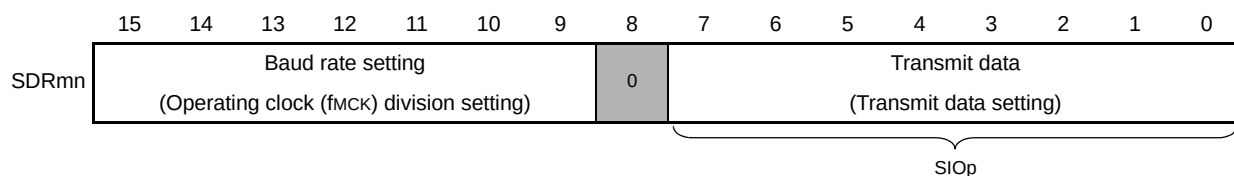
(a) Serial mode register mn (SMRmn)



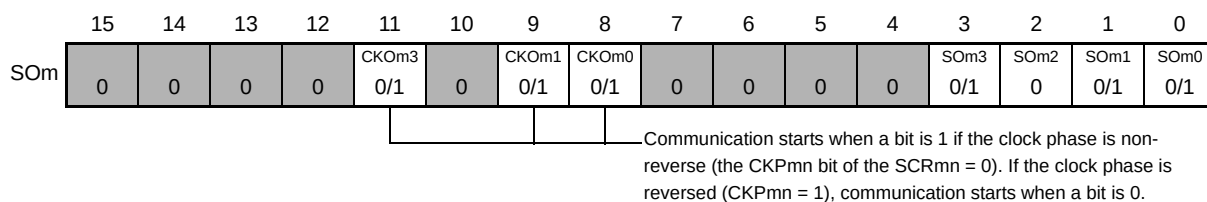
(b) Serial communication operation setting register mn (SCRmn)



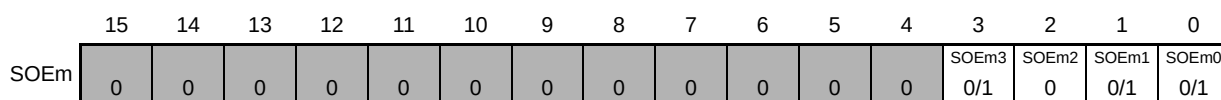
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOM) ... Set only the bit of the target channel.



(e) Serial output enable register m (SOEm) ... Set only the bit of the target channel to 1.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the simplified SPI (CSI) master transmission mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 28 Initial Setting Procedure for Master Transmission

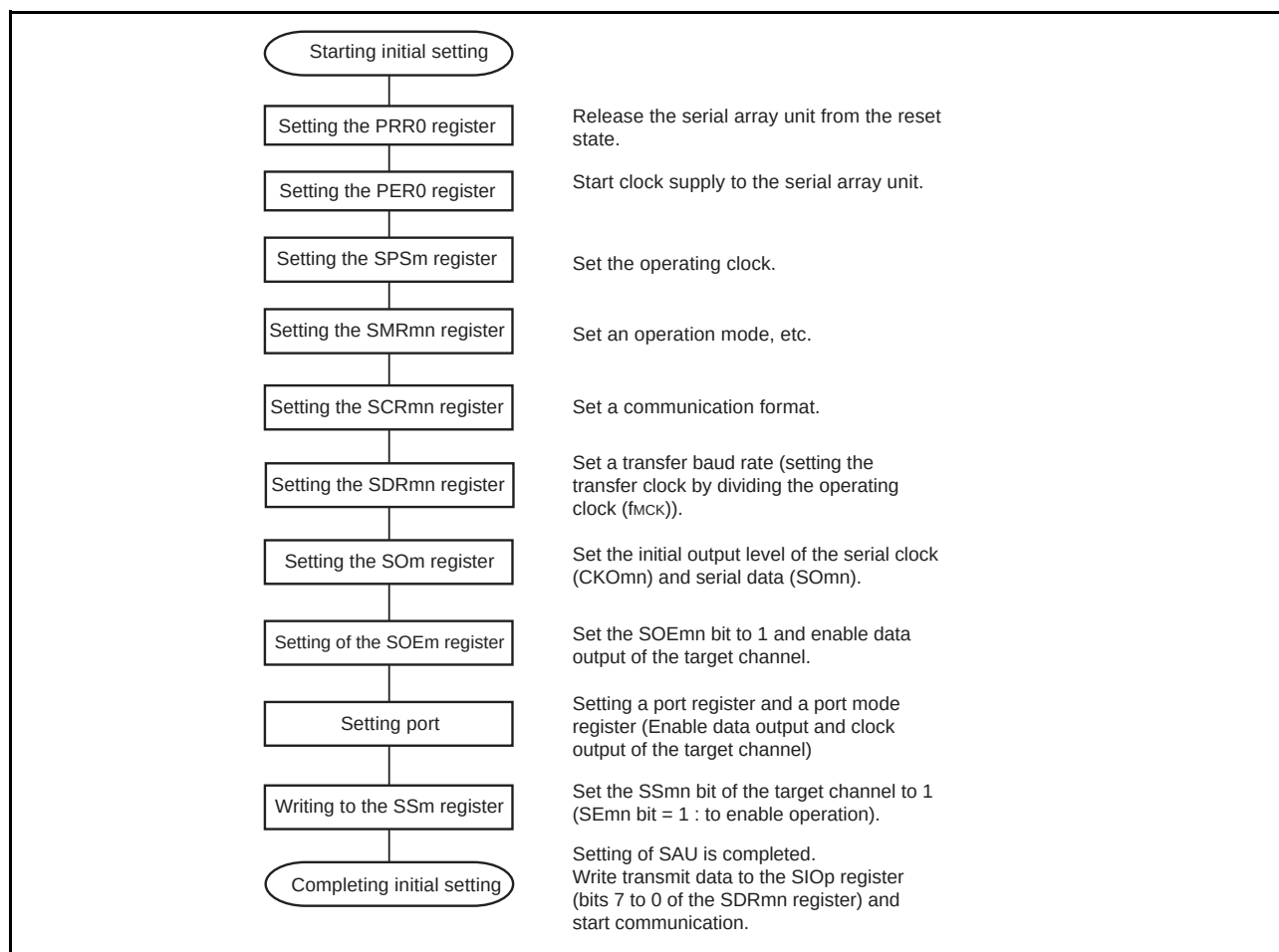


Figure 13 - 29 Procedure for Stopping Master Transmission

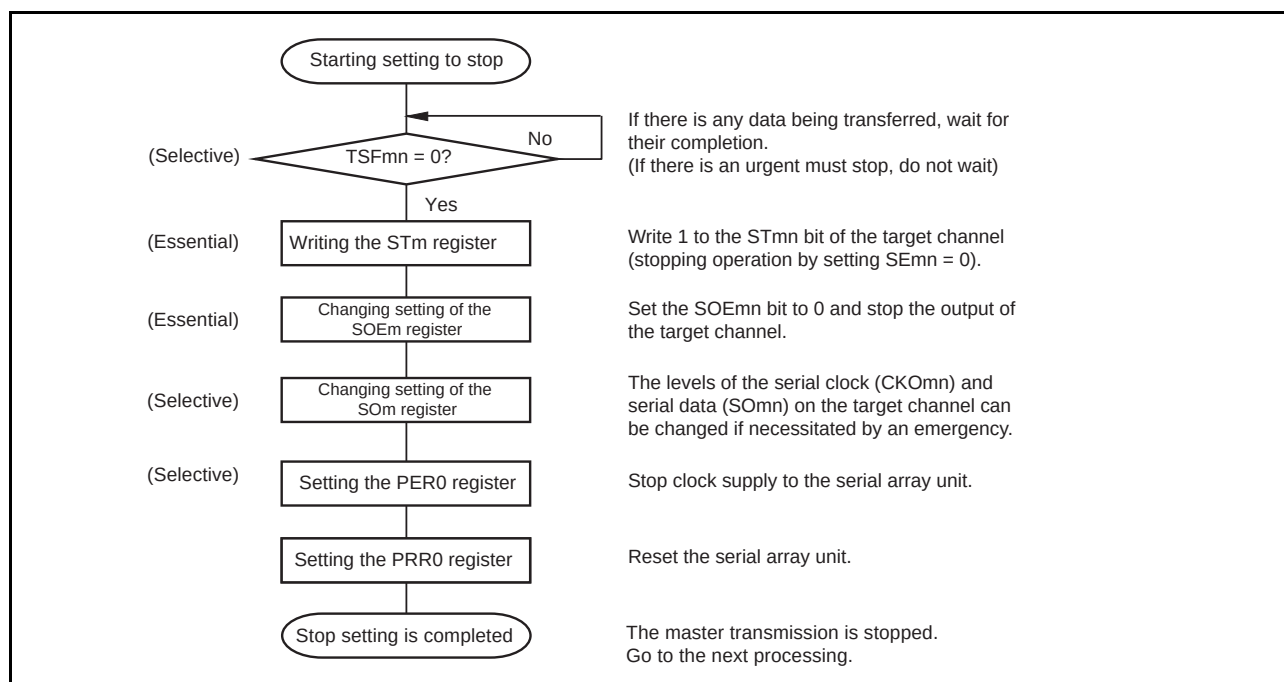
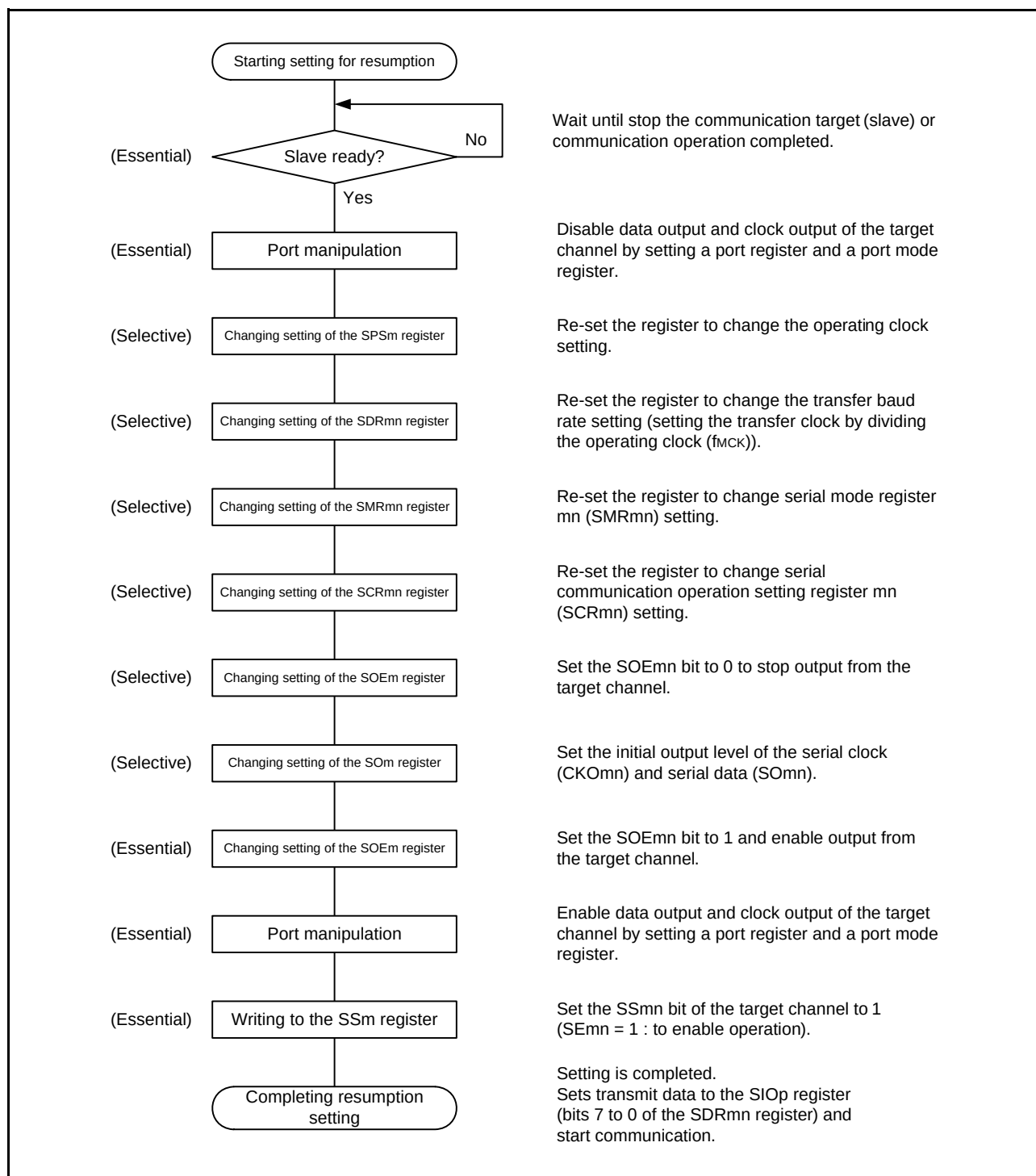


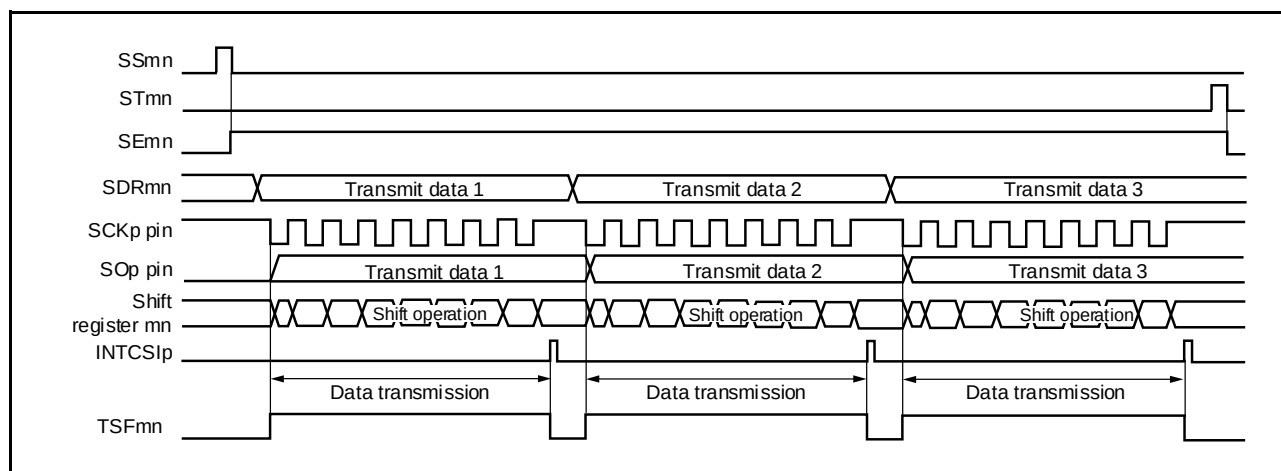
Figure 13 - 30 Procedure for Resuming Master Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then proceed initialization instead of restarting the communication.

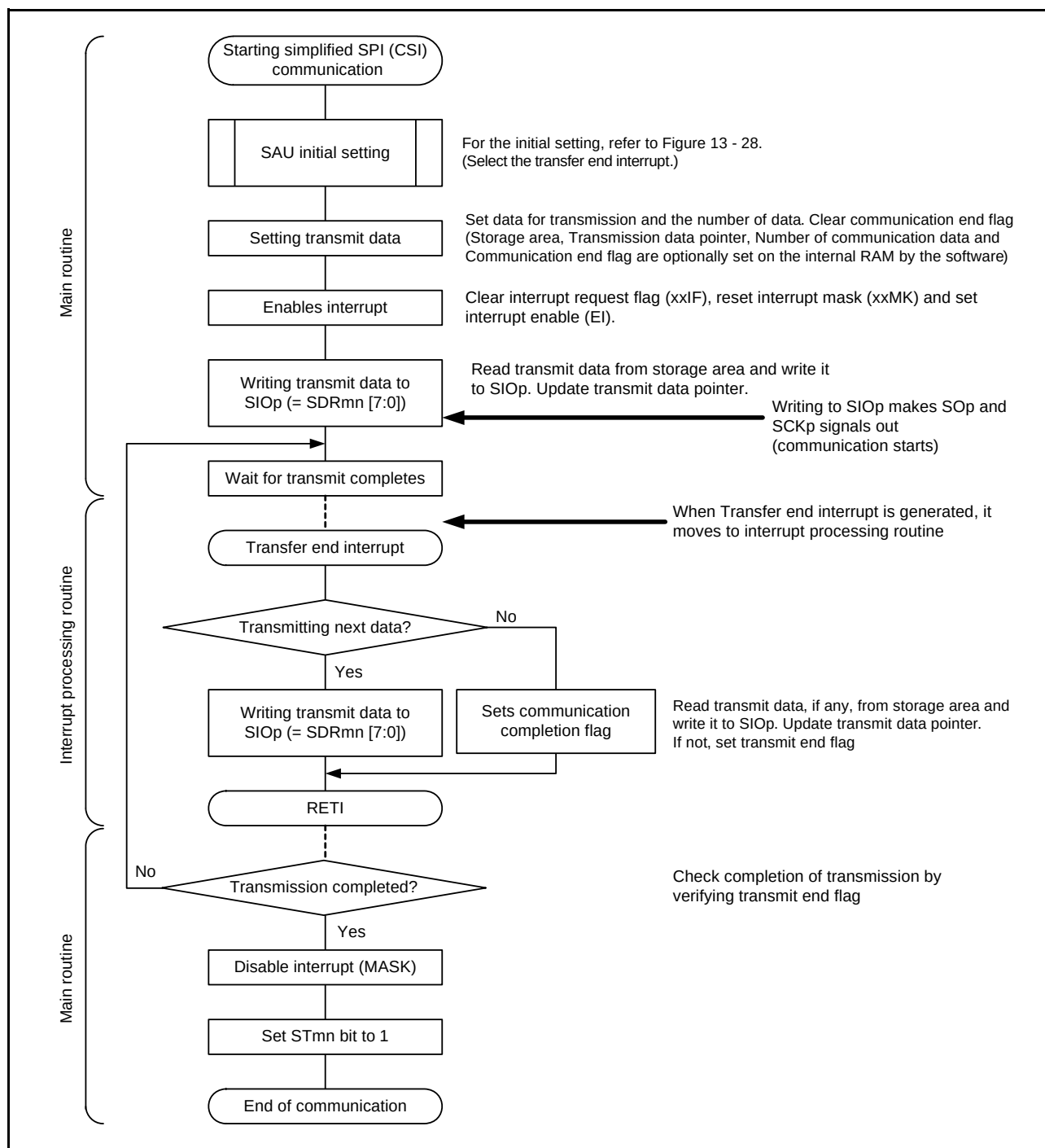
(3) Processing flow (in single-transmission mode)

Figure 13 - 31 Timing Chart of Master Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



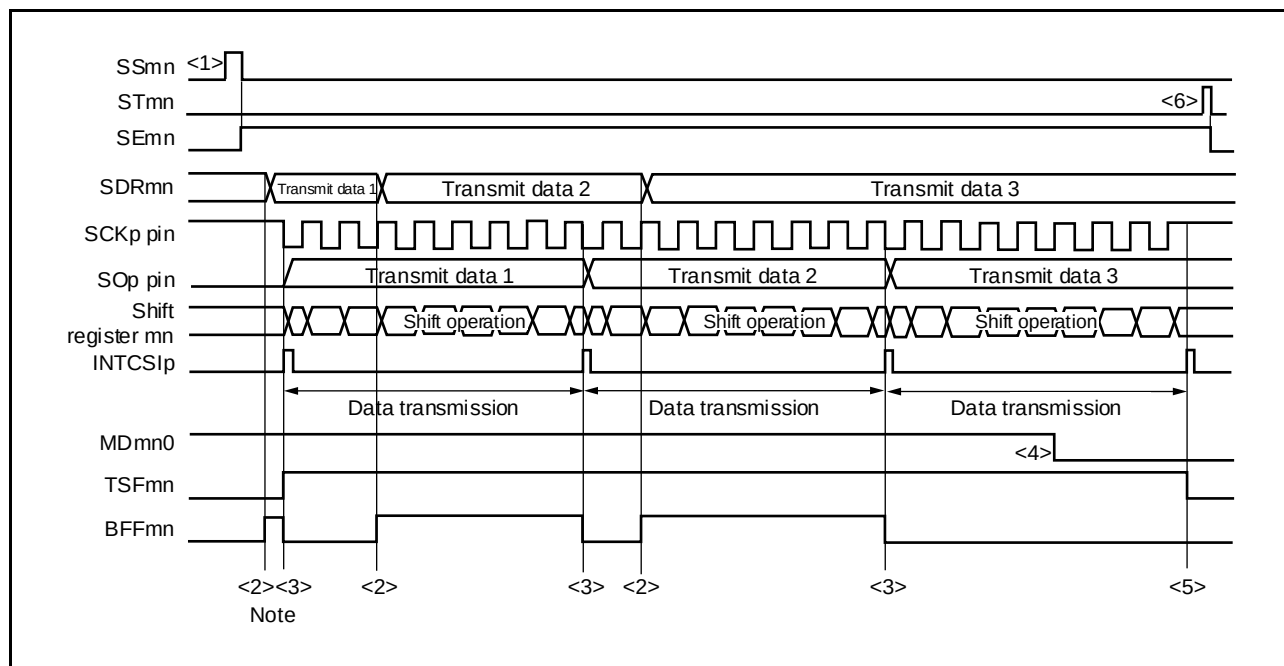
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 32 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

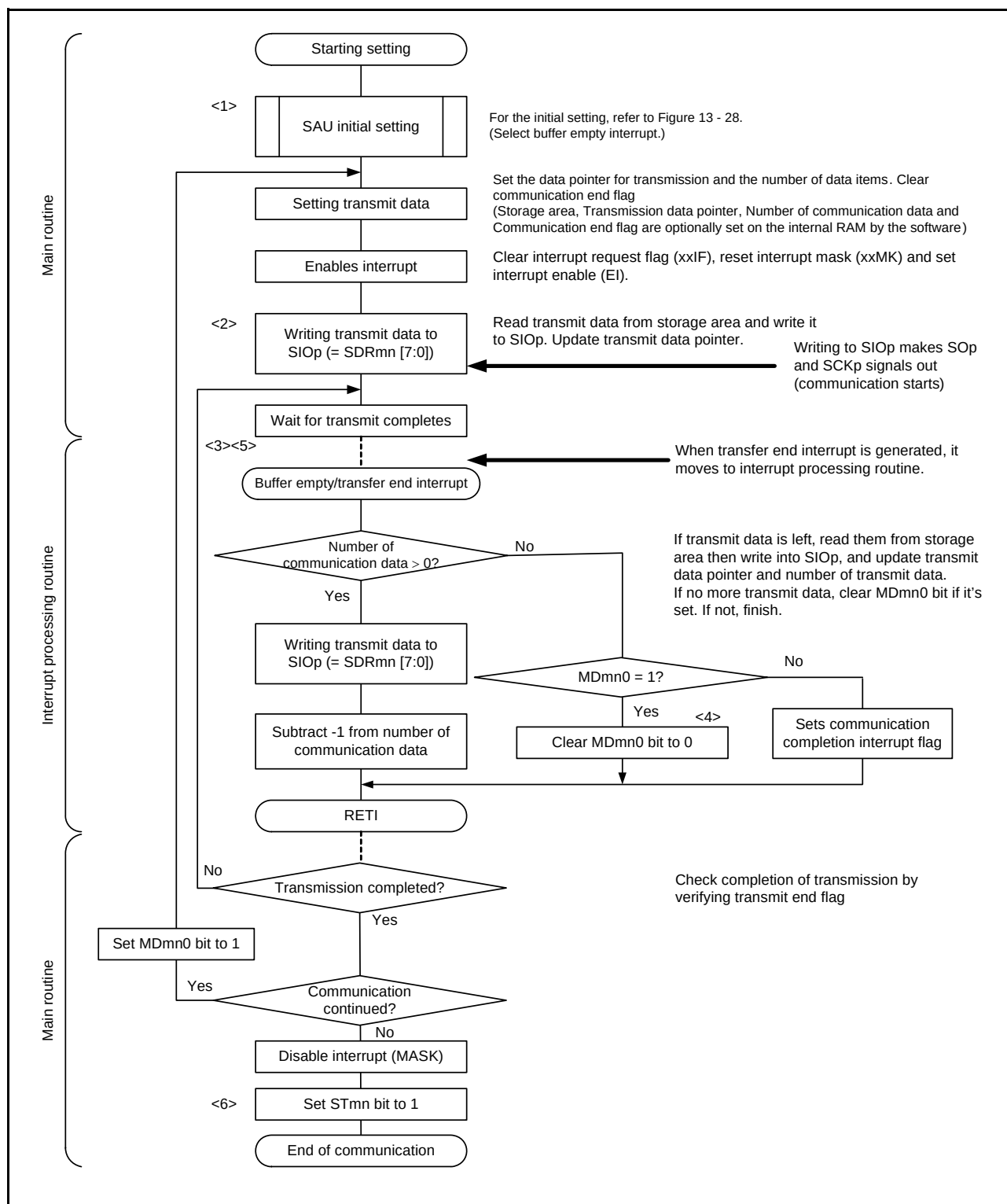


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 34 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.2 Master Reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from another device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	7 or 8 bits				
Transfer rate ^{Note}	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

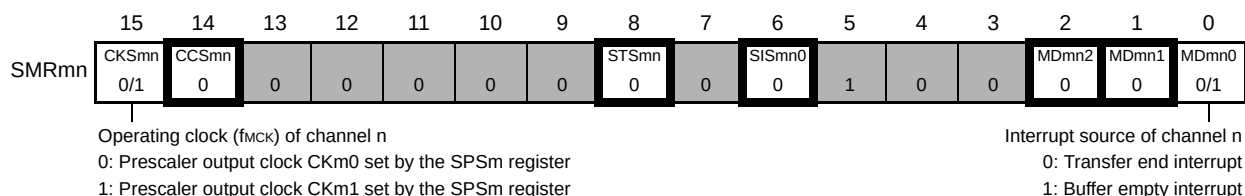
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

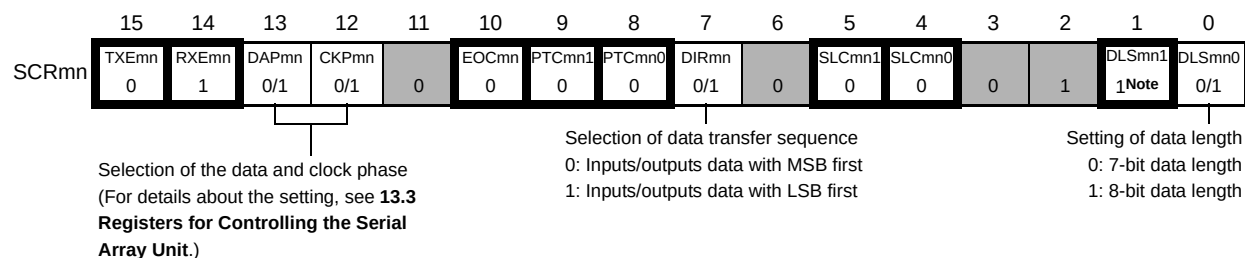
(1) Register setting

Figure 13 - 35 Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

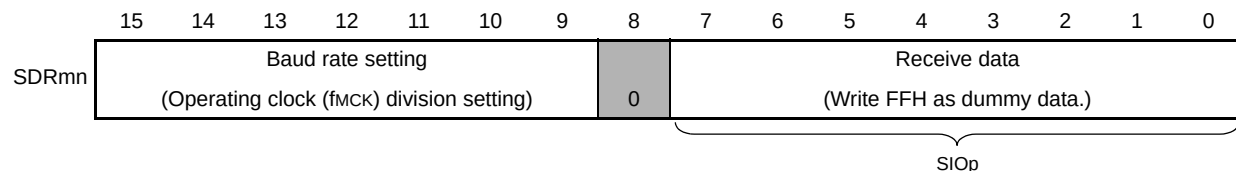
(a) Serial mode register mn (SMRmn)



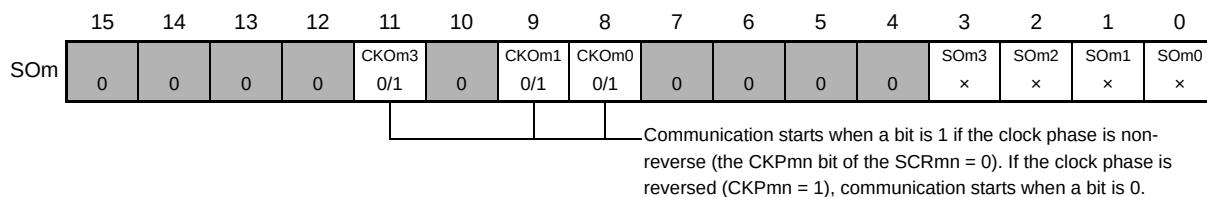
(b) Serial communication operation setting register mn (SCRmn)



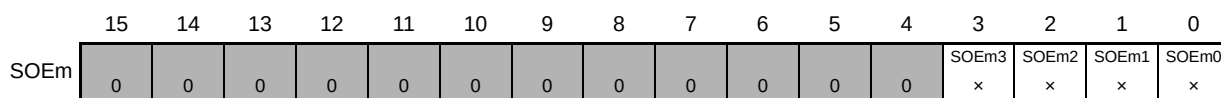
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOP)



(d) Serial output register m (SOM) ... Set only the bit of the target channel.



(e) Serial output enable register m (SOEm) ... This register is not used in this mode.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the simplified SPI (CSI) master reception mode, : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 36 Initial Setting Procedure for Master Reception

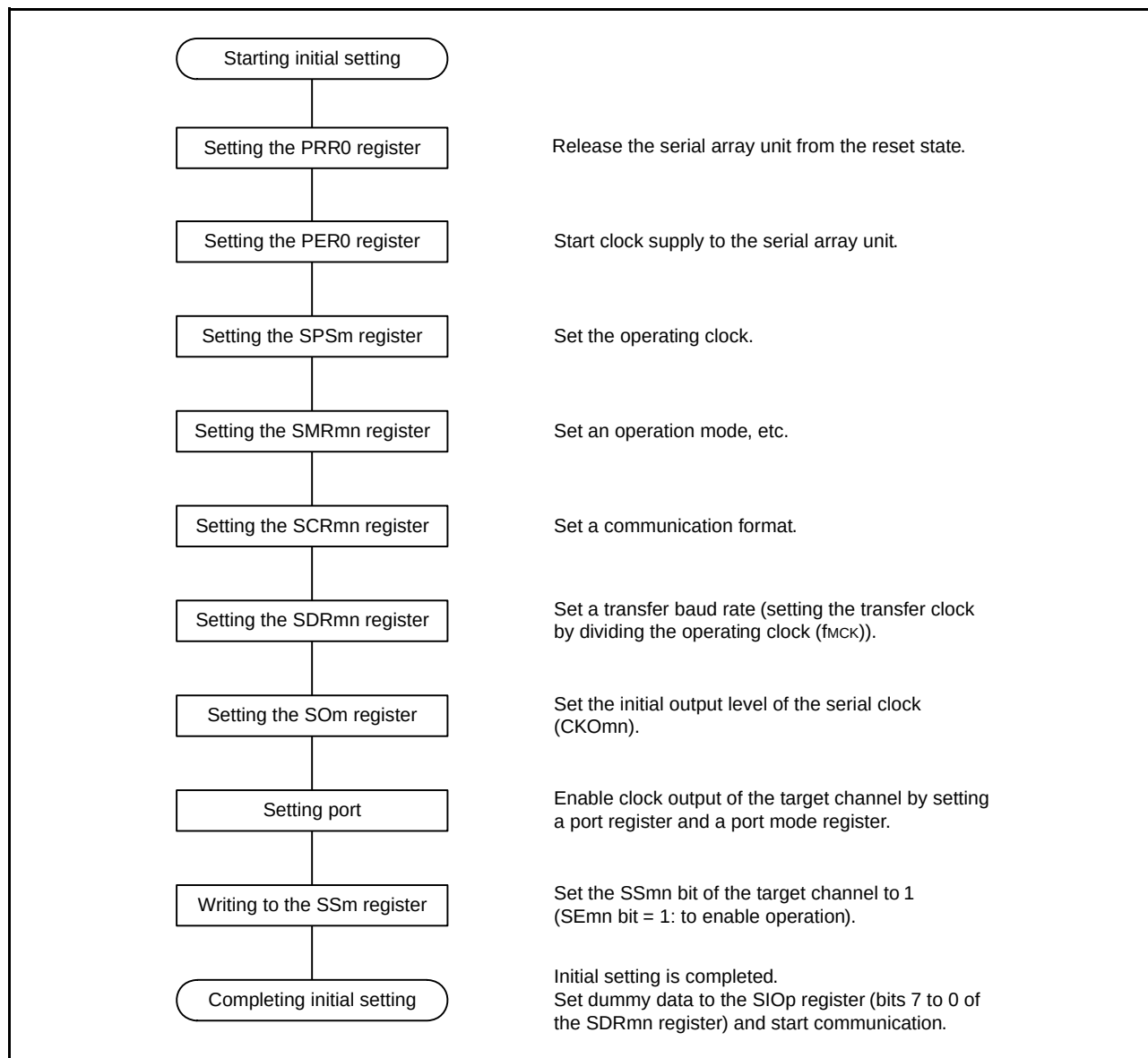


Figure 13 - 37 Procedure for Stopping Master Reception

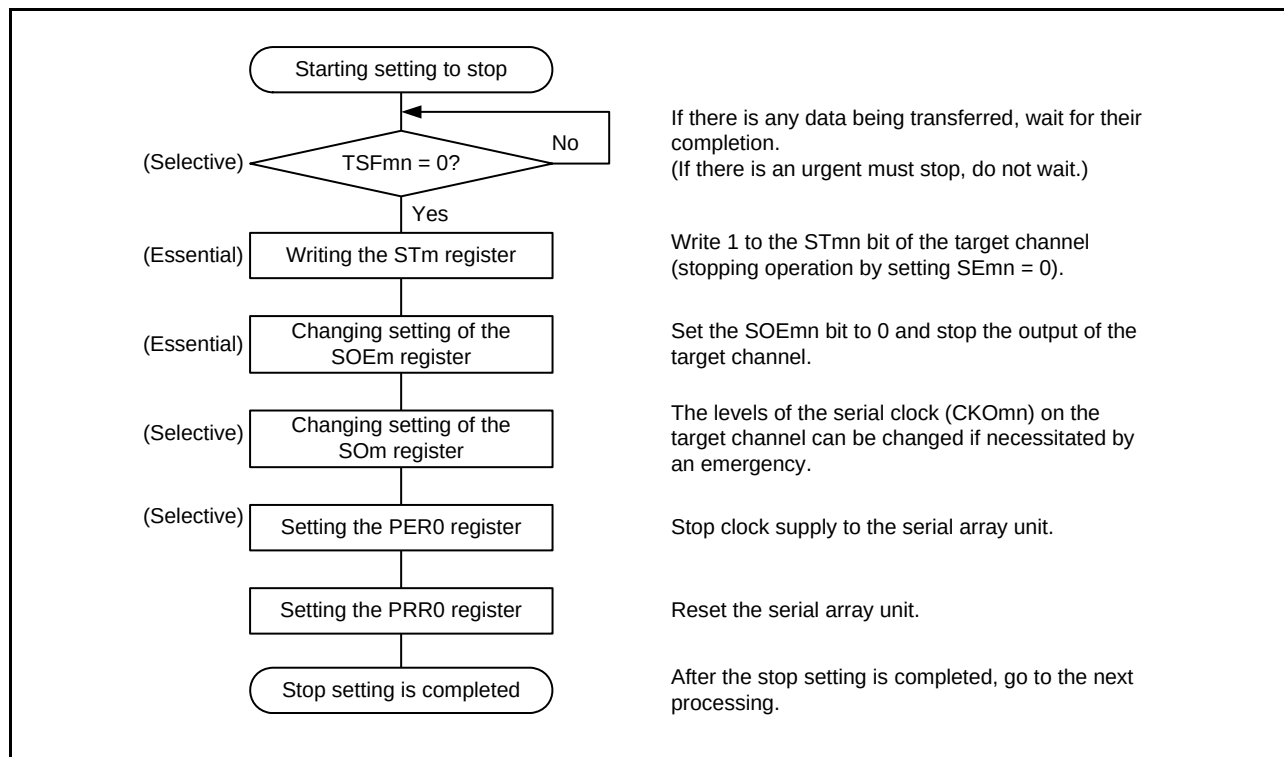
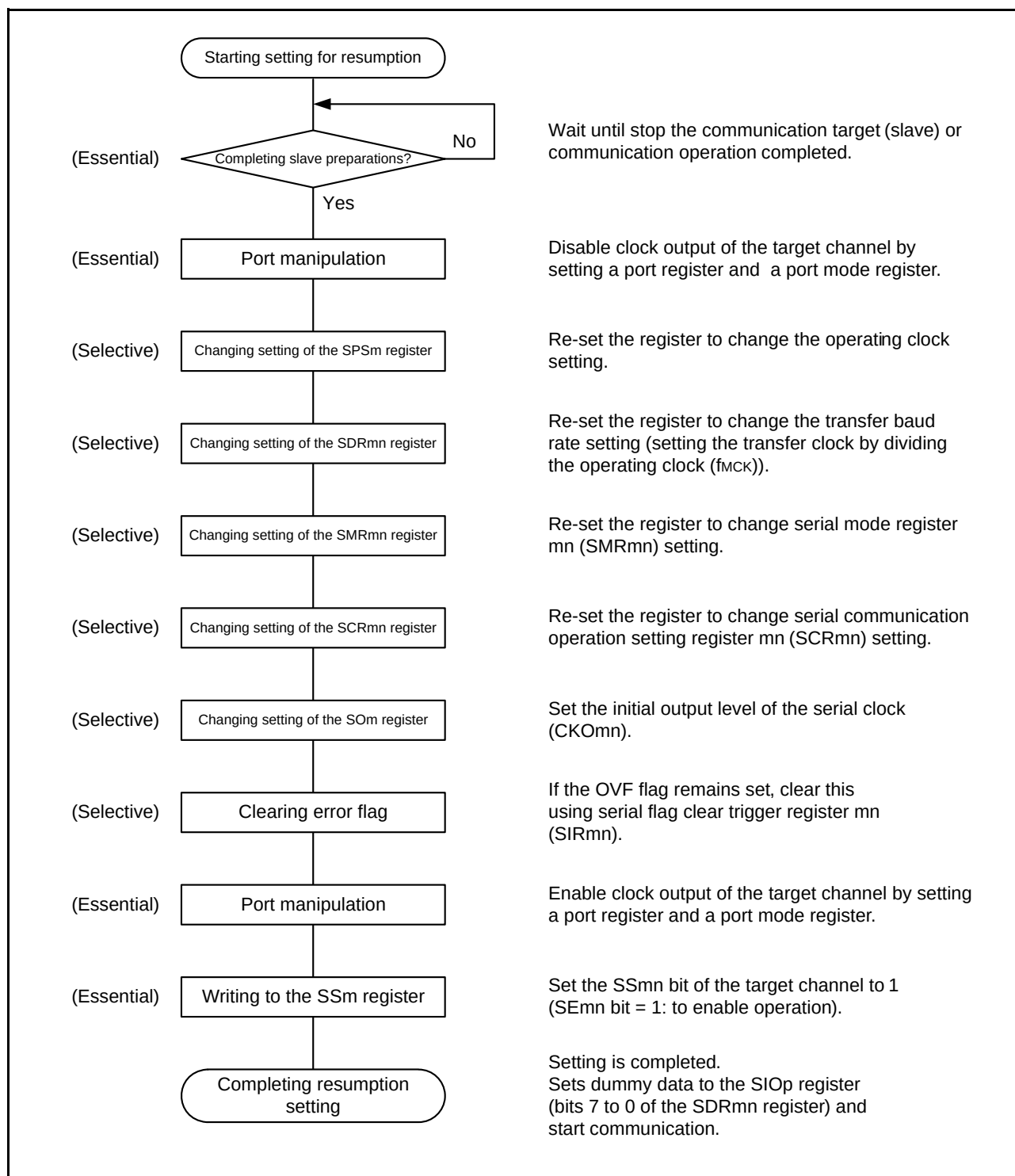


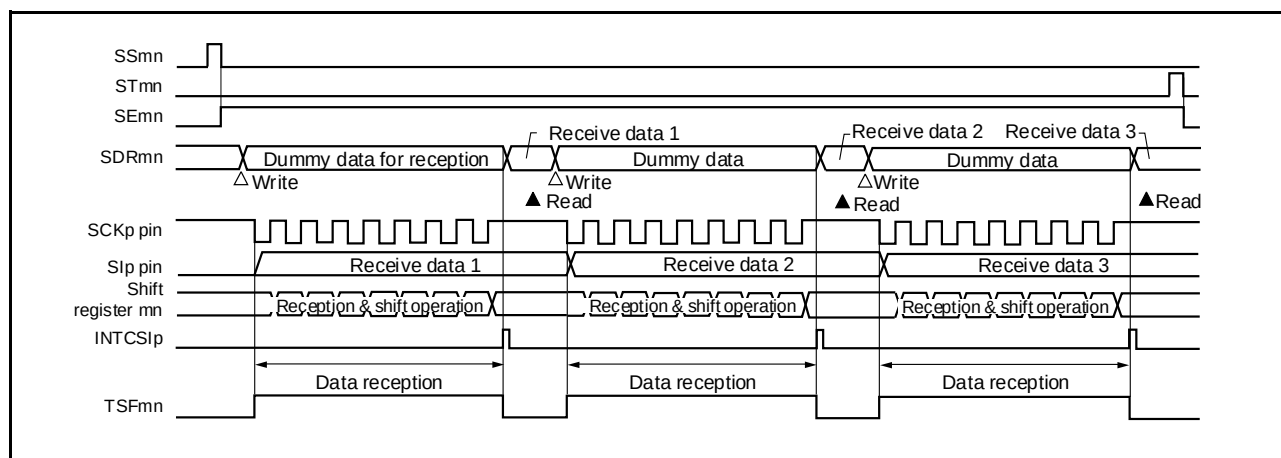
Figure 13 - 38 Procedure for Resuming Master Reception



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then proceed initialization instead of restarting the communication.

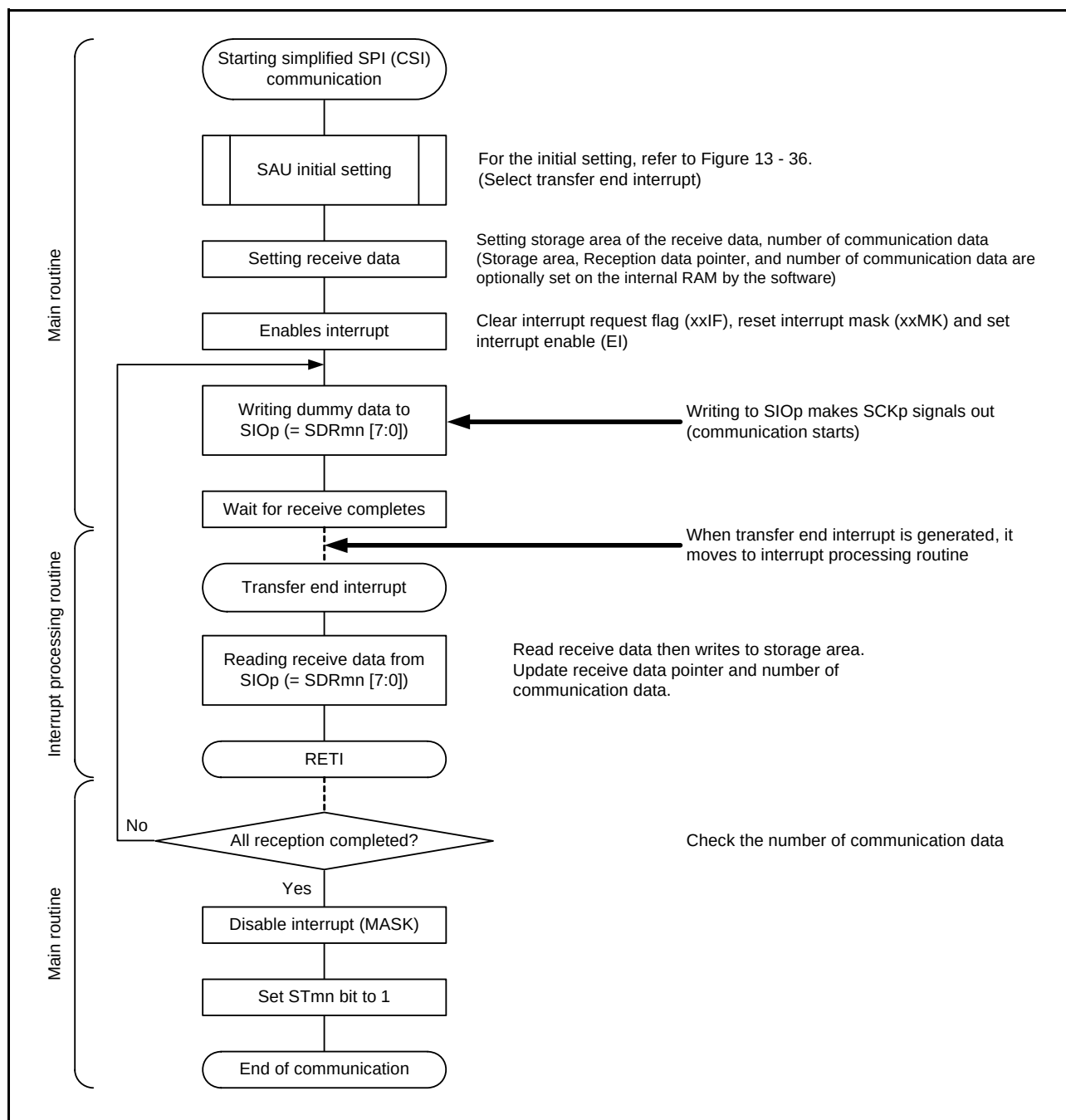
(3) Processing flow (in single-reception mode)

Figure 13 - 39 Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



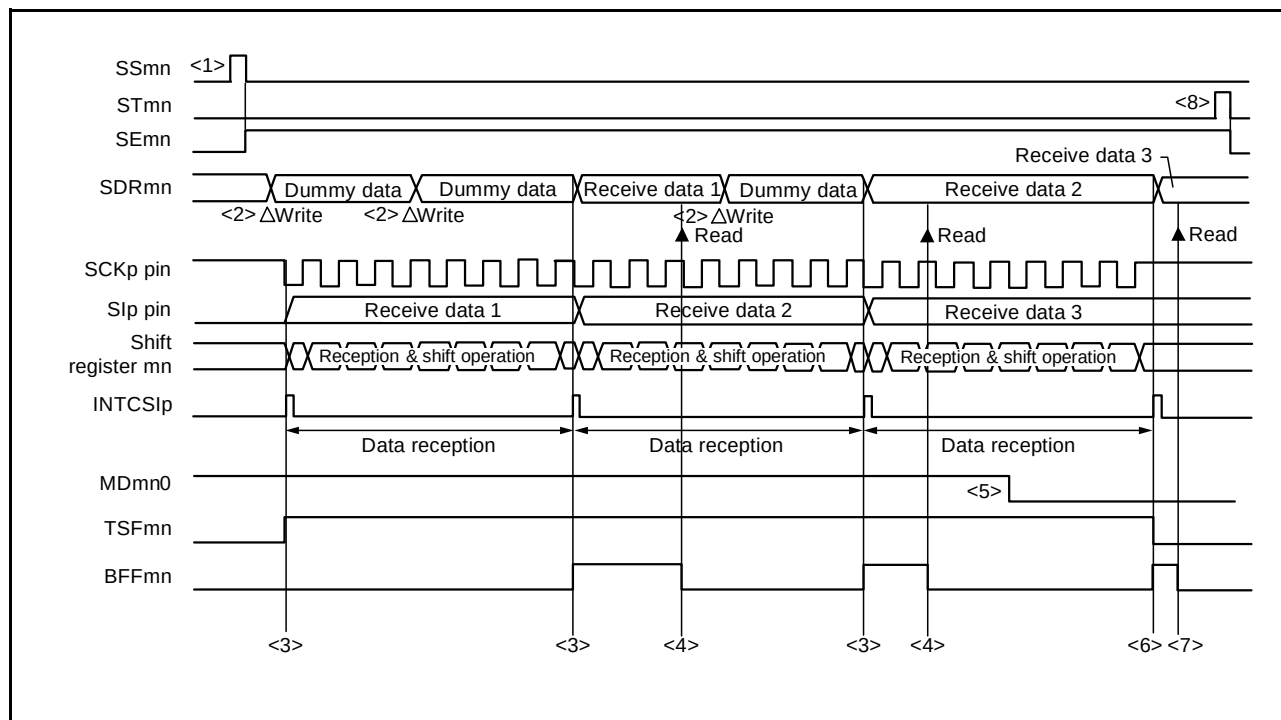
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 40 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 13 - 41 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



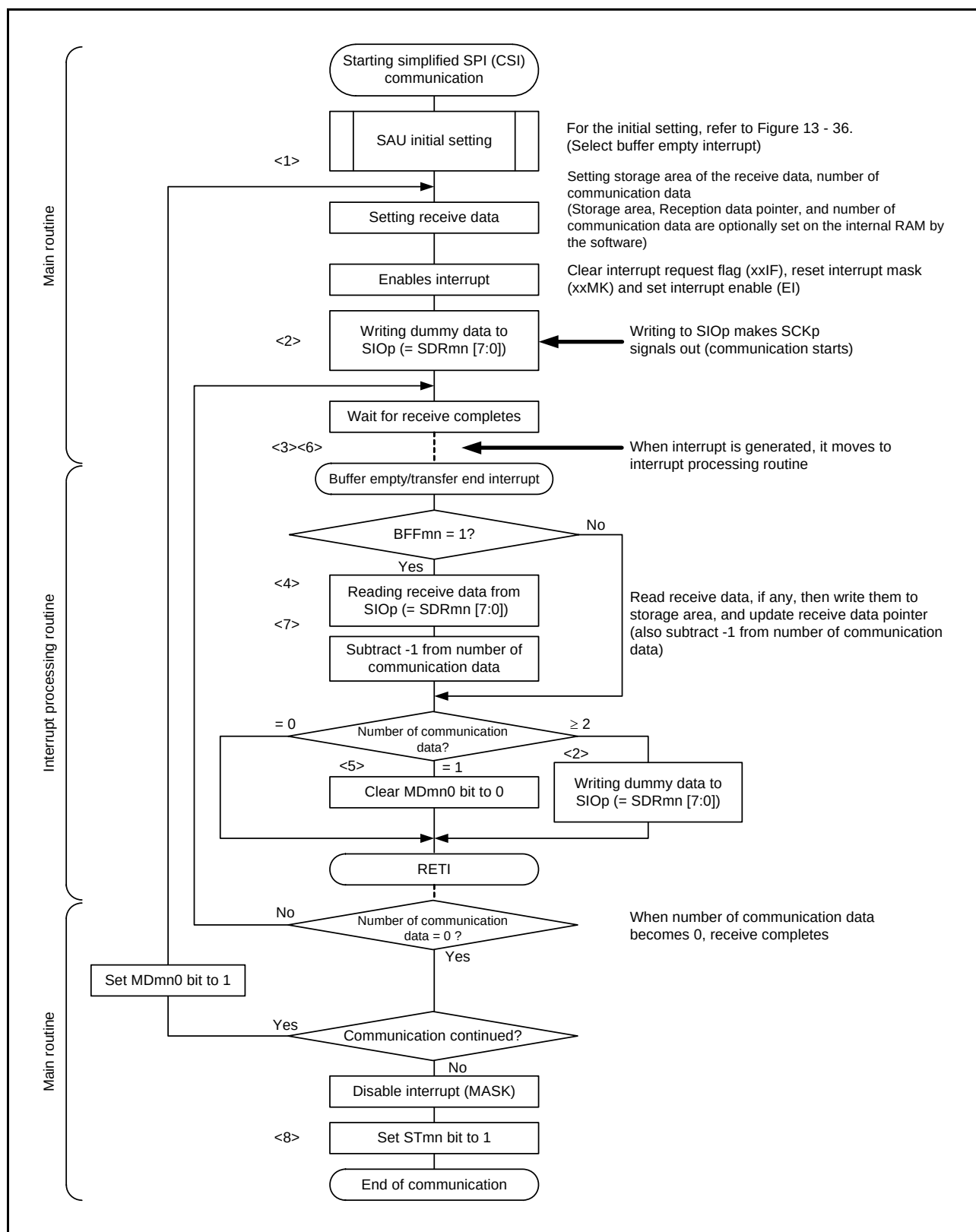
Caution The MDmn0 bit can be rewritten even during operation.

However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 13 - 42 Flowchart of Master Reception (in Continuous Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 42 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 41 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.3 Master Transmission/Reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	7 or 8 bits				
Transfer rate ^{Note}	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK} : System clock frequency				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

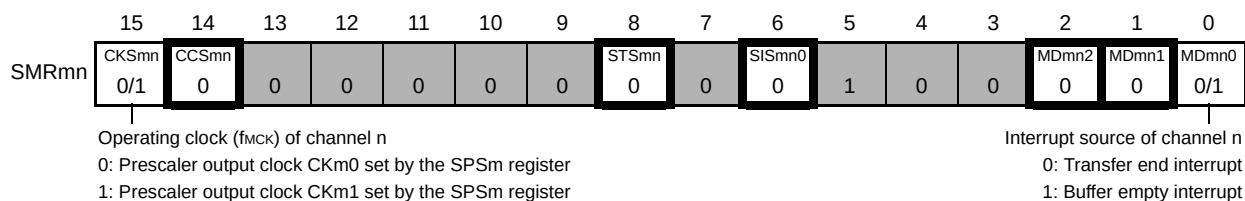
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for Ta = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

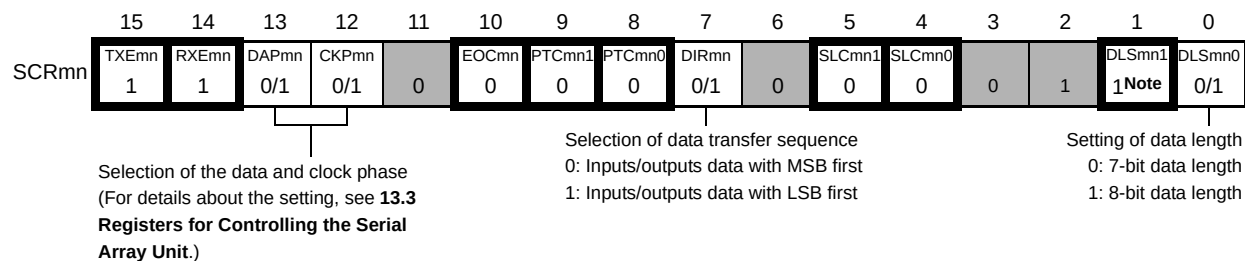
(1) Register setting

Figure 13 - 43 Example of Contents of Registers for Master Transmission/Reception of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

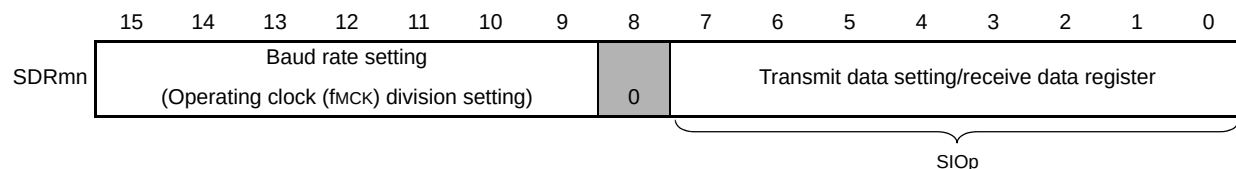
(a) Serial mode register mn (SMRmn)



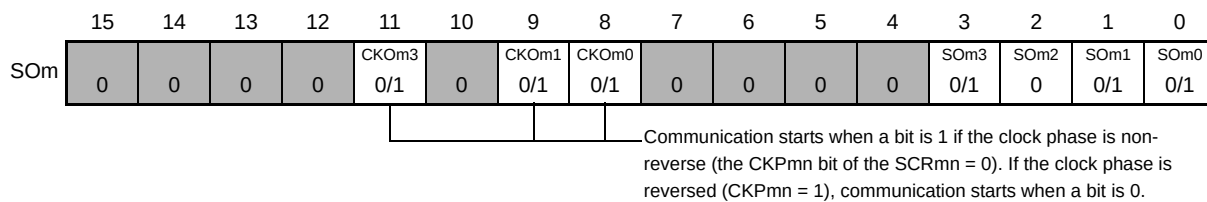
(b) Serial communication operation setting register mn (SCRmn)



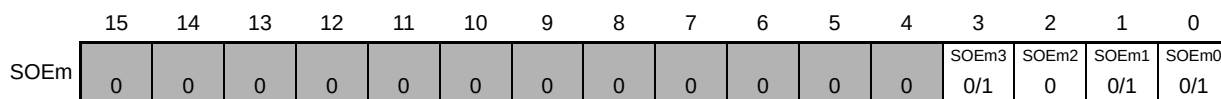
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOM) ... Set only the bit of the target channel.



(e) Serial output enable register m (SOEm) ... Set only the bit of the target channel to 1.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the simplified SPI (CSI) master transmission/reception mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 44 Initial Setting Procedure for Master Transmission/Reception

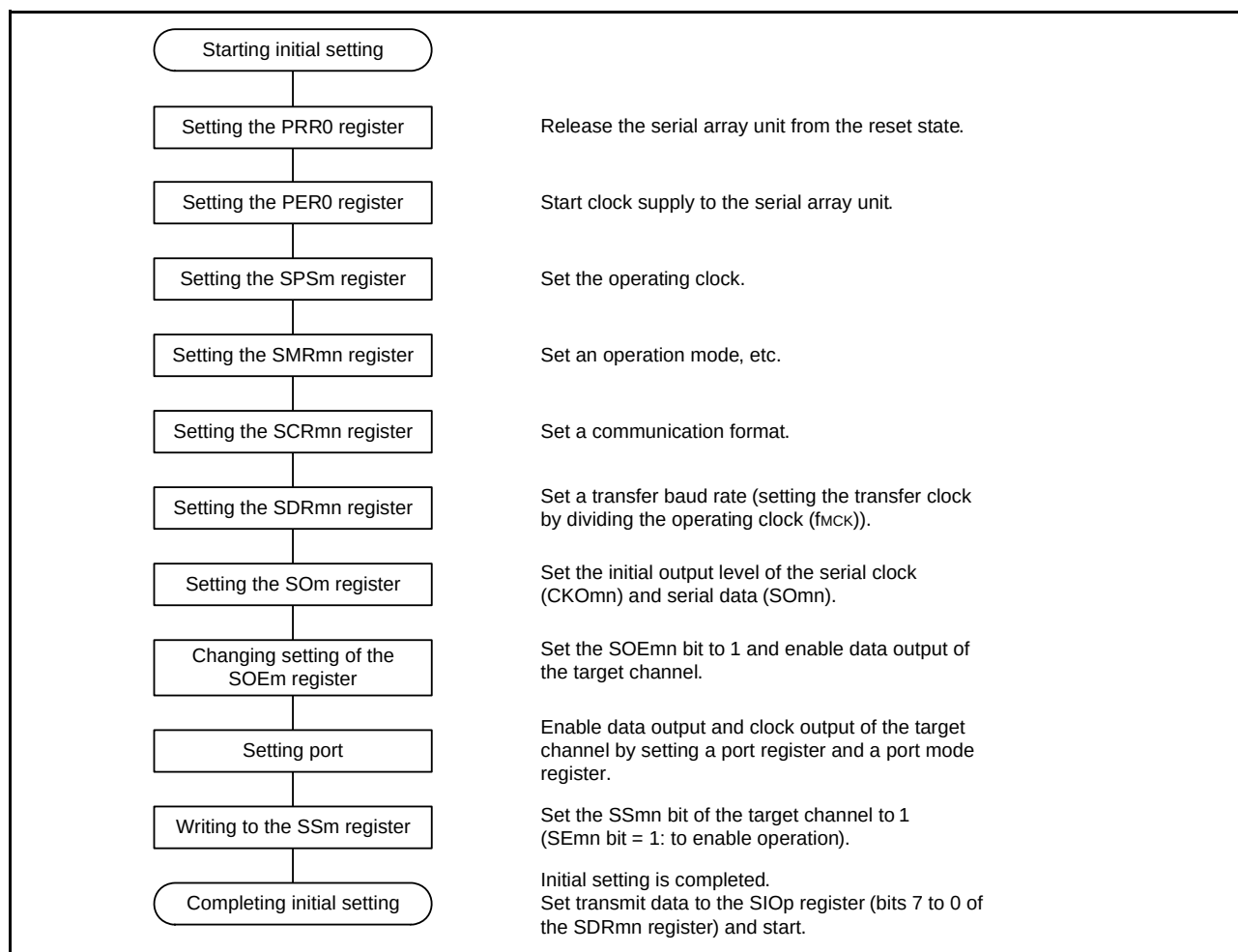


Figure 13 - 45 Procedure for Stopping Master Transmission/Reception

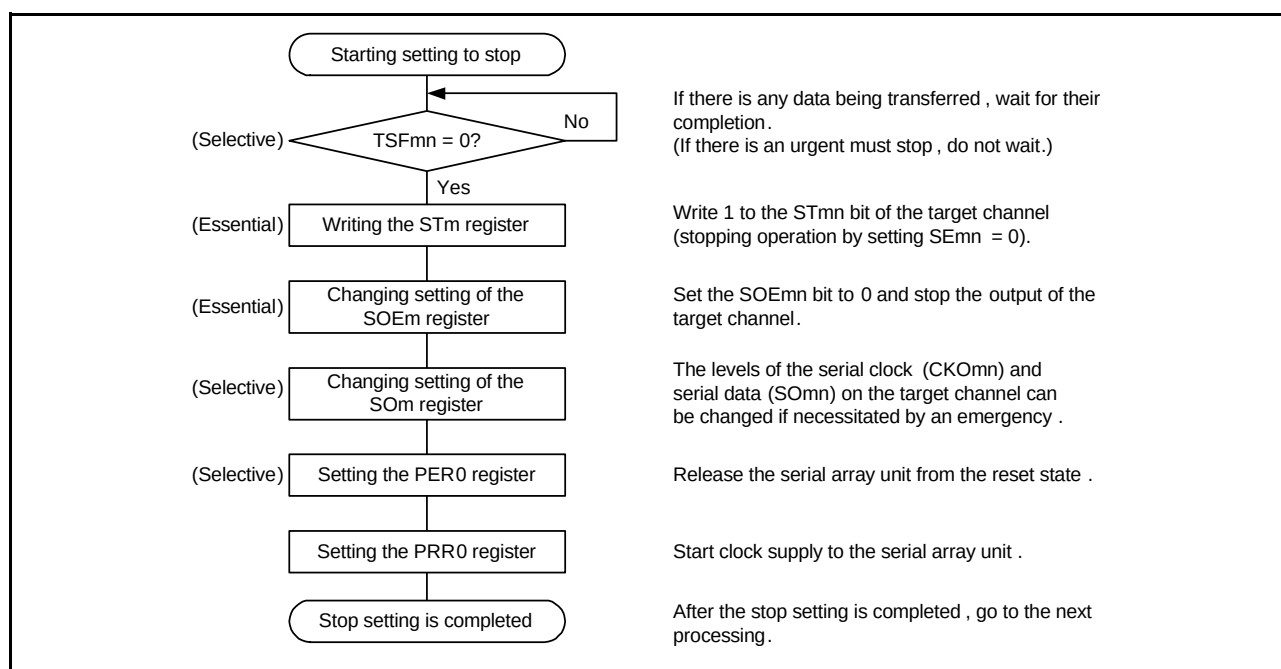
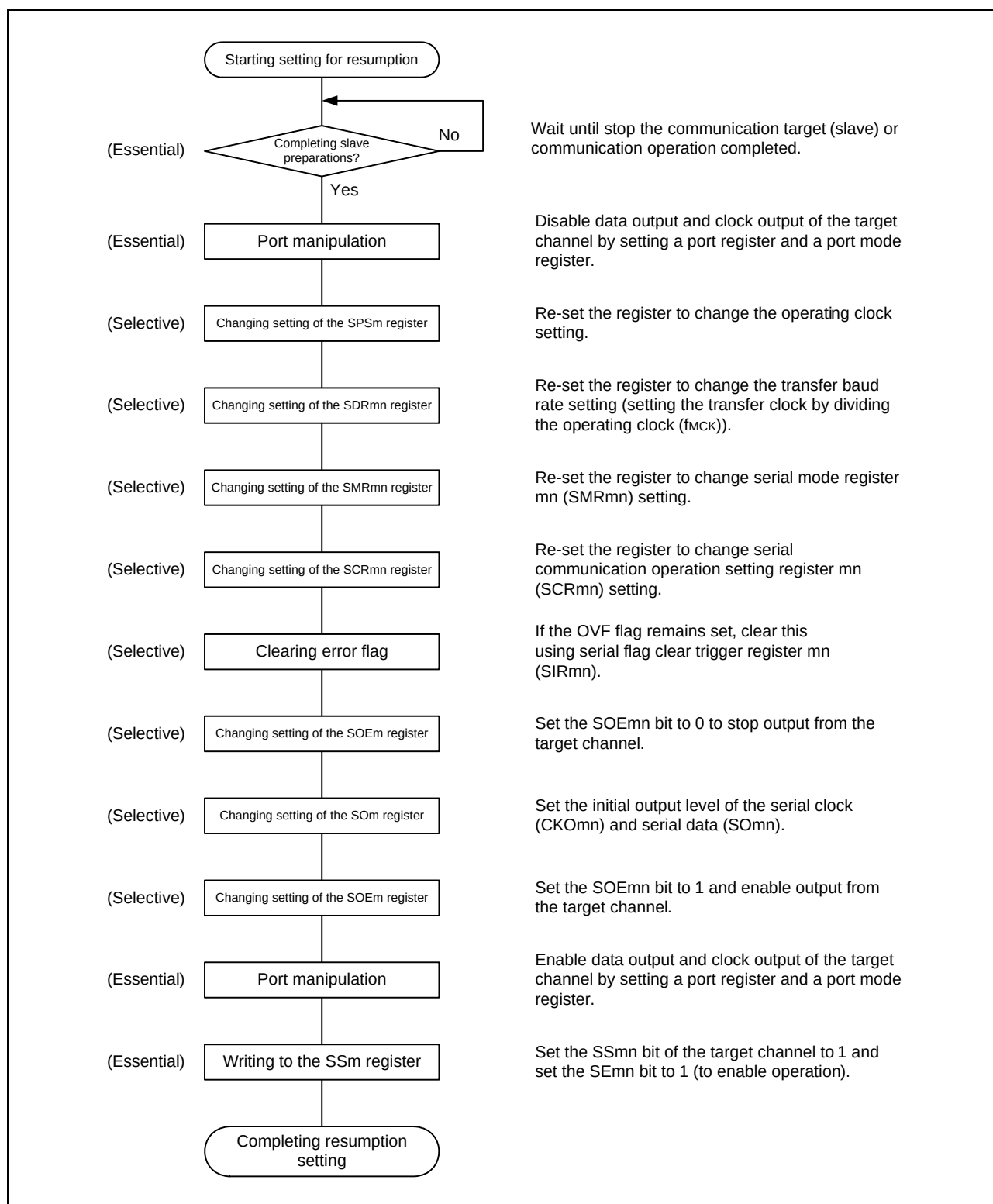
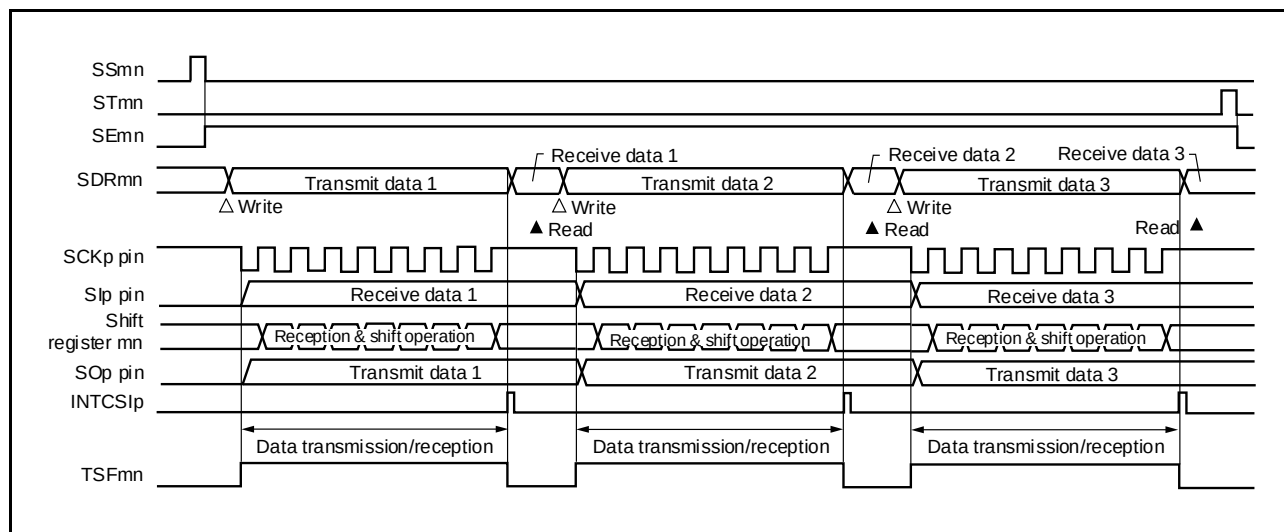


Figure 13 - 46 Procedure for Resuming Master Transmission/Reception



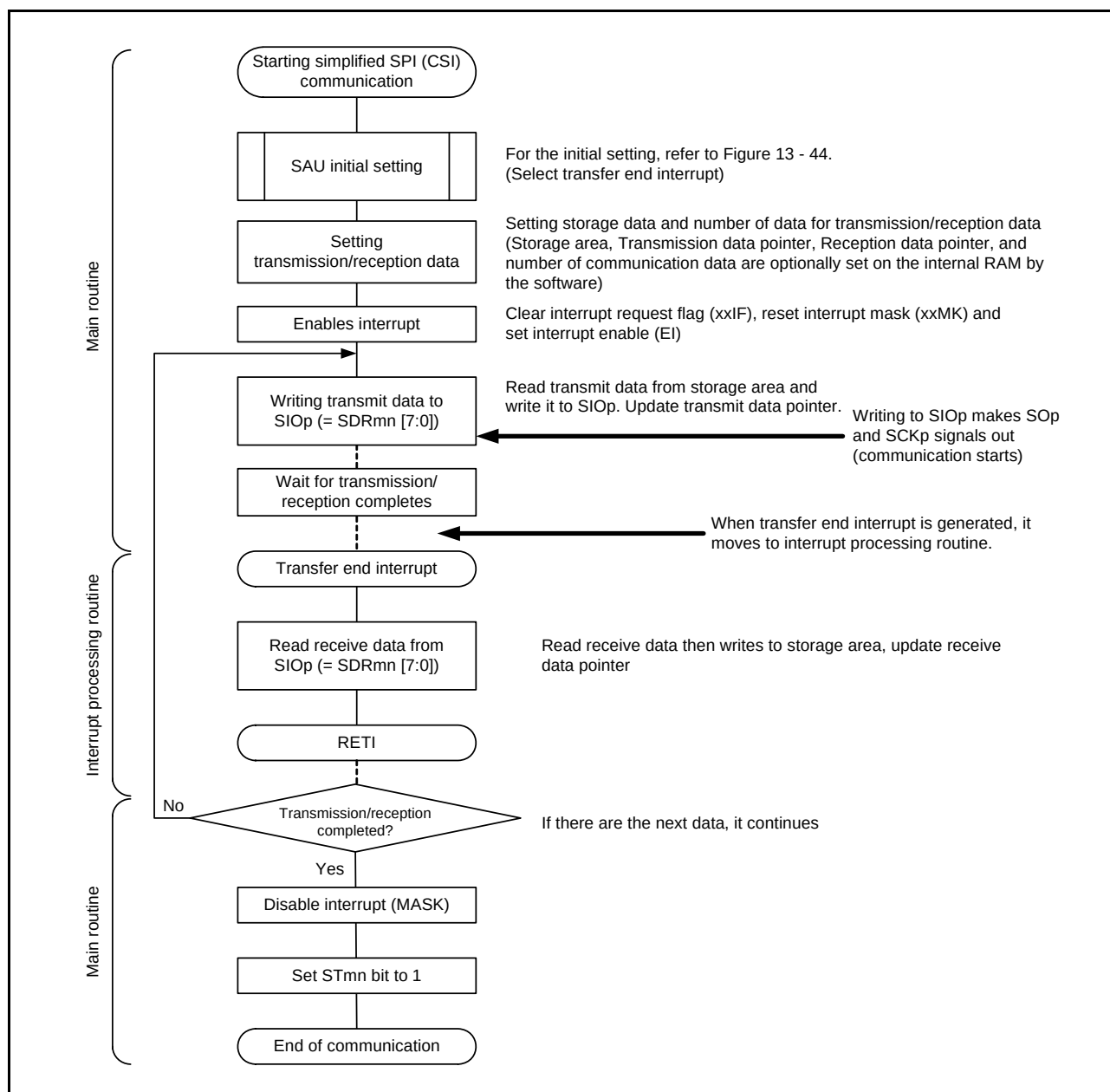
(3) Processing flow (in single-transmission/reception mode)

Figure 13 - 47 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



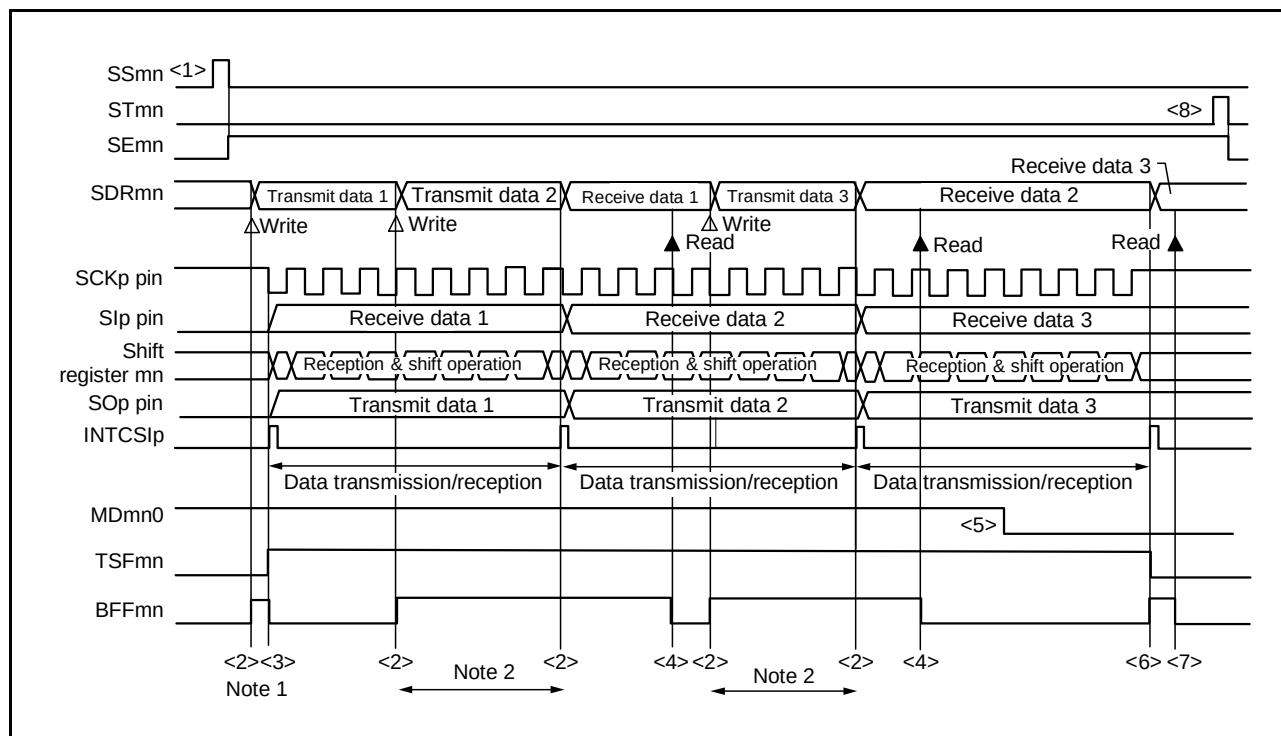
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 48 Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 13 - 49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

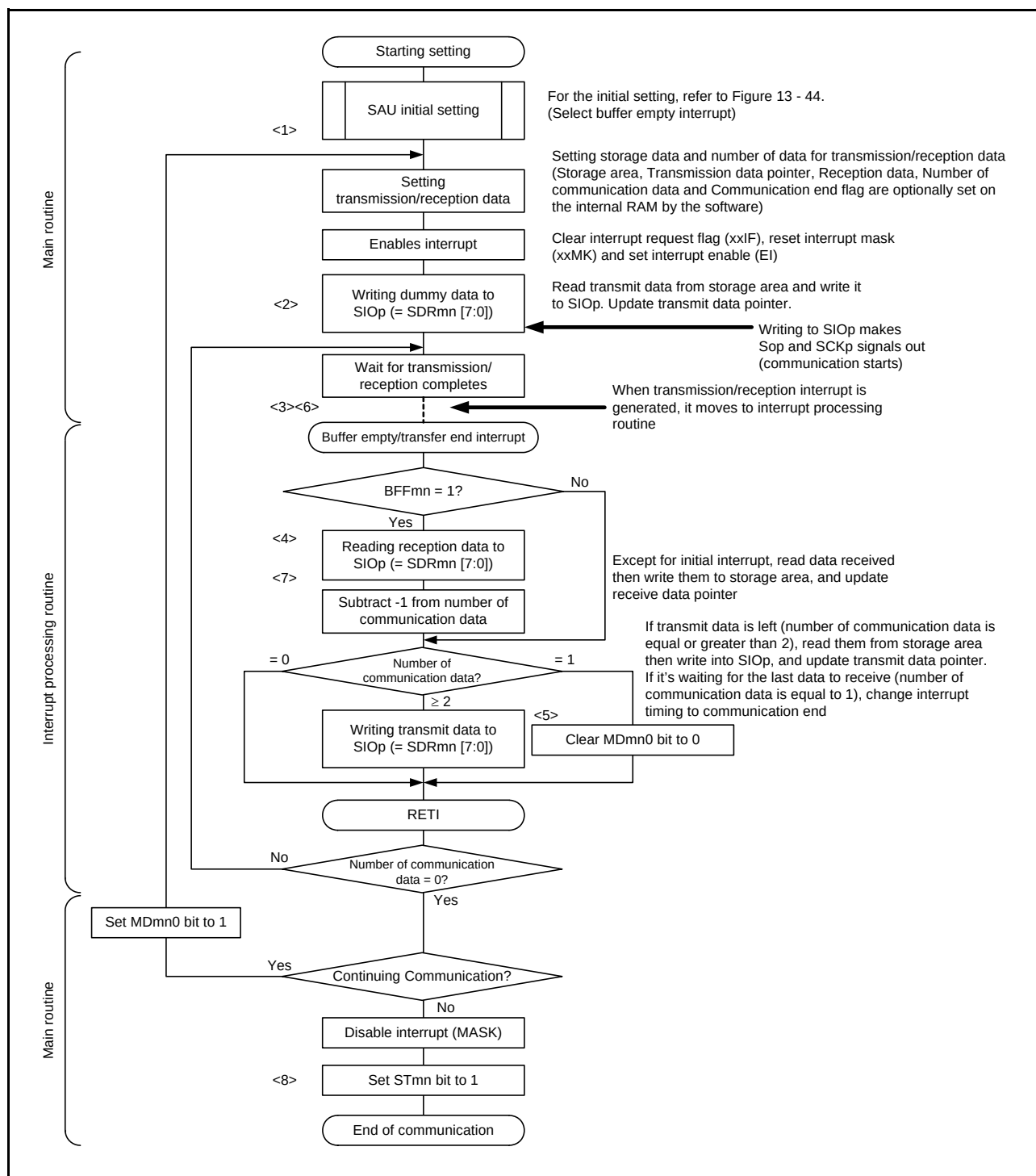
Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.4 Slave Transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	7 or 8 bits				
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Notes 1, 2}				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

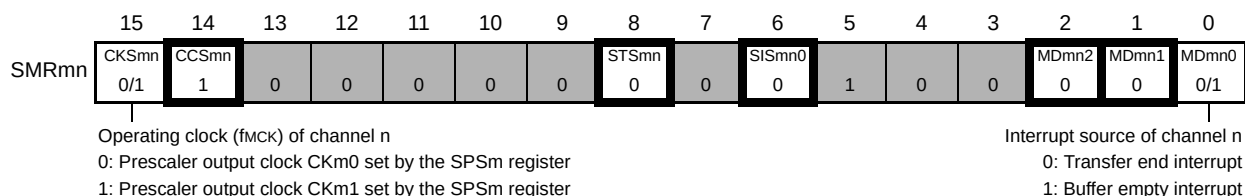
Remark 1. f_{MCK} : Operating clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

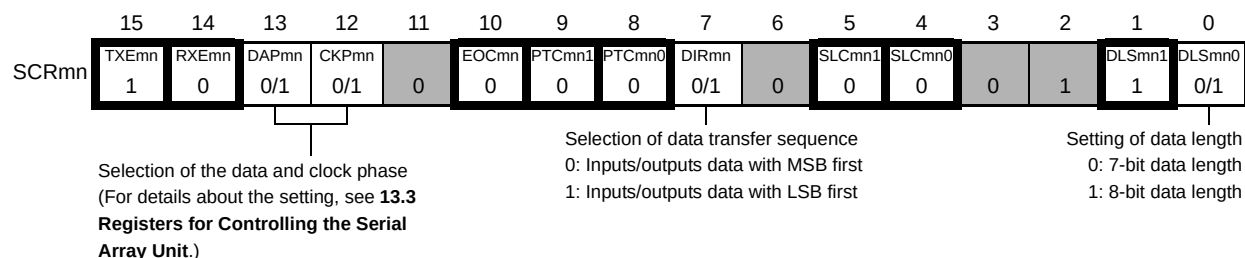
(1) Register setting

Figure 13 - 51 Example of Contents of Registers for Slave Transmission of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

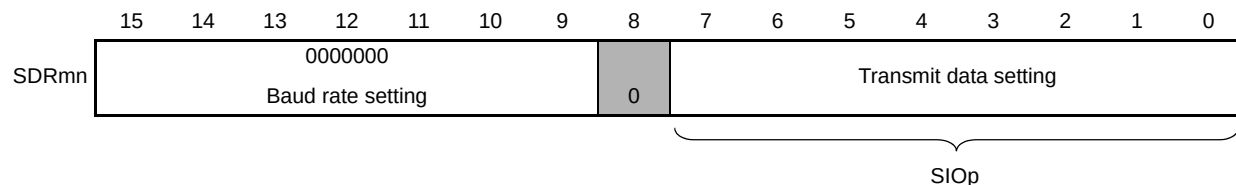
(a) Serial mode register mn (SMRmn)



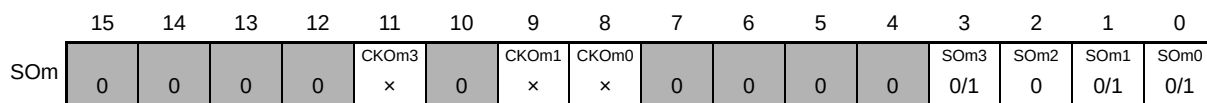
(b) Serial communication operation setting register mn (SCRmn)



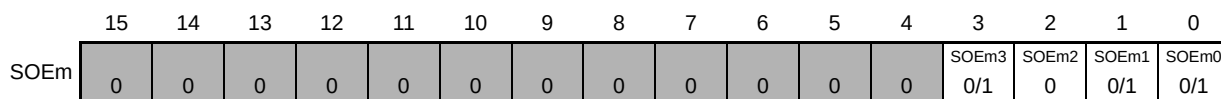
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOP)



(d) Serial output register m (SOM) ... Set only the bit of the target channel.



(e) Serial output enable register m (SOEm) ... Set only the bit of the target channel to 1.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00 to 03, 10 to 13

Remark 2. : Setting is fixed in the simplified SPI (CSI) slave transmission mode, : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 52 Initial Setting Procedure for Slave Transmission

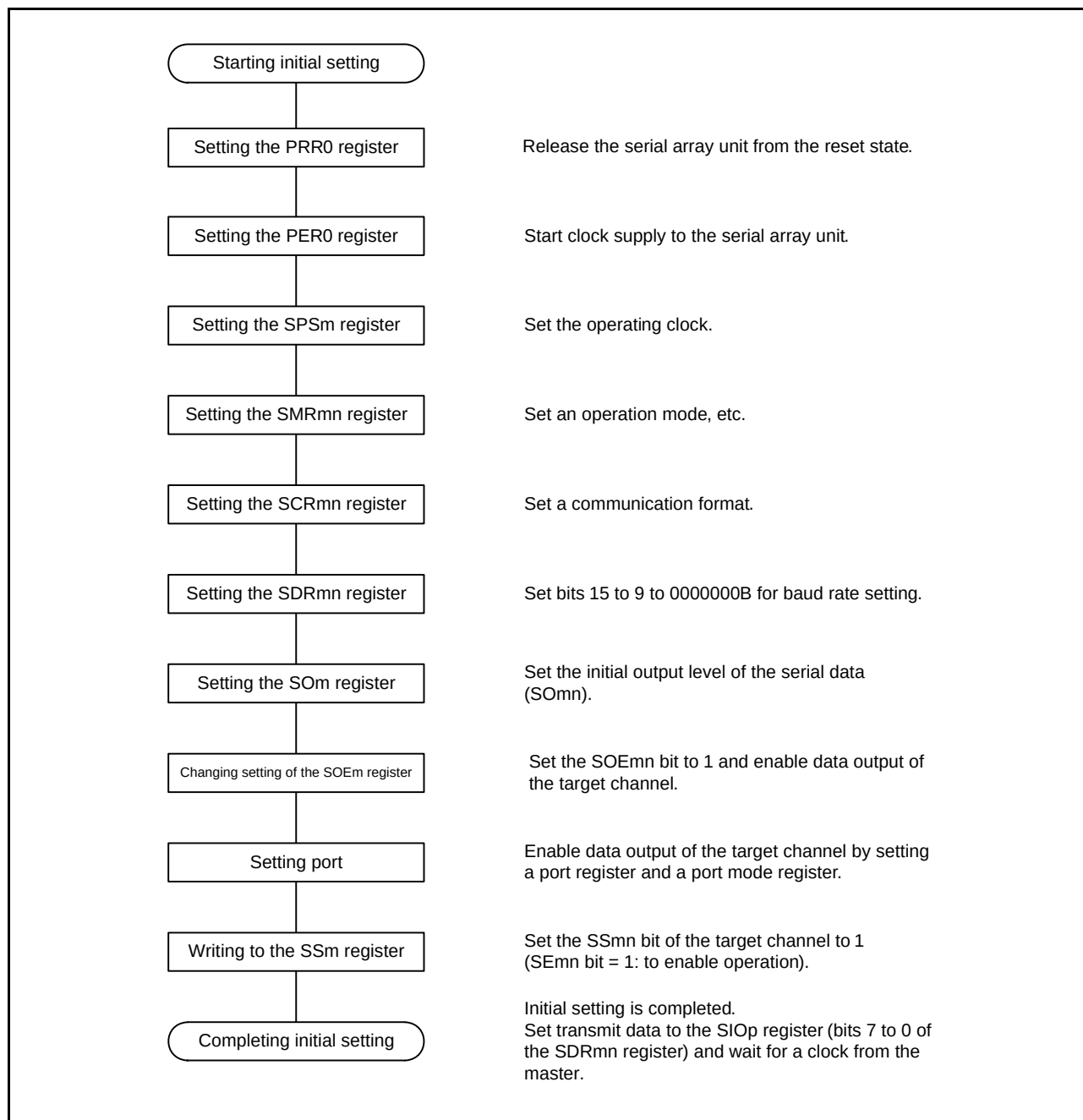


Figure 13 - 53 Procedure for Stopping Slave Transmission

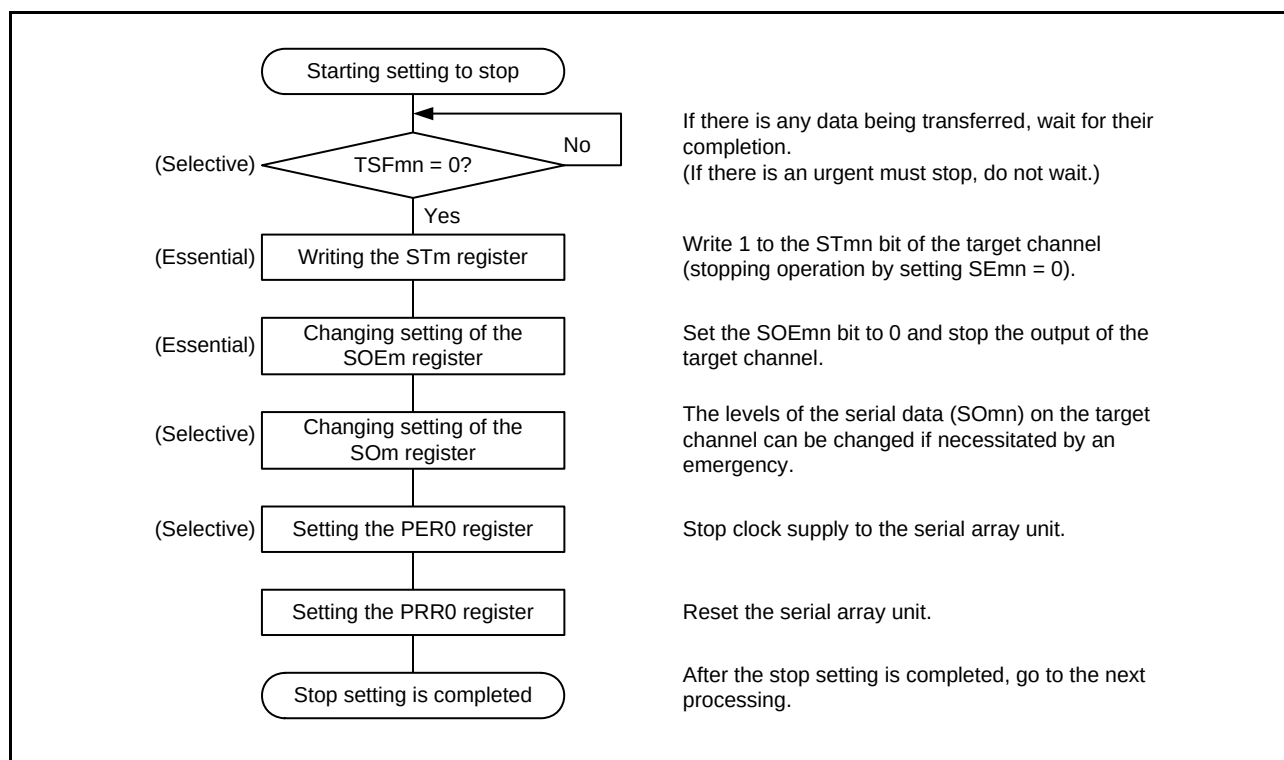
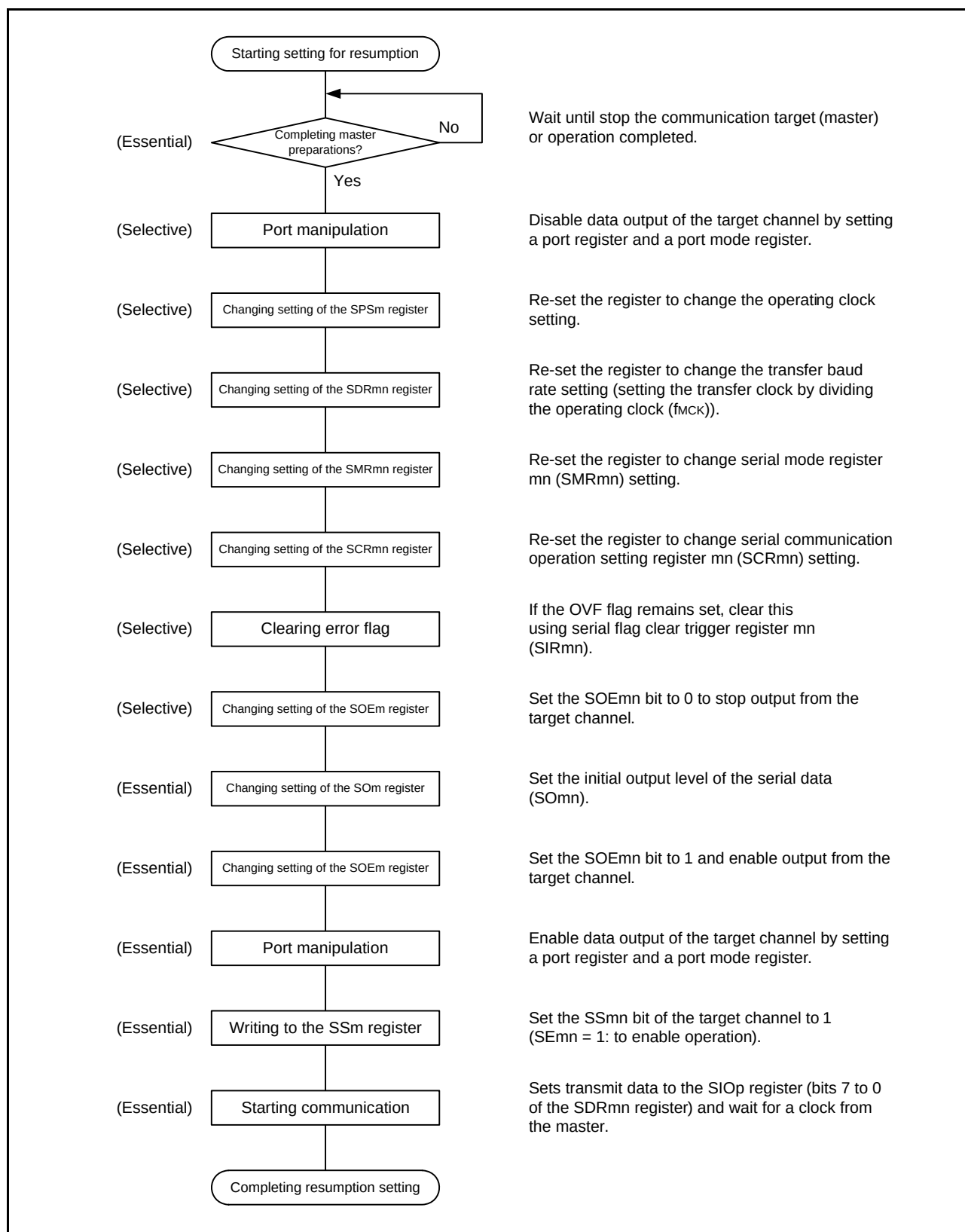


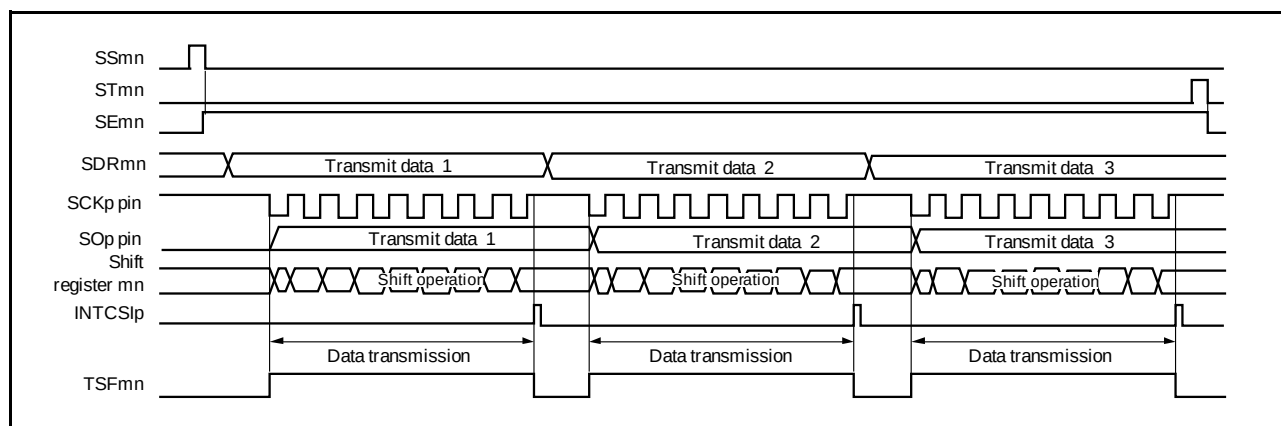
Figure 13 - 54 Procedure for Resuming Slave Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then proceed initialization instead of restarting the communication.

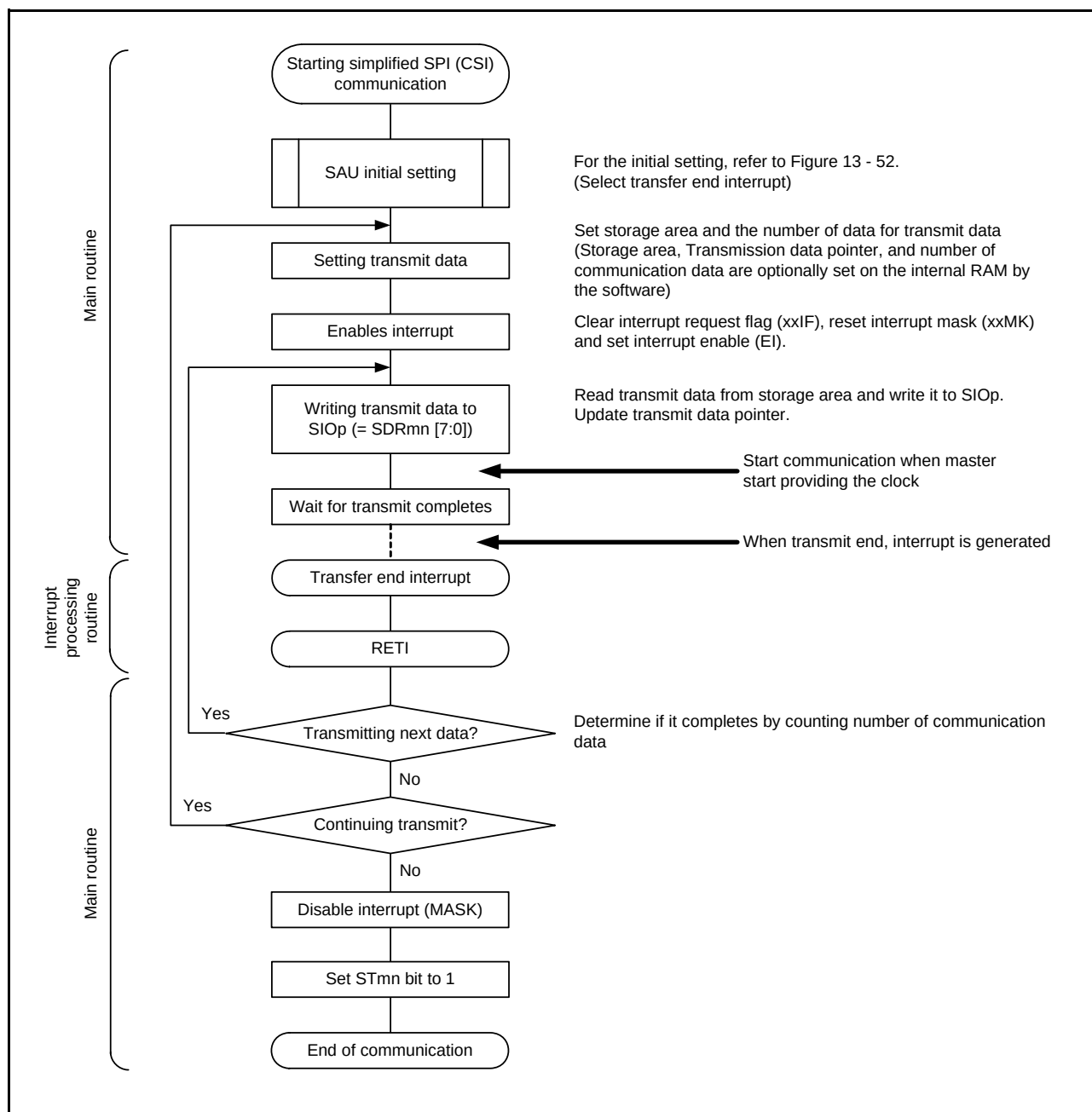
(3) Processing flow (in single-transmission mode)

Figure 13 - 55 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



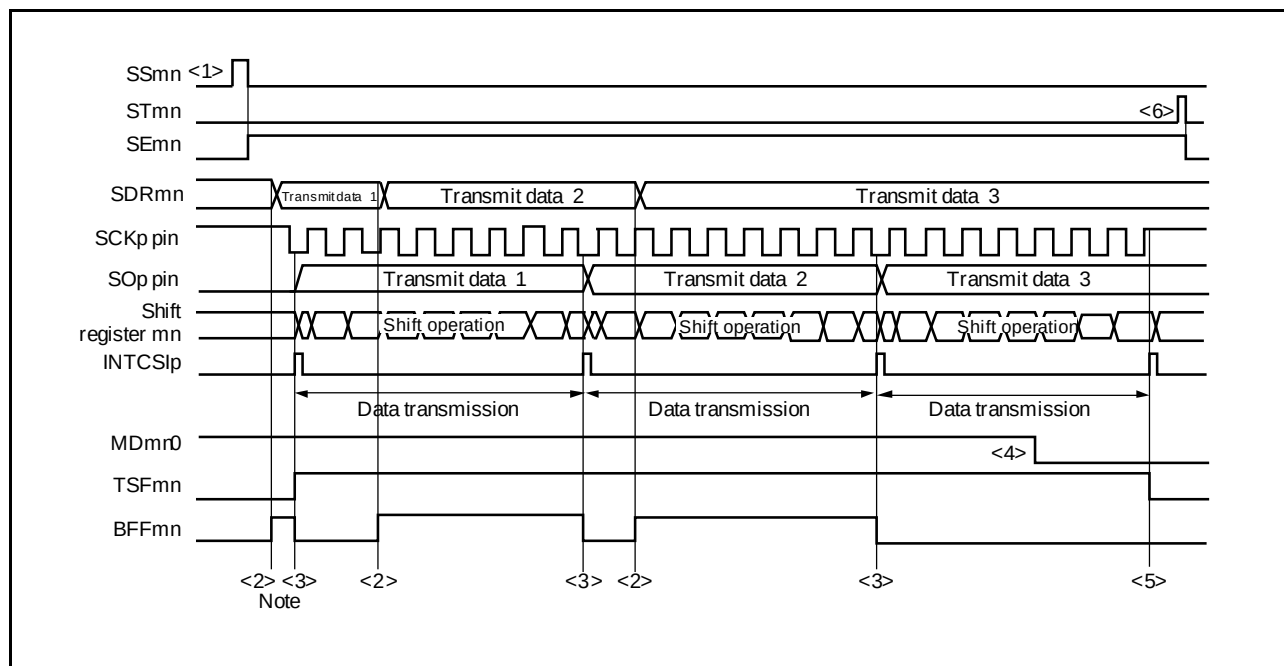
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 56 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

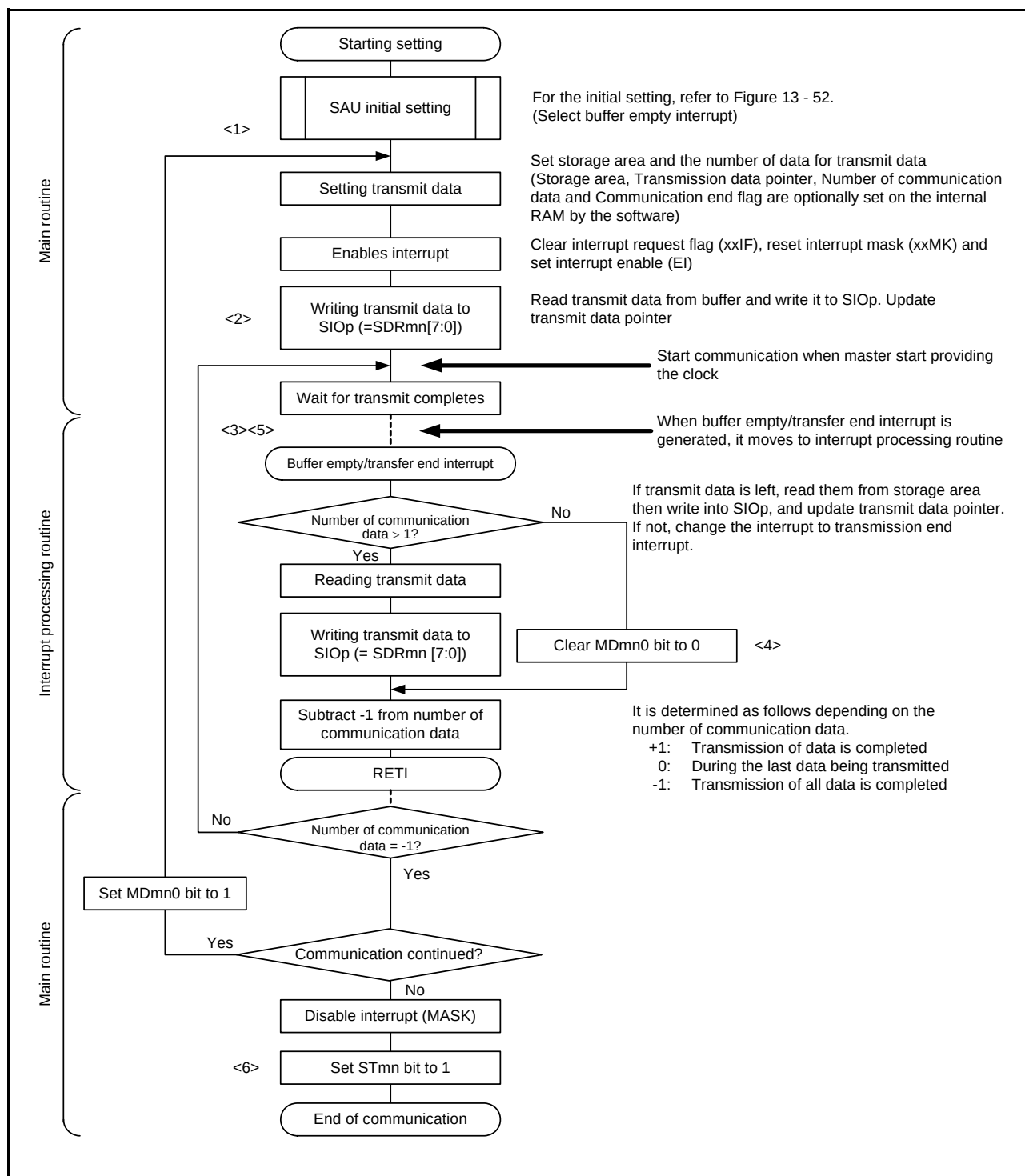


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 58 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.5 Slave Reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	7 or 8 bits				
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Notes 1, 2}				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

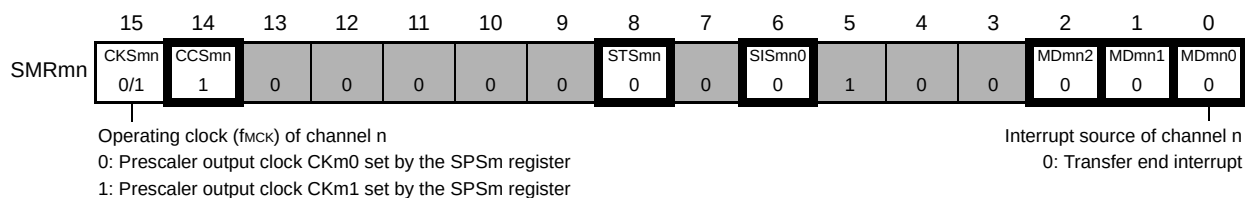
Remark 1. f_{MCK} : Operating clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

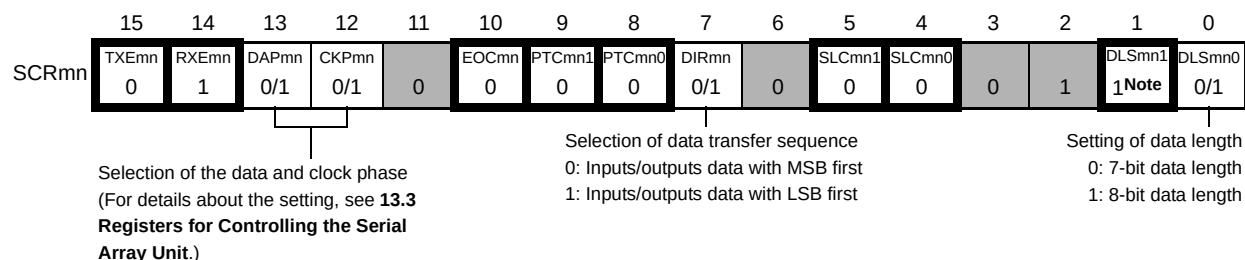
(1) Register setting

Figure 13 - 59 Example of Contents of Registers for Slave Reception of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

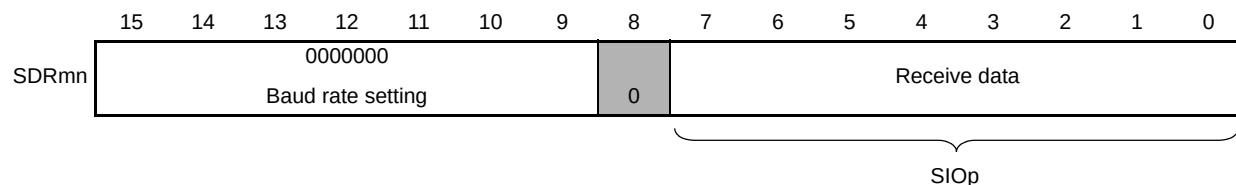
(a) Serial mode register mn (SMRmn)



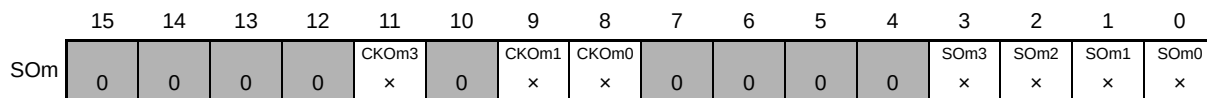
(b) Serial communication operation setting register mn (SCRmn)



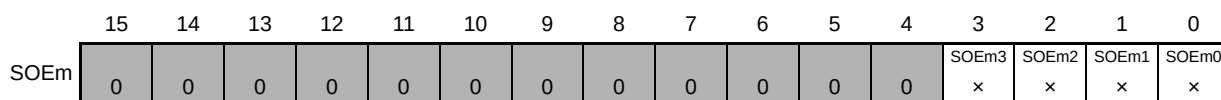
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOP)



(d) Serial output register m (SOM) ... This register is not used in this mode.



(e) Serial output enable register m (SOEm) ... This register is not used in this mode.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the slave reception mode, : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 60 Initial Setting Procedure for Slave Reception

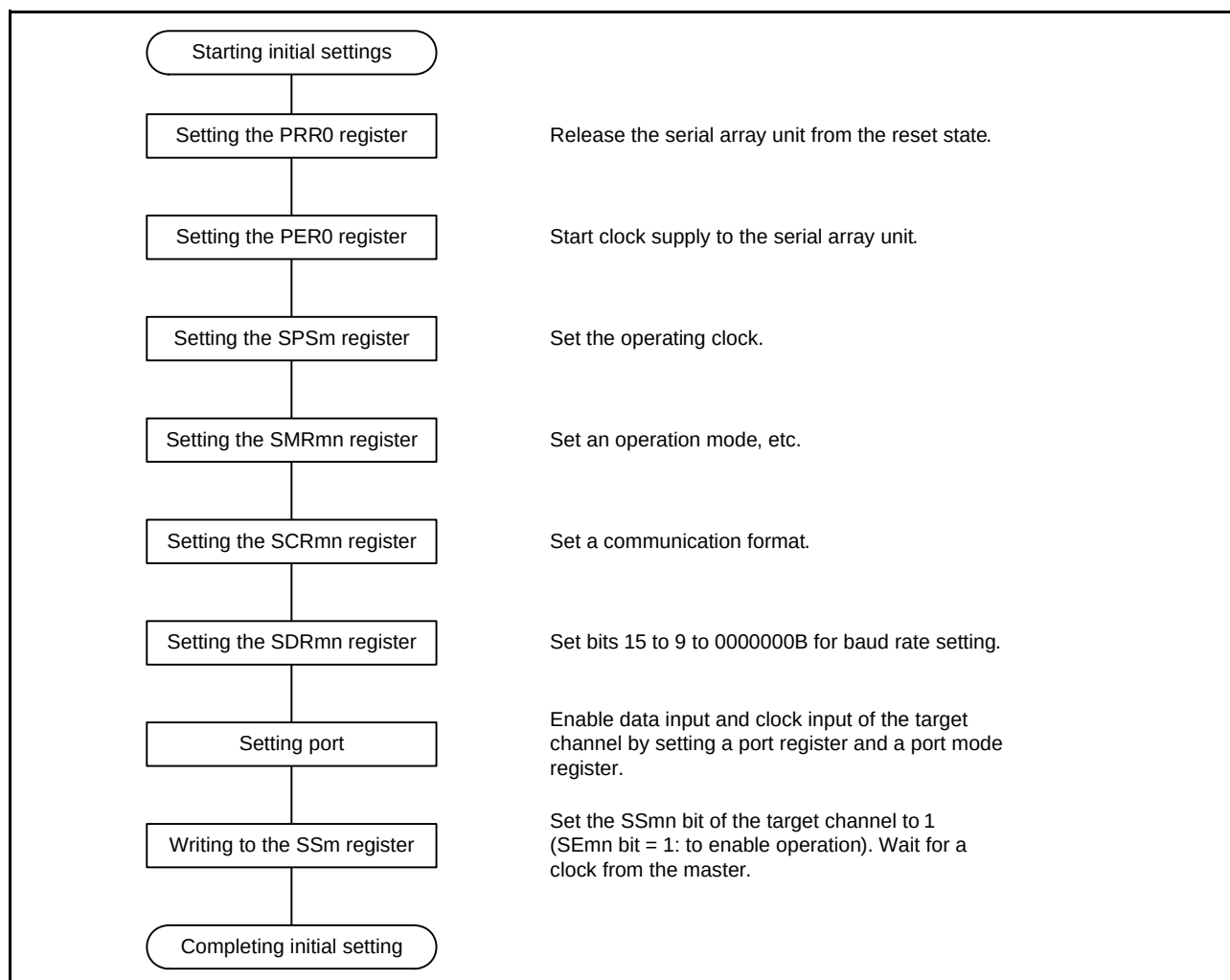


Figure 13 - 61 Procedure for Stopping Slave Reception

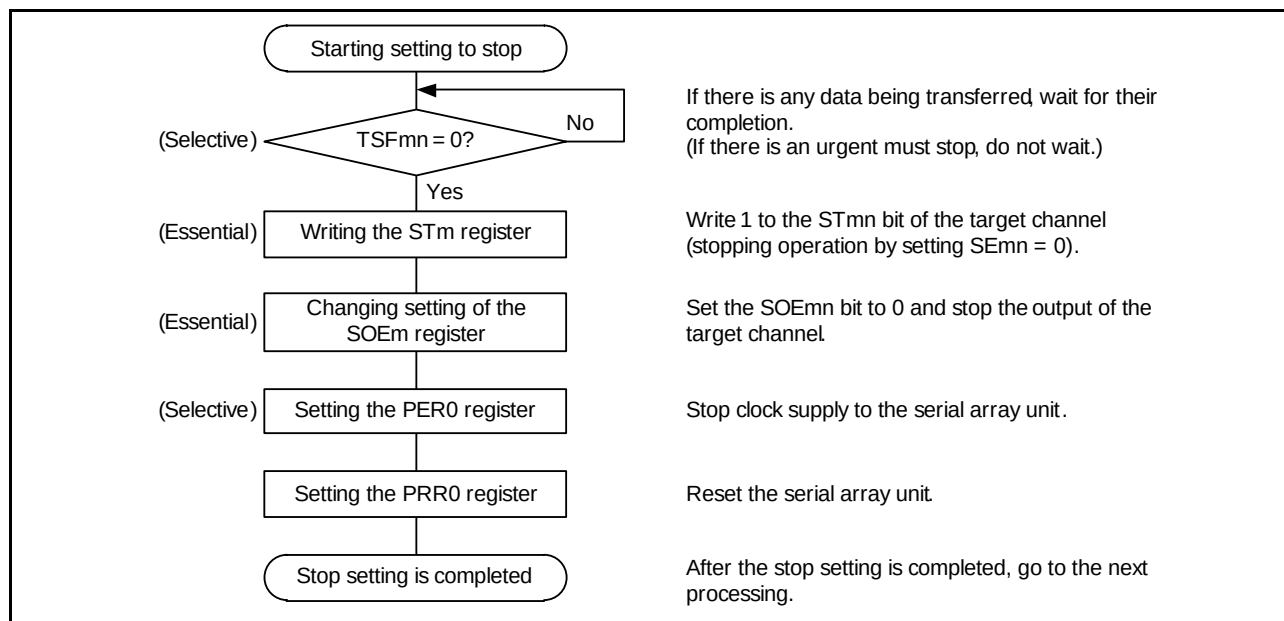
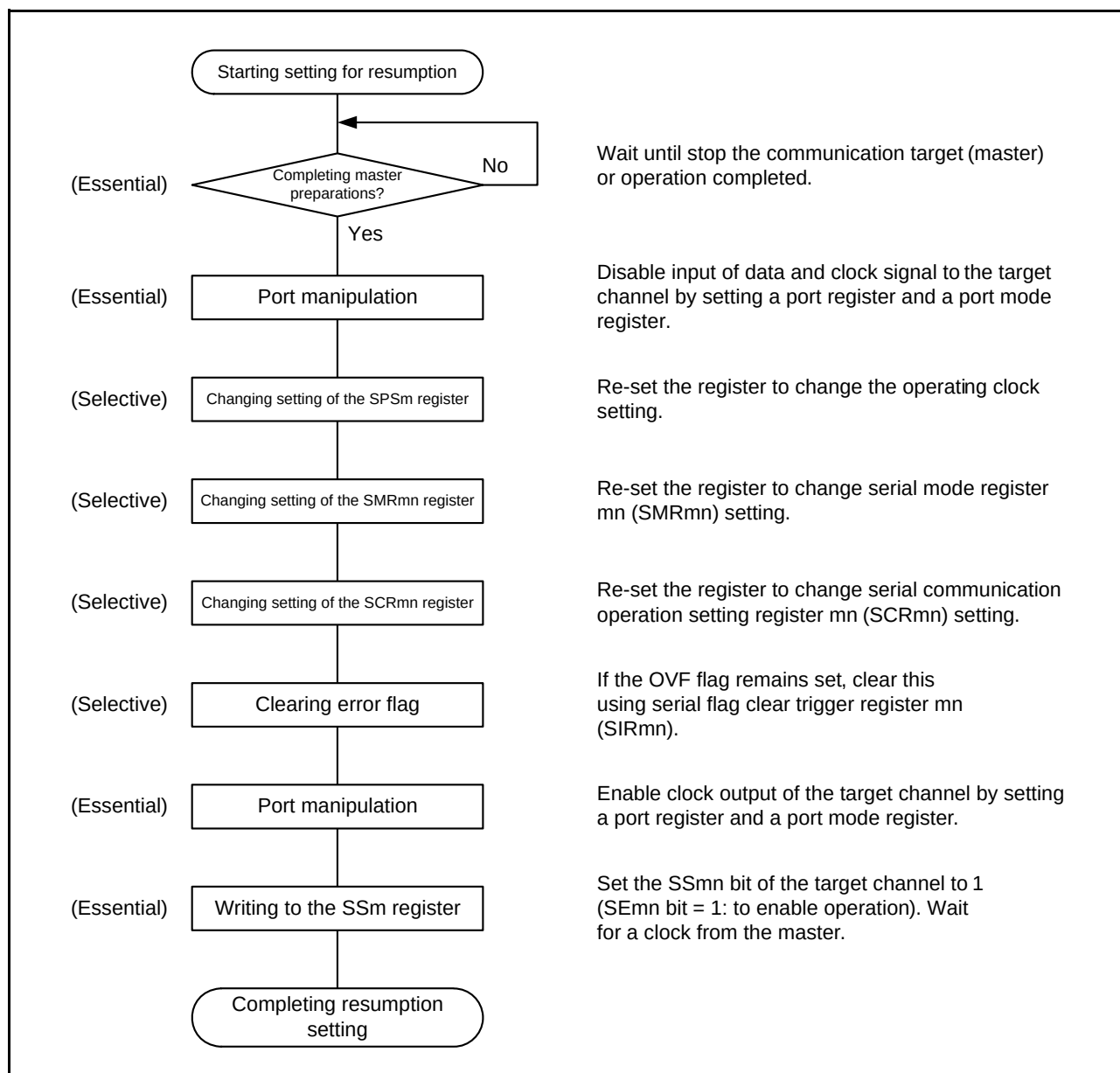


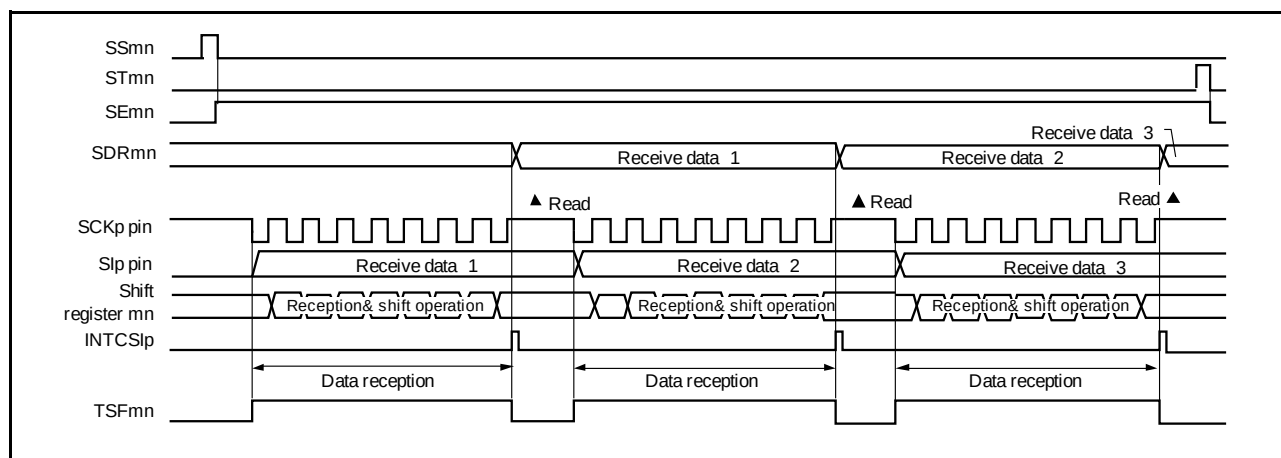
Figure 13 - 62 Procedure for Resuming Slave Reception



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then proceed initialization instead of restarting the communication.

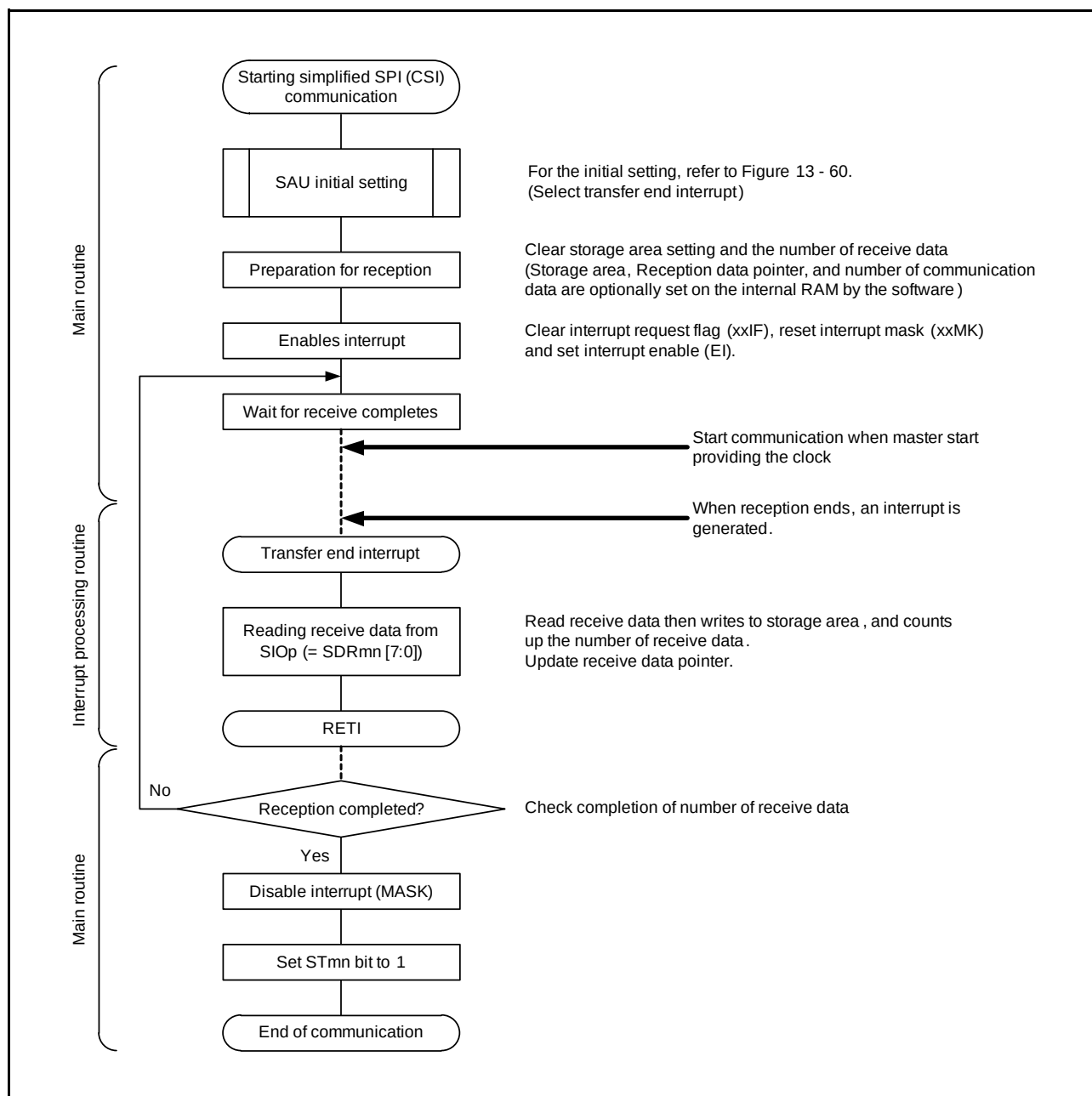
(3) Processing flow (in single-reception mode)

Figure 13 - 63 Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 64 Flowchart of Slave Reception (in Single-Reception Mode)



13.5.6 Slave Transmission/Reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	7 or 8 bits				
Transfer rate	Max. $f_{MCK}/6$ [Hz] ^{Notes 1, 2}				
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.				
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse				
Data direction	MSB or LSB first				

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

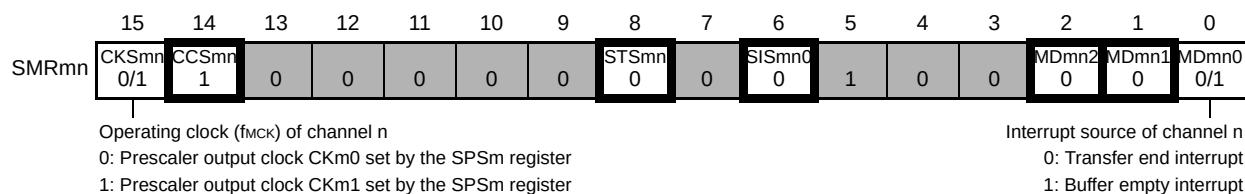
Remark 1. f_{MCK} : Operating clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

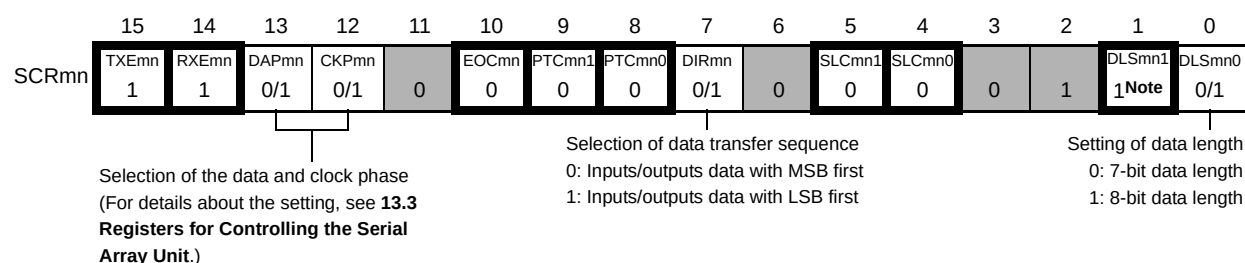
(1) Register setting

Figure 13 - 65 Example of Contents of Registers for Slave Transmission/Reception of simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21)

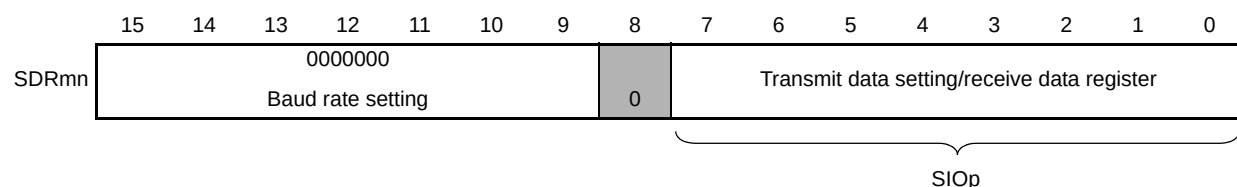
(a) Serial mode register mn (SMRmn)



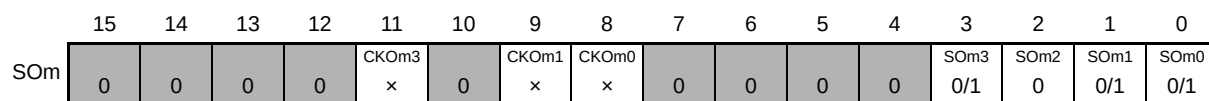
(b) Serial communication operation setting register mn (SCRmn)



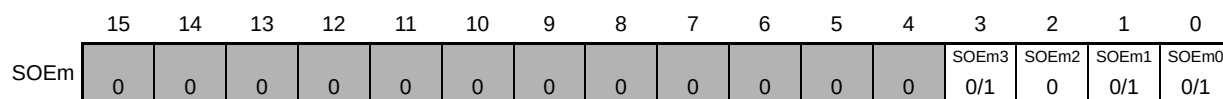
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOP)



(d) Serial output register m (SOM) ... Set only the bit of the target channel.



(e) Serial output enable register m (SOEm) ... Set only the bit of the target channel to 1.



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

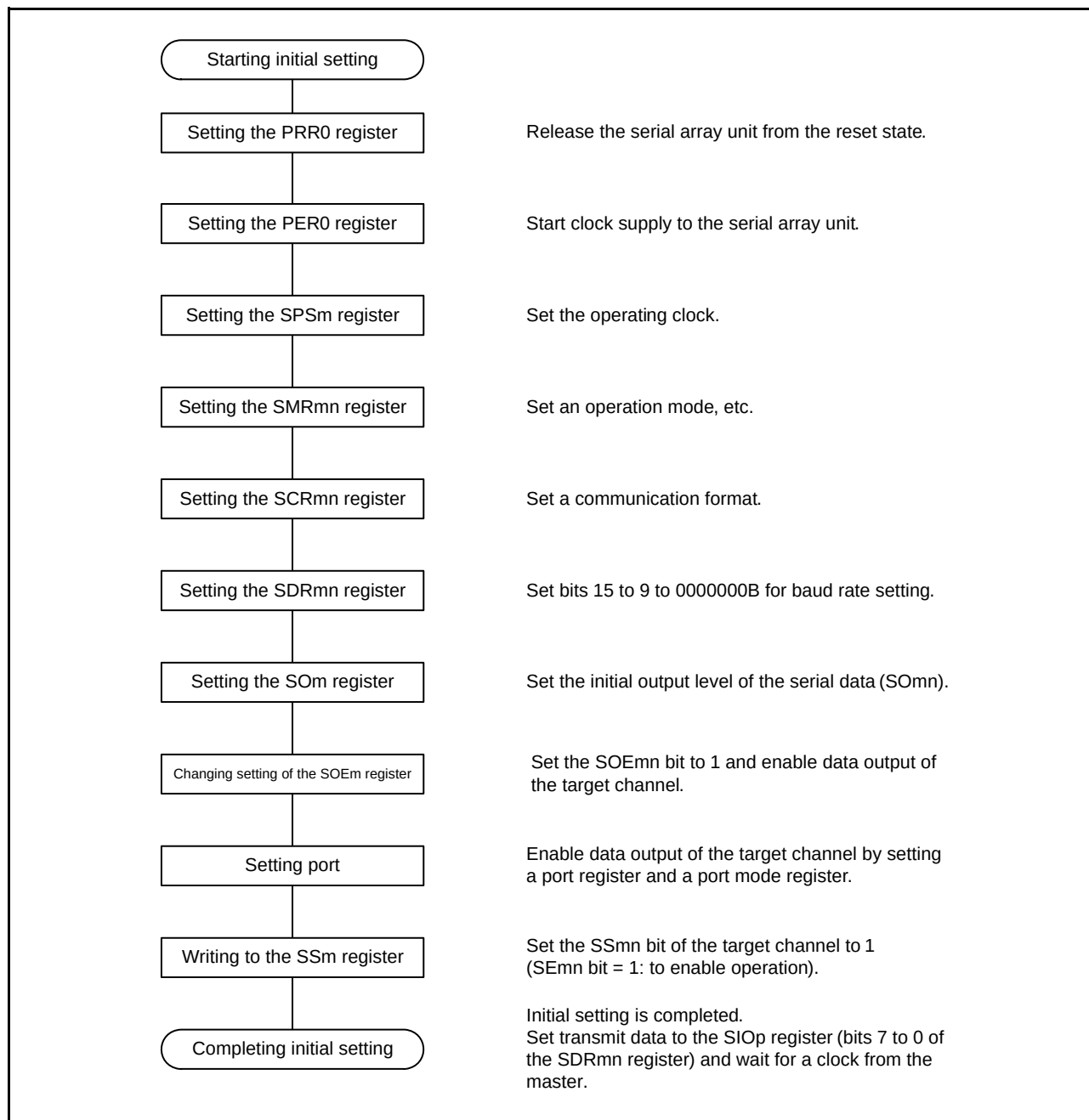
Remark 2. : Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode, : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 66 Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 13 - 67 Procedure for Stopping Slave Transmission/Reception

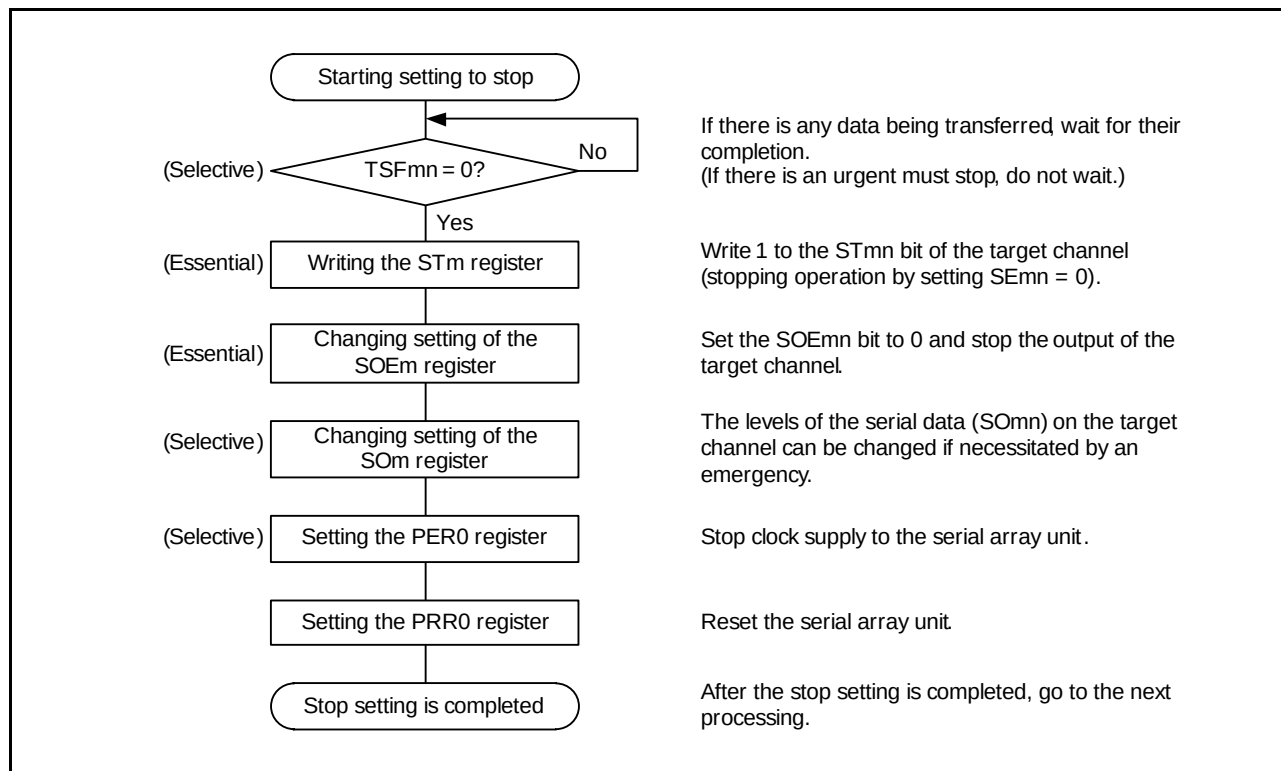
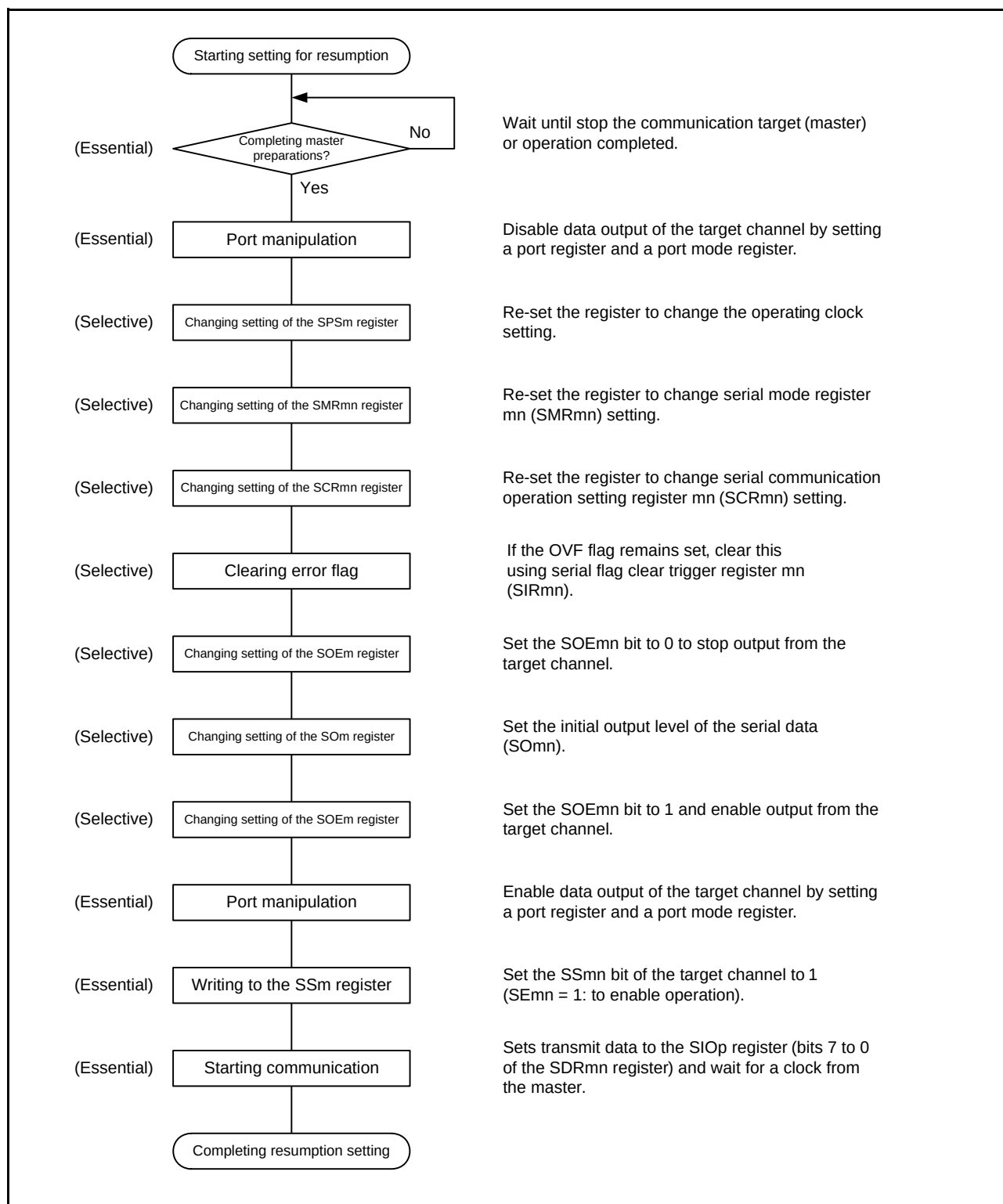


Figure 13 - 68 Procedure for Resuming Slave Transmission/Reception

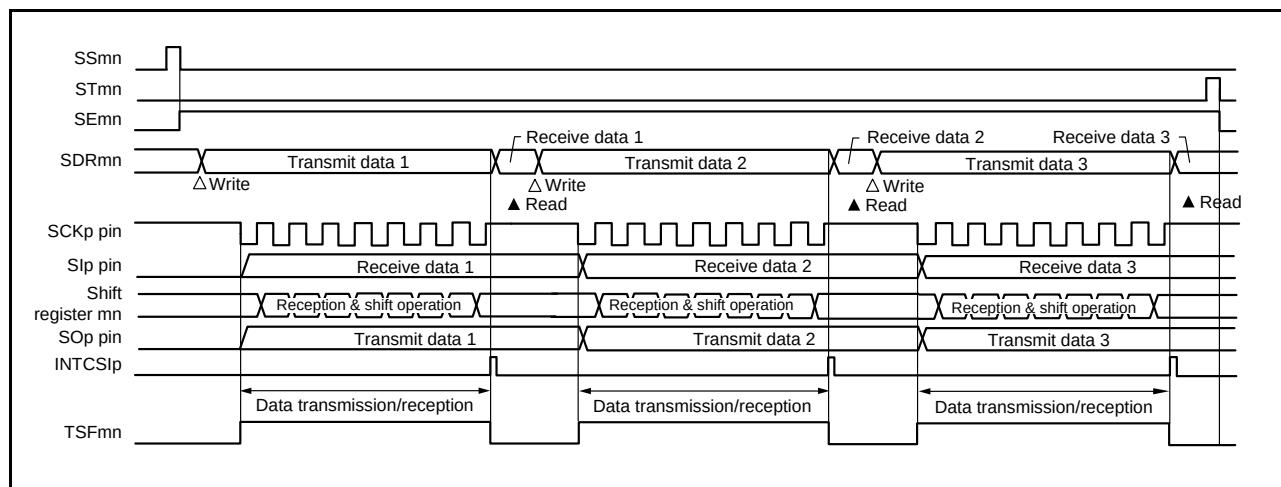


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then proceed initialization instead of restarting the communication.

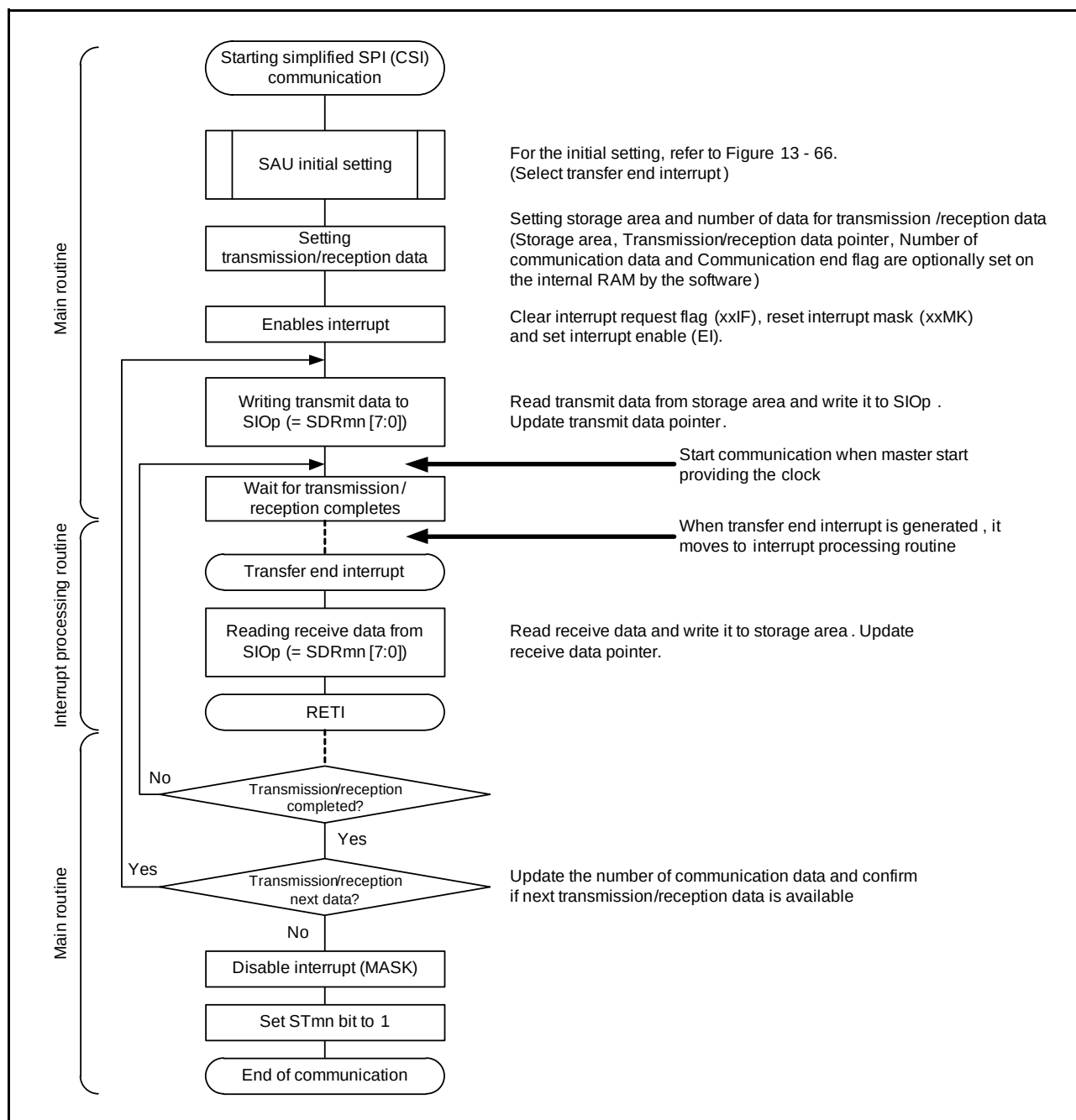
(3) Processing flow (in single-transmission/reception mode)

Figure 13 - 69 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

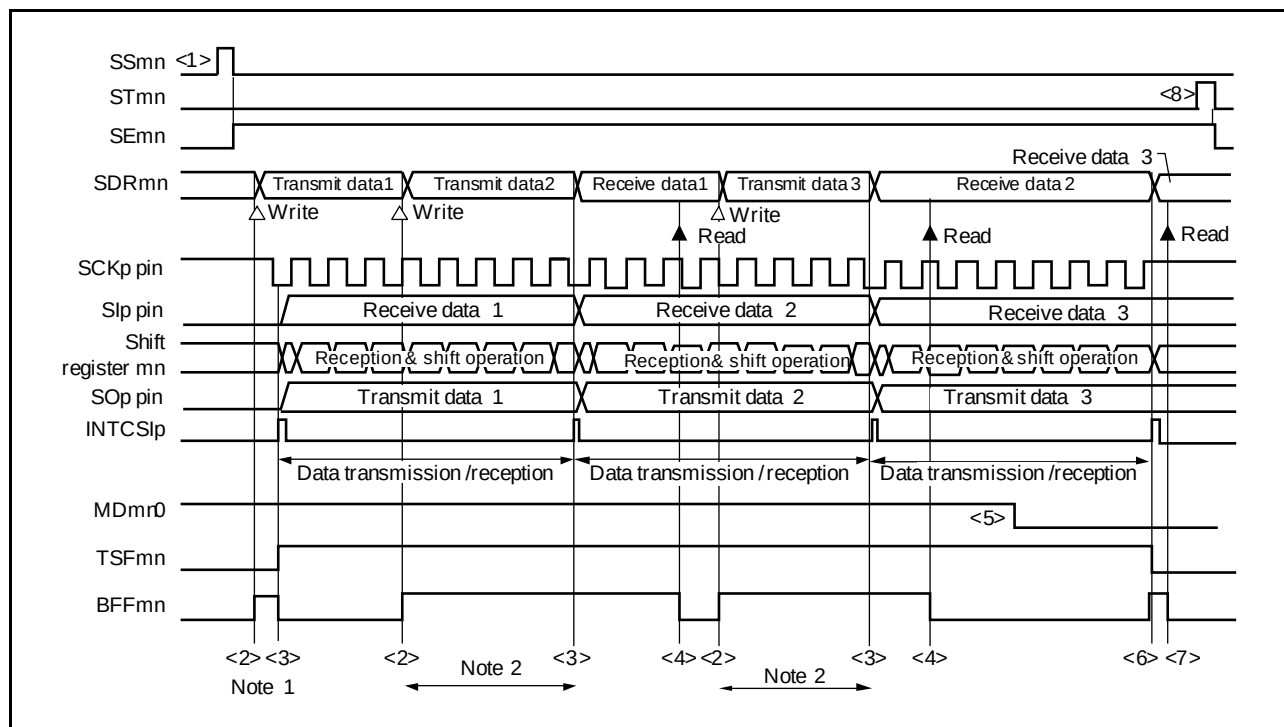
Figure 13 - 70 Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 13 - 71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

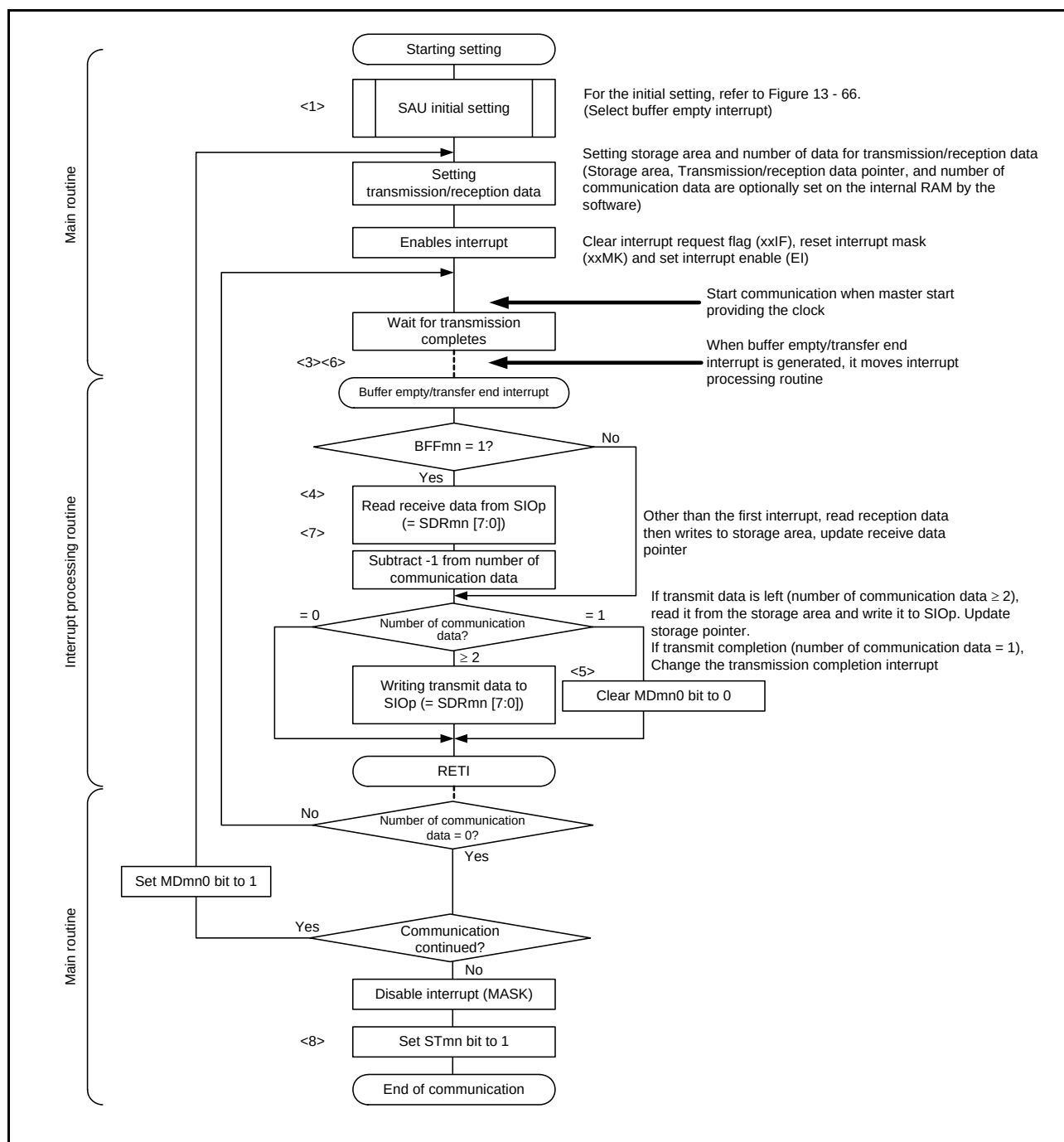
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.7 SNOOZE Mode

The SNOOZE mode enables the simplified SPI (CSI) to handle reception on detection of an input on the SCK00 pin in the STOP mode. Normally the simplified SPI (CSI) stops communication in the STOP mode. However, using the SNOOZE mode enables the simplified SPI (CSI) to handle reception without CPU operation upon detection of the SCK00 pin input. The SNOOZE mode is only available in CSI00.

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (See **Figure 13 - 74 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 13 - 76 Flowchart of SNOOZE Mode Operation (continuous startup).**)

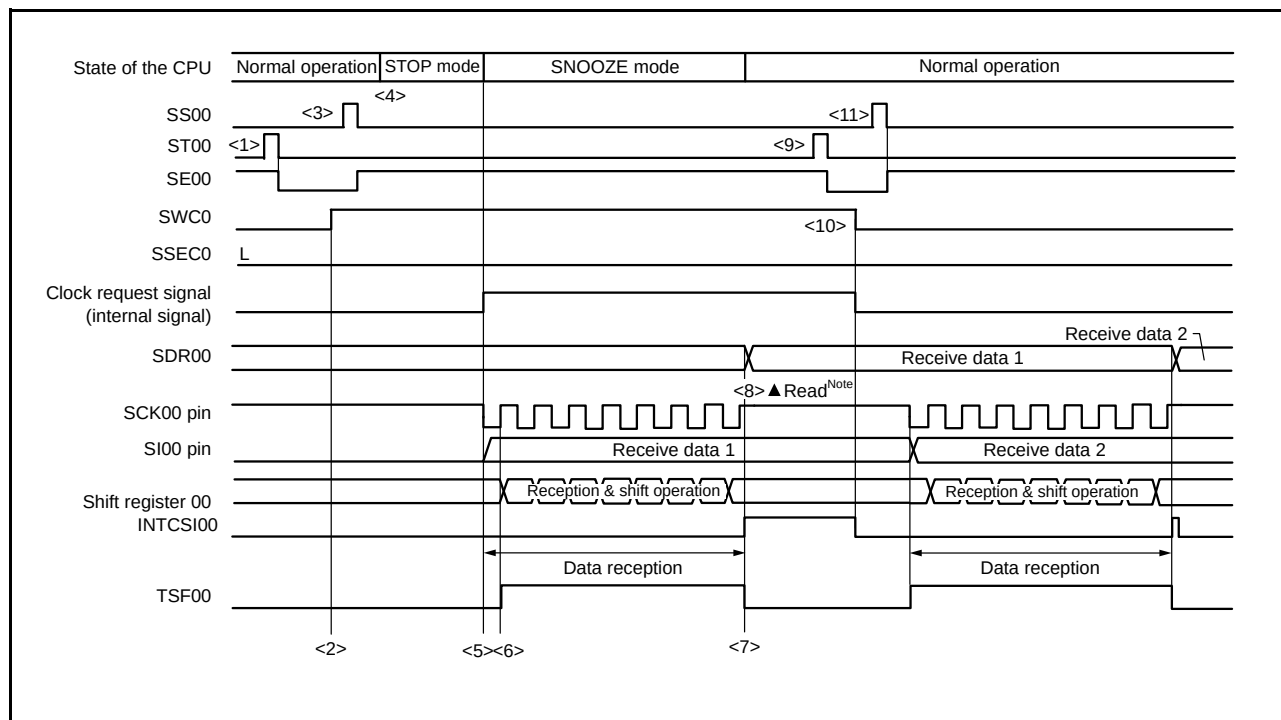
- When using the SNOOZE mode function, set the SWC0 bit of serial standby control register 0 (SSC0) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SS00 bit of serial channel start register 0 (SS0) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCK00 signal following a transition to the STOP mode. A CSI00 starts reception on detecting input of the serial clock on the SCK00 pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for fCLK.

Caution 2. The maximum transfer rate when using CSI00 in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 13 - 73 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAP00 = 0, CKP00 = 0)



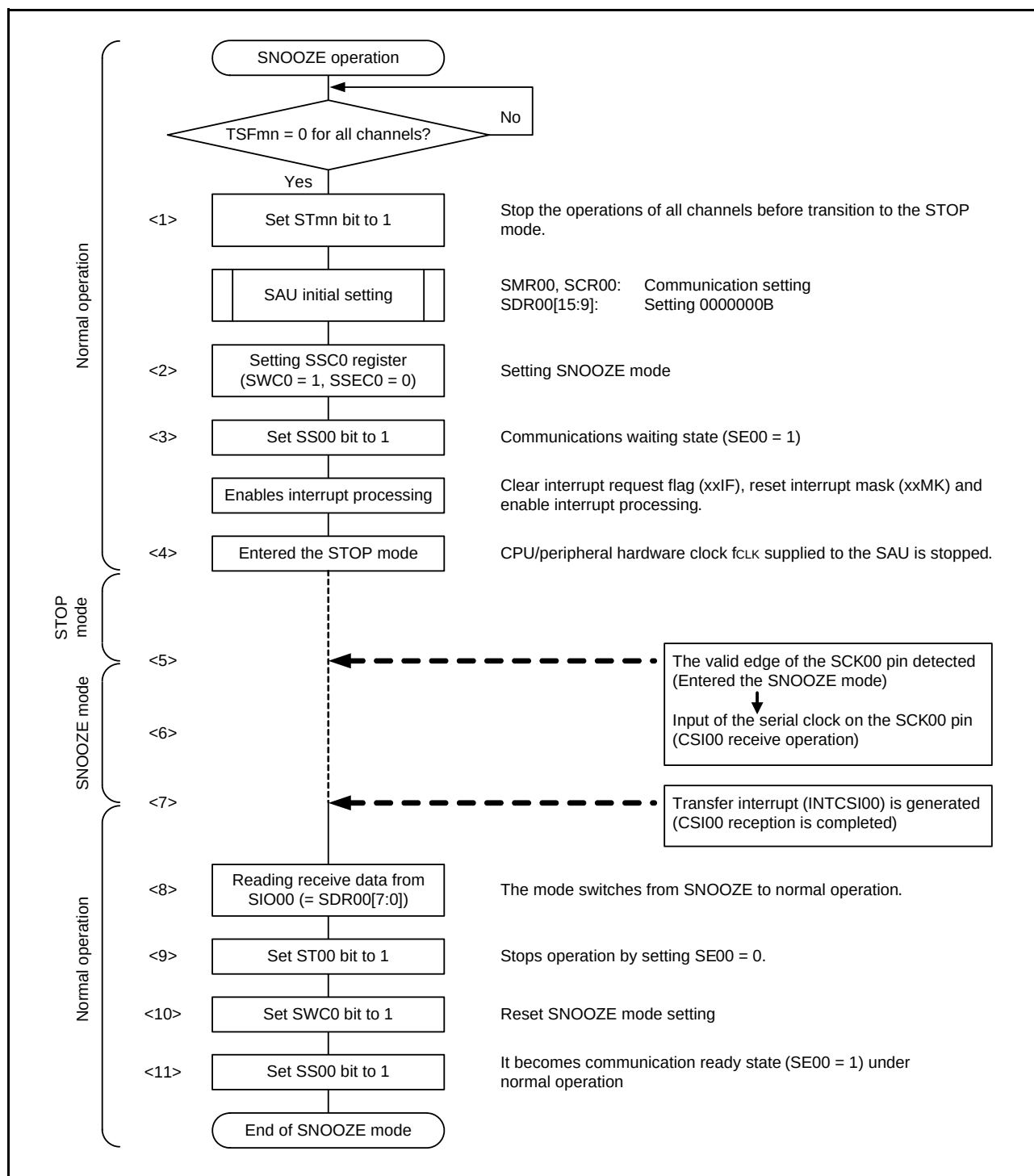
Note Only read received data while SWC0 = 1 and before the next valid edge of the SCK00 pin input is detected.

Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST00 bit to 1 (the SE00 bit is cleared and the operation stops). After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).

Caution 2. When SWC0 = 1, the BFF00 and OVF00 flags will not change.

Remark <1> to <11> in the figure correspond to <1> to <11> in Figure 13 - 74 Flowchart of SNOOZE Mode Operation (once startup).

Figure 13 - 74 Flowchart of SNOOZE Mode Operation (once startup)

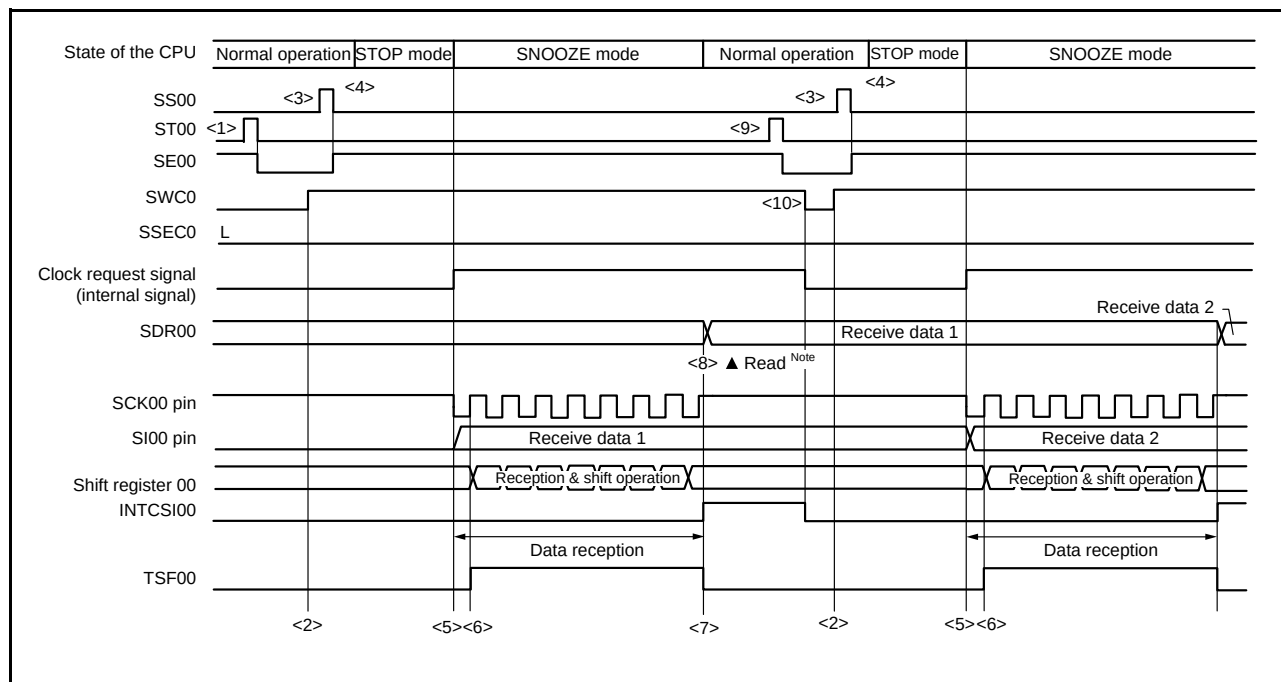


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 13 - 73 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAP00 = 0, CKP00 = 0).

Remark 2. 16-pin products: m = 0; n = 0
20- to 25-pin products: m = 0; n = 0, 3
30- and 32-pin products: m = 0, 1; n = 0, 3
36- to 44-pin products: m = 0, 1; n = 0, 1, 3
48-pin products: m = 0, 1; n = 0, 1, 3

(2) SNOOZE mode operation (continuous startup)

Figure 13 - 75 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAP00 = 0, CKP00 = 0)



Note Only read received data while SWC0 = 1 and before the next valid edge of the SCK00 pin input is detected.

Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST00 bit to 1 (the SE00 bit is cleared and the operation stops). After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).

Caution 2. When SWC0 = 1, the BFF00 and OVF00 flags will not change.

Remark <1> to <10> in the figure correspond to <1> to <10> in **Figure 13 - 76 Flowchart of SNOOZE Mode Operation (continuous startup)**.

```

graph TD
    Start([SNOOZE operation]) --> Decision{TSFmn = 0 for all channels?}
    Decision -- No --> Start
    Decision -- Yes --> S1[Set STmn bit to 1]
    S1 --> S2[SAU initial setting]
    S2 --> S3[Setting SSC0 register  
(SWC0 = 1, SSEC0 = 0)]
    S3 --> S4[Set SS00 bit to 1]
    S4 --> S5[Enables interrupt processing]
    S5 --> S6[Entered the STOP mode]
    S6 -.-> S5
    S6 -.-> S7[Transfer interrupt (INTCSI00) is generated  
(CSI00 reception is completed)]
    S7 -.-> S5
    S6 -.-> S8[Reading receive data from  
SIO00 (= SDR00[7:0])]
    S8 --> S9[Set ST00 bit to 1]
    S9 --> S10[Clear SWC0 bit to 0]
    S10 --> S3

```

Normal operation

<1> Set STmn bit to 1
Stop the operations of all channels before transition to the STOP mode.

SAU initial setting
SMR00, SCR00: Communication setting
SDR00[15:9]: Setting 0000000B

STOP mode

<2> Setting SSC0 register (SWC0 = 1, SSEC0 = 0)
Setting SNOOZE mode

<3> Set SS00 bit to 1
Communications waiting state (SE00 = 1)

Enables interrupt processing
Clear interrupt request flag (xxIF), reset interrupt mask (xxMK) and enable interrupt processing.

<4> Entered the STOP mode
CPU/peripheral hardware clock fCLK supplied to the SAU is stopped.

SNOOZE mode

<5> The valid edge of the SCK00 pin detected (Entered the SNOOZE mode)
↓
Input of the serial clock on the SCK00 pin (CSI00 receive operation)

<6>

<7> Transfer interrupt (INTCSI00) is generated (CSI00 reception is completed)

Normal operation

<8> Reading receive data from SIO00 (= SDR00[7:0])
The mode switches from SNOOZE to normal operation.

<9> Set ST00 bit to 1

<10> Clear SWC0 bit to 0
Reset SNOOZE mode setting

Remark 2. 16-pin products: $m = 0$; $n = 0$
 20- to 25-pin products: $m = 0$; $n = 0, 3$
 30- and 32-pin products: $m = 0, 1$; $n = 0, 3$
 36- to 44-pin products: $m = 0, 1$; $n = 0, 1, 3$
 48-pin products: $m = 0, 1$; $n = 0, 1, 3$

13.5.8 Calculating Transfer Clock Frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01, CSI11, CSI20, CSI21) communication can be calculated by the following expressions.

(1) Master

$$(\text{Transfer clock frequency}) = \{\text{Operating clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) Slave

$$(\text{Transfer clock frequency}) = \{\text{Frequency of serial clock (SCK) supplied by master}\} \text{Note} [\text{Hz}]$$

Note The permissible maximum transfer clock frequency is fMCK/6.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operating clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 13 - 2 Selection of Operating Clock For Simplified SPI (1/2)

SMRmn Register	SPSm Register								Operating Clock (fMCK) ^{Note}	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		fCLK = 32 MHz
0	x	x	x	x	0	0	0	0	fCLK	32 MHz
	x	x	x	x	0	0	0	1	fCLK/2	16 MHz
	x	x	x	x	0	0	1	0	fCLK/2 ²	8 MHz
	x	x	x	x	0	0	1	1	fCLK/2 ³	4 MHz
	x	x	x	x	0	1	0	0	fCLK/2 ⁴	2 MHz
	x	x	x	x	0	1	0	1	fCLK/2 ⁵	1 MHz
	x	x	x	x	0	1	1	0	fCLK/2 ⁶	500 kHz
	x	x	x	x	0	1	1	1	fCLK/2 ⁷	250 kHz
	x	x	x	x	1	0	0	0	fCLK/2 ⁸	125 kHz
	x	x	x	x	1	0	0	1	fCLK/2 ⁹	62.5 kHz
	x	x	x	x	1	0	1	0	fCLK/2 ¹⁰	31.25 kHz
	x	x	x	x	1	0	1	1	fCLK/2 ¹¹	15.63 kHz
	x	x	x	x	1	1	0	0	fCLK/2 ¹²	7.81 kHz
	x	x	x	x	1	1	0	1	fCLK/2 ¹³	3.91 kHz
	x	x	x	x	1	1	1	0	fCLK/2 ¹⁴	1.95 kHz
	x	x	x	x	1	1	1	1	fCLK/2 ¹⁵	977 Hz

Table 13 - 2 Selection of Operating Clock For Simplified SPI (2/2)

SMRmn Register	SPSm Register								Operating Clock (fMCK) ^{Note}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		fCLK = 32 MHz
1	0	0	0	0	x	x	x	x	fCLK	32 MHz
	0	0	0	1	x	x	x	x	fCLK/2	16 MHz
	0	0	1	0	x	x	x	x	fCLK/2 ²	8 MHz
	0	0	1	1	x	x	x	x	fCLK/2 ³	4 MHz
	0	1	0	0	x	x	x	x	fCLK/2 ⁴	2 MHz
	0	1	0	1	x	x	x	x	fCLK/2 ⁵	1 MHz
	0	1	1	0	x	x	x	x	fCLK/2 ⁶	500 kHz
	0	1	1	1	x	x	x	x	fCLK/2 ⁷	250 kHz
	1	0	0	0	x	x	x	x	fCLK/2 ⁸	125 kHz
	1	0	0	1	x	x	x	x	fCLK/2 ⁹	62.5 kHz
	1	0	1	0	x	x	x	x	fCLK/2 ¹⁰	31.25 kHz
	1	0	1	1	x	x	x	x	fCLK/2 ¹¹	15.63 kHz
	1	1	0	0	x	x	x	x	fCLK/2 ¹²	7.81 kHz
	1	1	0	1	x	x	x	x	fCLK/2 ¹³	3.91 kHz
	1	1	1	0	x	x	x	x	fCLK/2 ¹⁴	1.95 kHz
	1	1	1	1	x	x	x	x	fCLK/2 ¹⁵	977 Hz
Other than above									Setting prohibited	

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. x: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

13.5.9 Procedure for Processing Errors that Occurred during Simplified SPI (CSI00, CSI01, CSI11, CSI20, and CSI21) Communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI11, CSI20, and CSI21) communication is described in **Table 13 - 3**.

Table 13 - 3 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

13.6 Operation of UART (UART0 to UART2) Communication

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex asynchronous communication UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using UART2, timer array unit 0 (channel 7), and an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits **Note**
- MSB/LSB first selectable
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception supports the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

UART2 (channels 0 and 1 of unit 1) in 30-pin to 48-pin products only supports the LIN-bus.

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and the timer array unit 0 (channel 7)

Note UART0 only supports the 9-bit data length.

When the medium-speed on-chip oscillator clock (f_{IM}) or low-speed on-chip oscillator clock (f_{IL}) is selected for f_{CLK}, use the medium-speed on-chip oscillator trimming register (MIOTRM) or the low-speed on-chip oscillator trimming register (LIOTRM) to correct the accuracy of the oscillation frequency.

UART0 uses channels 0 and 1 of SAU0.

UART1 uses channels 2 and 3 of SAU0.

UART2 uses channels 0 and 1 of SAU1.

<16-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		IIC11

<20-, 24-, and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	—		—

<36-, 40-, and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

<48-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

Select a single function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, the CSI00 and CSI01 functions cannot be used. However, channel 3 of the same unit can be used for a function other than UART0, such as CSI11, UART1, and IIC11.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

- UART transmission (See **13.6.1.**)
- UART reception (See **13.6.2.**)
- LIN transmission (UART2 only) (See **13.7.1.**)
- LIN reception (UART2 only) (See **13.7.2.**)

13.6.1 UART Transmission

UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1	UART2
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7, 8, or 9 bits Note 1		
Transfer rate Note 2	Max. $f_{MCK}/6$ [bps] ($SDR_{mn}[15:9] = 2$ or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bit		
Data direction	MSB or LSB first		

Note 1. UART0 only supports the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

Remark 1. f_{MCK} : Operating clock frequency of target channel

f_{CLK} : System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

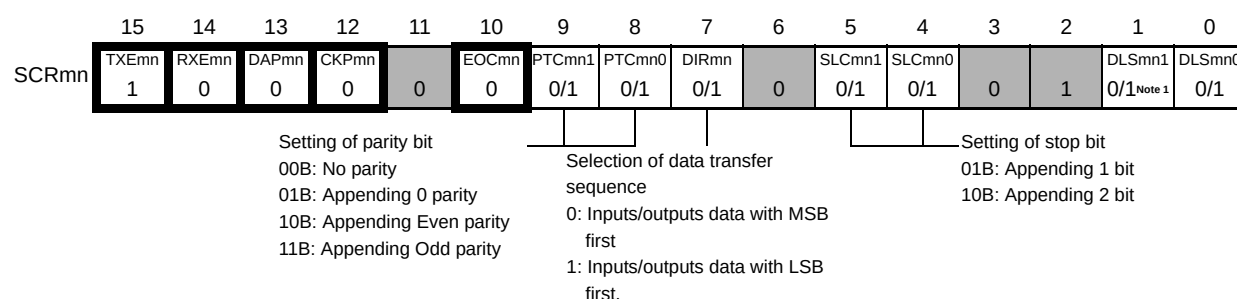
(1) Register setting

Figure 13 - 77 Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (1/2)

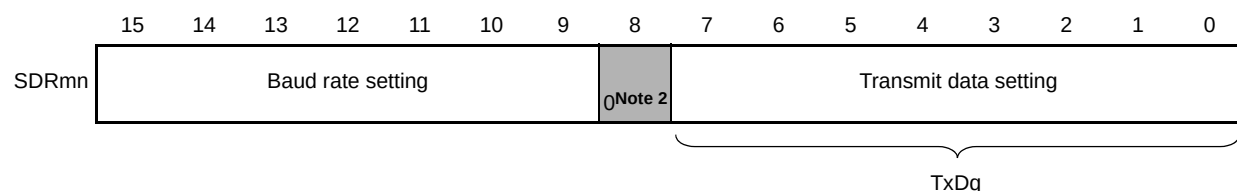
(a) Serial mode register mn (SMRmn)



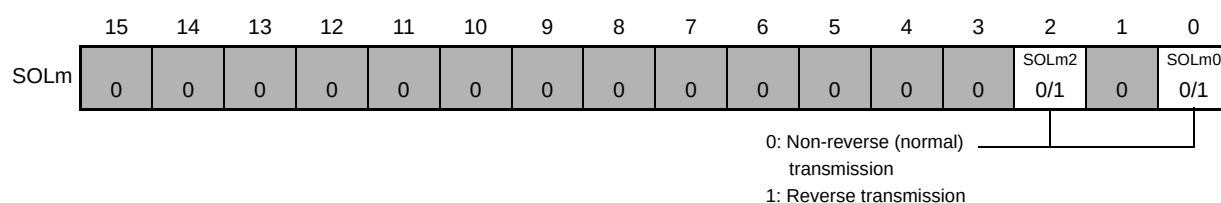
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TxDq)



(d) Serial output level register m (SOLm) ... Set only the bit of the target channel.



Note 1. This bit is only present in the SCR00 register, and is fixed to 1 in the other registers.

Note 2. Bits 0 to 8 of the SDRm0 register are used as the transmission data specification area in the 9-bit communications.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2),
mn = 00, 02, 10

Remark 2. : Setting is fixed in the UART transmission mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

Figure 13 - 77 Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (2/2)

(e) Serial output register m (SOM) ... Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3 ×	0	CKOm1 ×	CKOm0 ×	0	0	0	0	SOM3 ×	SOM2 0/1 ^{Note}	SOM1 ×	SOM0 0/1 ^{Note}

0: Serial data output value is 0
1: Serial data output value is 1

(f) Serial output enable register m (SOEm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 ×	SOEm2 0/1	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2),
mn = 00, 02, 10

Remark 2. ☐: Setting is fixed in the UART transmission mode, ☐: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 78 Initial Setting Procedure for UART Transmission

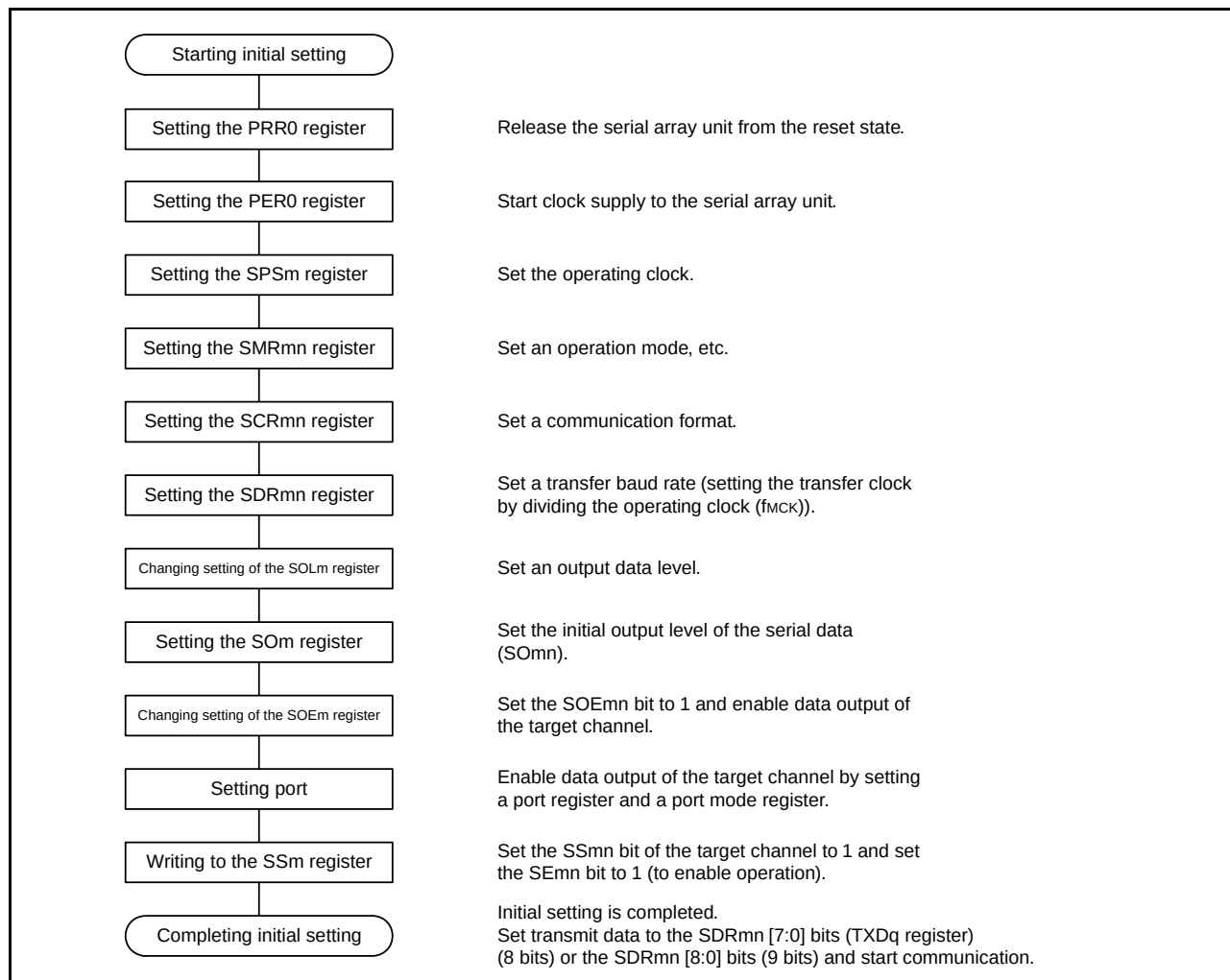


Figure 13 - 79 Procedure for Stopping UART Transmission

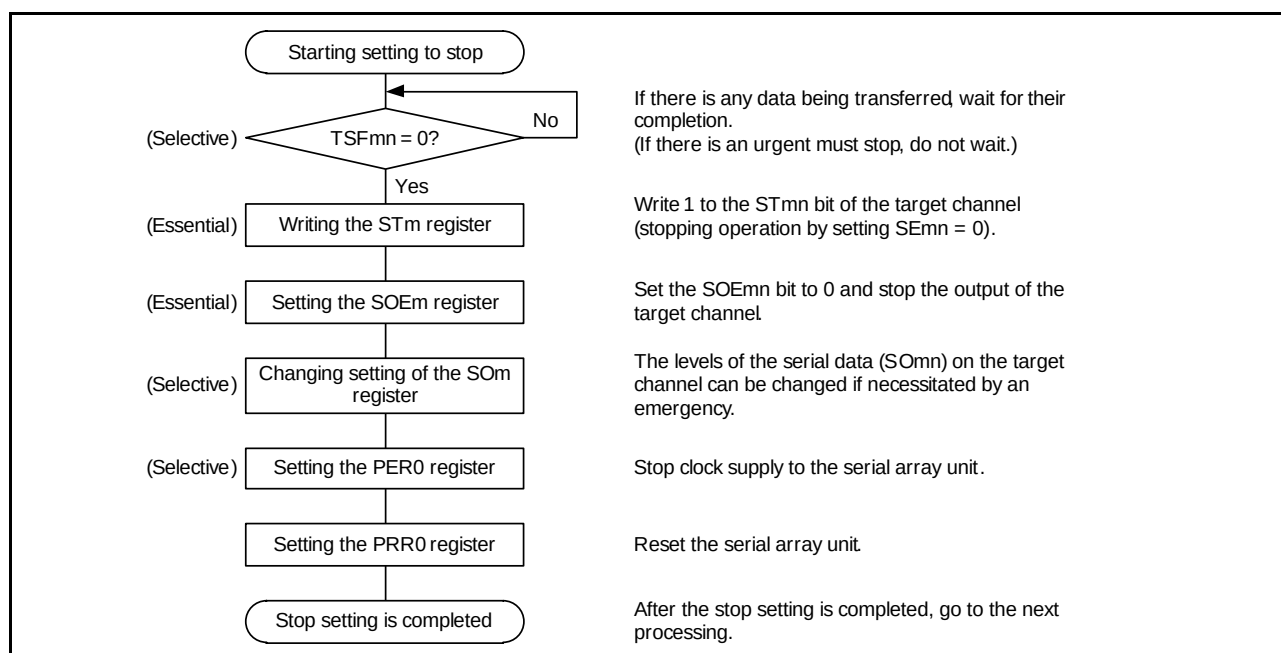
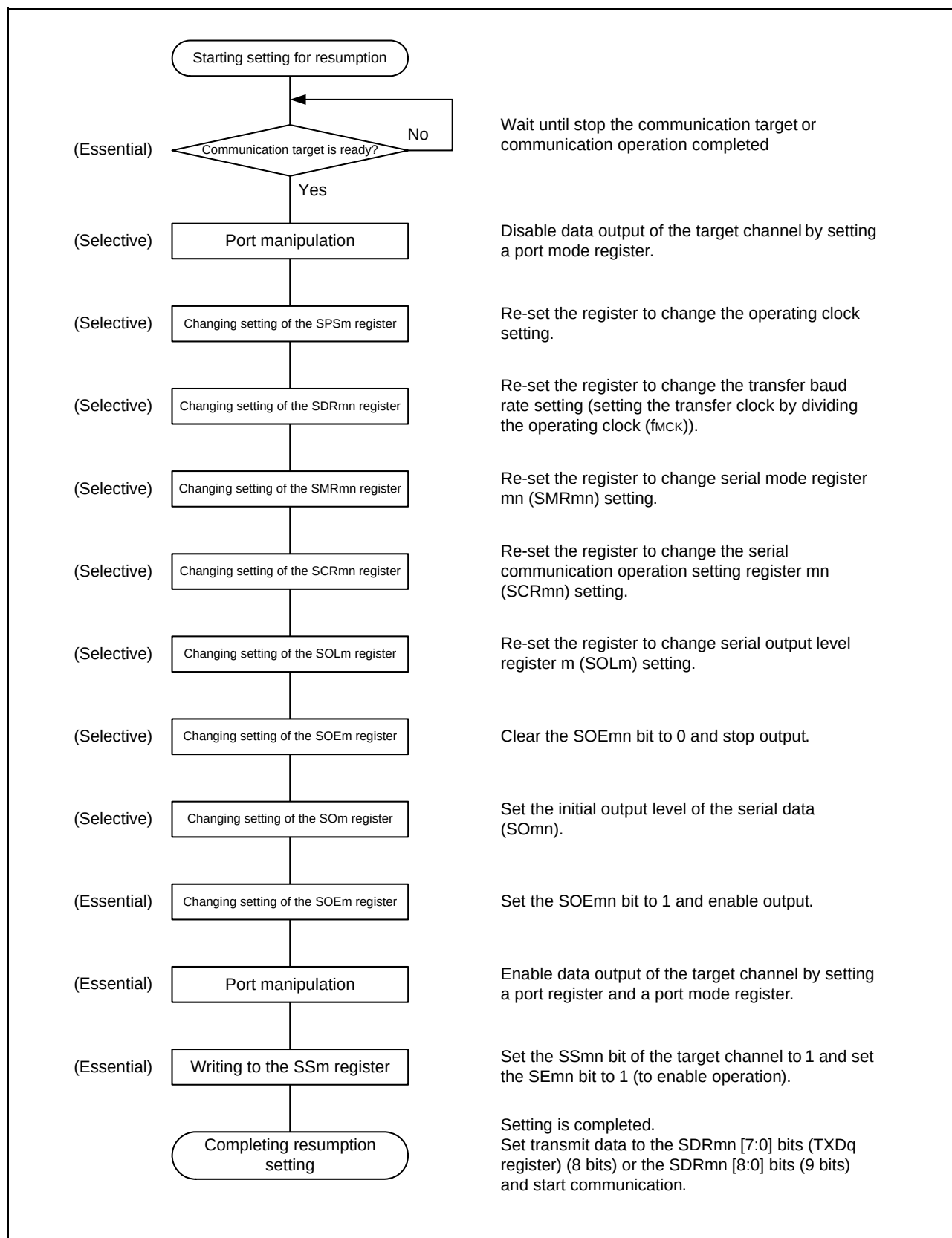


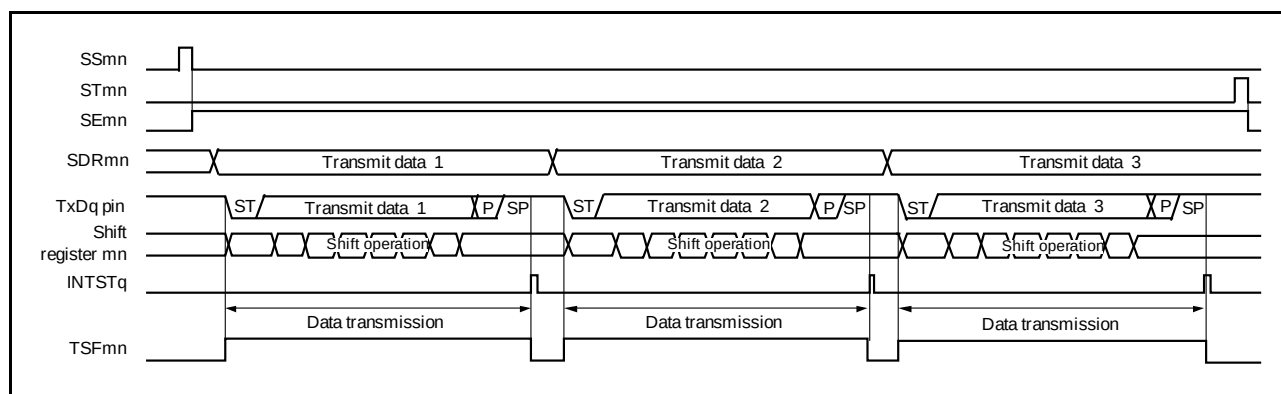
Figure 13 - 80 Procedure for Resuming UART Transmission



Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then proceed initialization instead of restarting the communication.

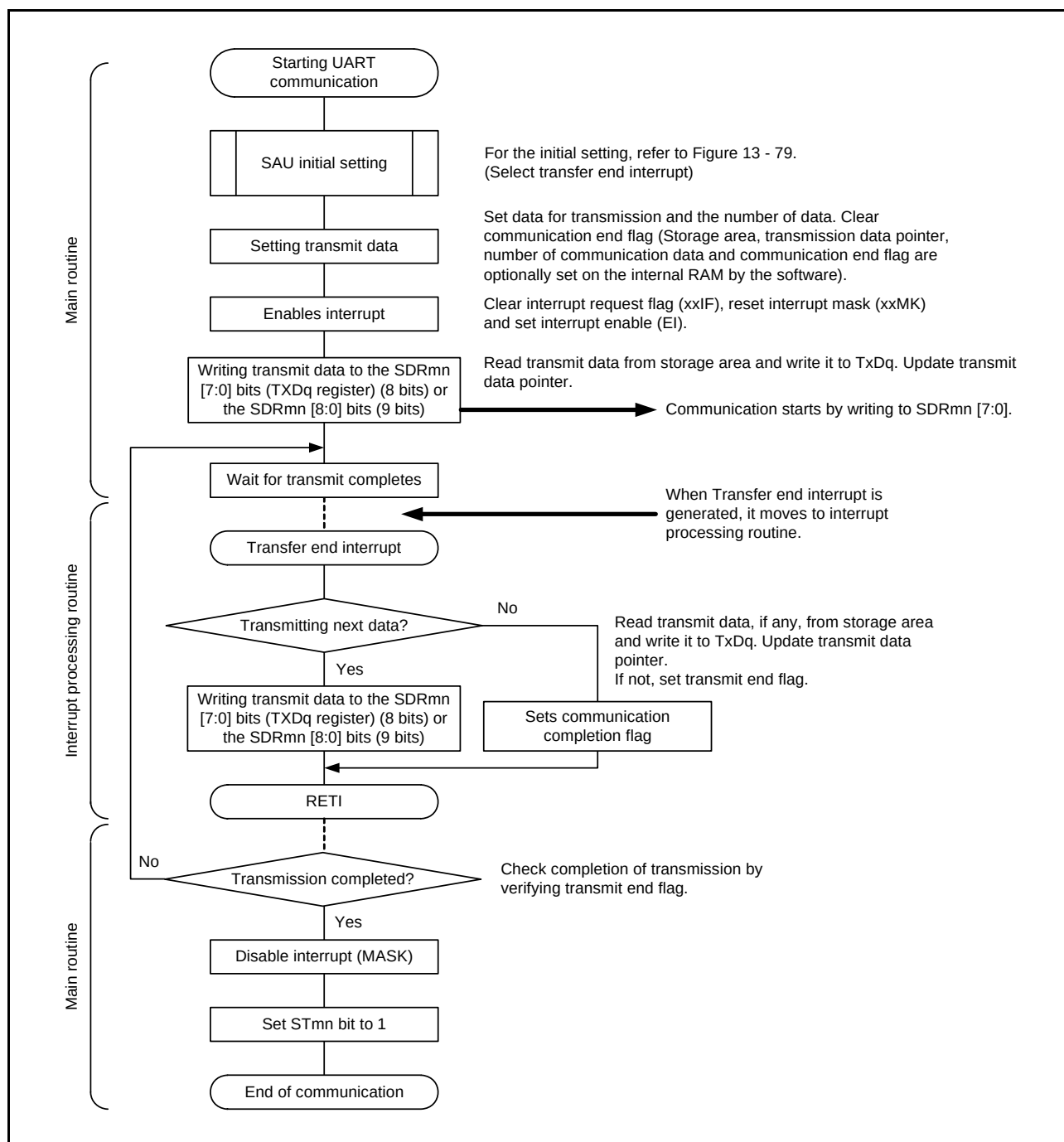
(3) Processing flow (in single-transmission mode)

Figure 13 - 81 Timing Chart of UART Transmission (in Single-Transmission Mode)



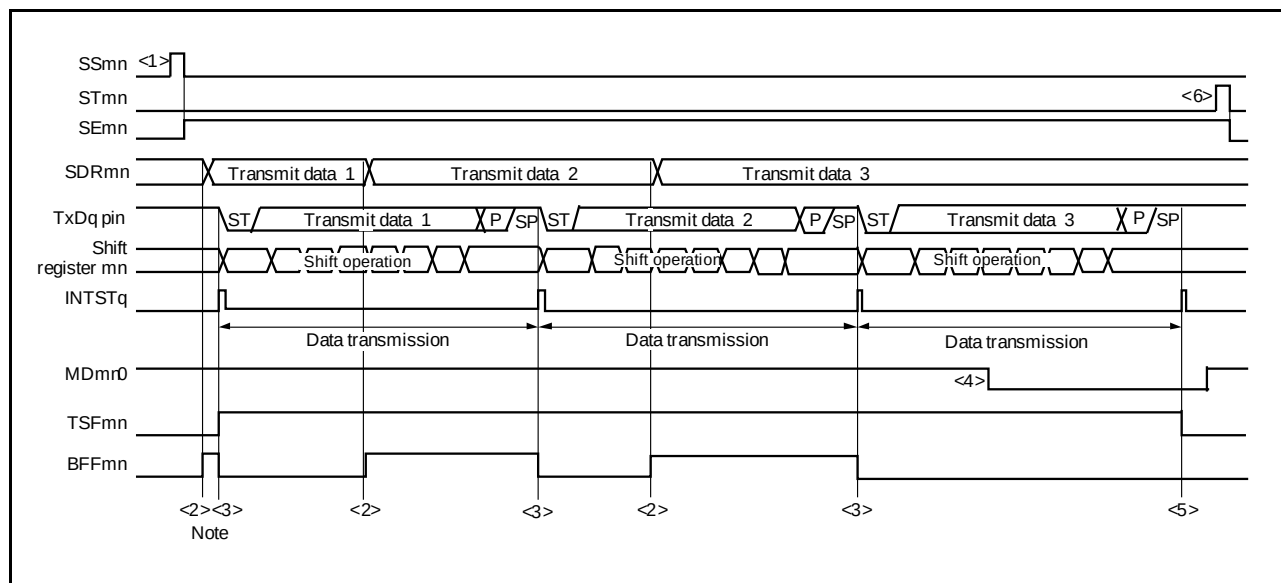
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2),
mn = 00, 02, 10

Figure 13 - 82 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13 - 83 Timing Chart of UART Transmission (in Continuous Transmission Mode)

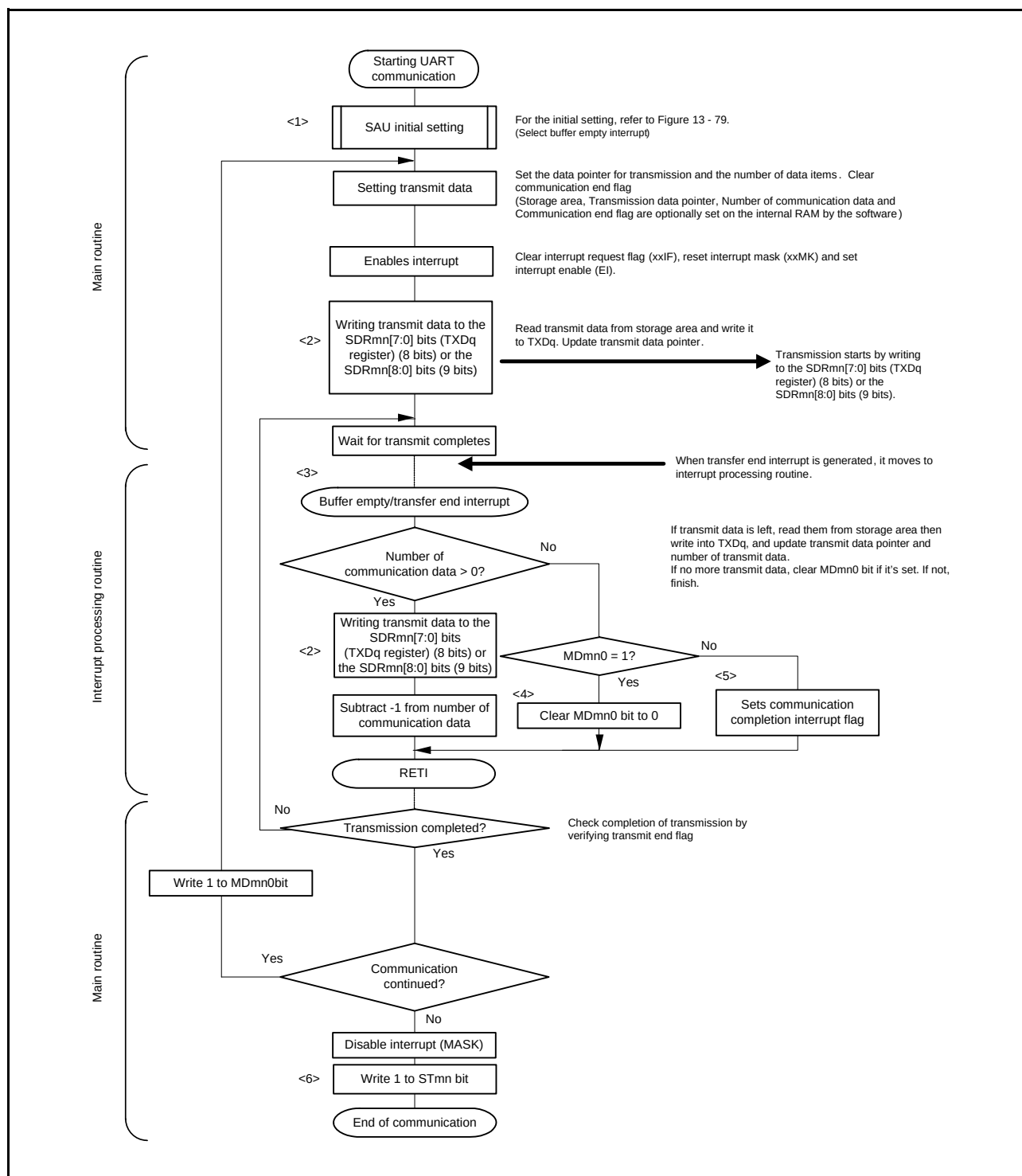


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2),
mn = 00, 02, 10

Figure 13 - 84 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 83 Timing Chart of UART Transmission (in Continuous Transmission Mode).

13.6.2 UART Reception

UART reception is an operation wherein the RL78 microcontroller asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1	UART2
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	- Framing error detection flag (FEFmn) - Parity error detection flag (PEFmn) - Overrun error detection flag (OVFmn)		
Transfer data length	7, 8, or 9 bits Note 1		
Transfer rate Note 2	Max. $f_{MCK}/6$ [bps] ($SDR_{mn}[15:9] = 2$ or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable - No parity bit (no parity check) - No parity judgment (0 parity) - Even parity check - Odd parity check		
Stop bit	Appending 1 bit		
Data direction	MSB or LSB first		

Note 1. UART0 only supports the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

Remark 1. f_{MCK} : Operating clock frequency of target channel

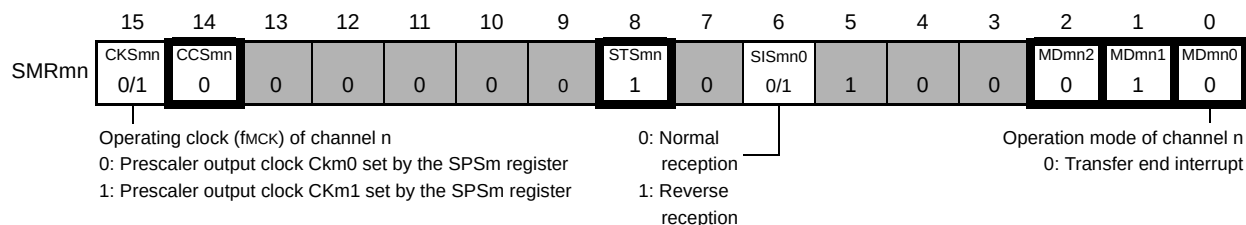
f_{CLK} : System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

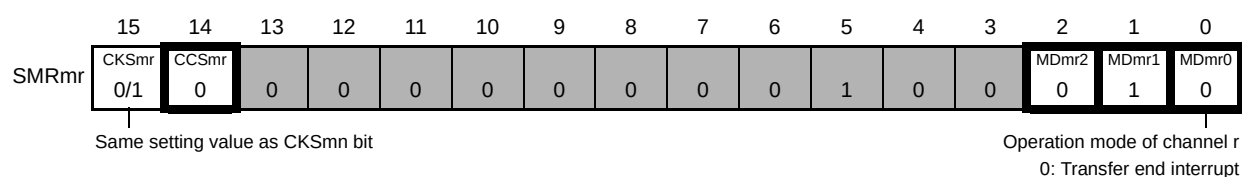
(1) Register setting

Figure 13 - 85 Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (1/2)

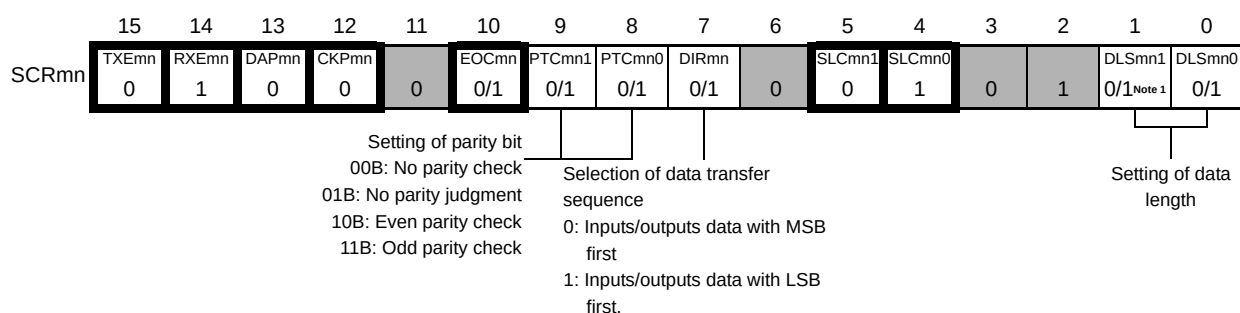
(a) Serial mode register mn (SMRmn)



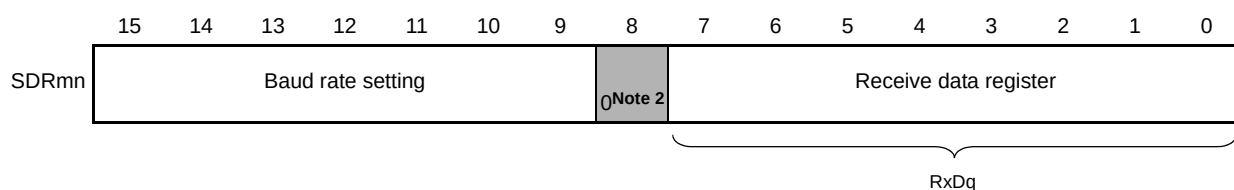
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RxDq)



Note 1. This bit is only present in the SCR01 register, and is fixed to 1 in the other registers.

Note 2. Bits 0 to 8 of the SDRm1 register are used as the reception data specification area in the 9-bit communications. UART0 only supports the 9-bit communications.

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n - 1), q: UART number (q = 0 to 2)

Remark 2. : Setting is fixed in the UART reception mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

Figure 13 - 85 Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (2/2)

(e) Serial output register m (SOM) ... This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3 ×	0	CKOm1 ×	CKOm0 ×	0	0	0	0	SOM3 ×	SOM2 ×	SOM1 ×	SOM0 ×

(f) Serial output enable register m (SOEm) ... This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 ×	SOEm2 ×	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0, 1)

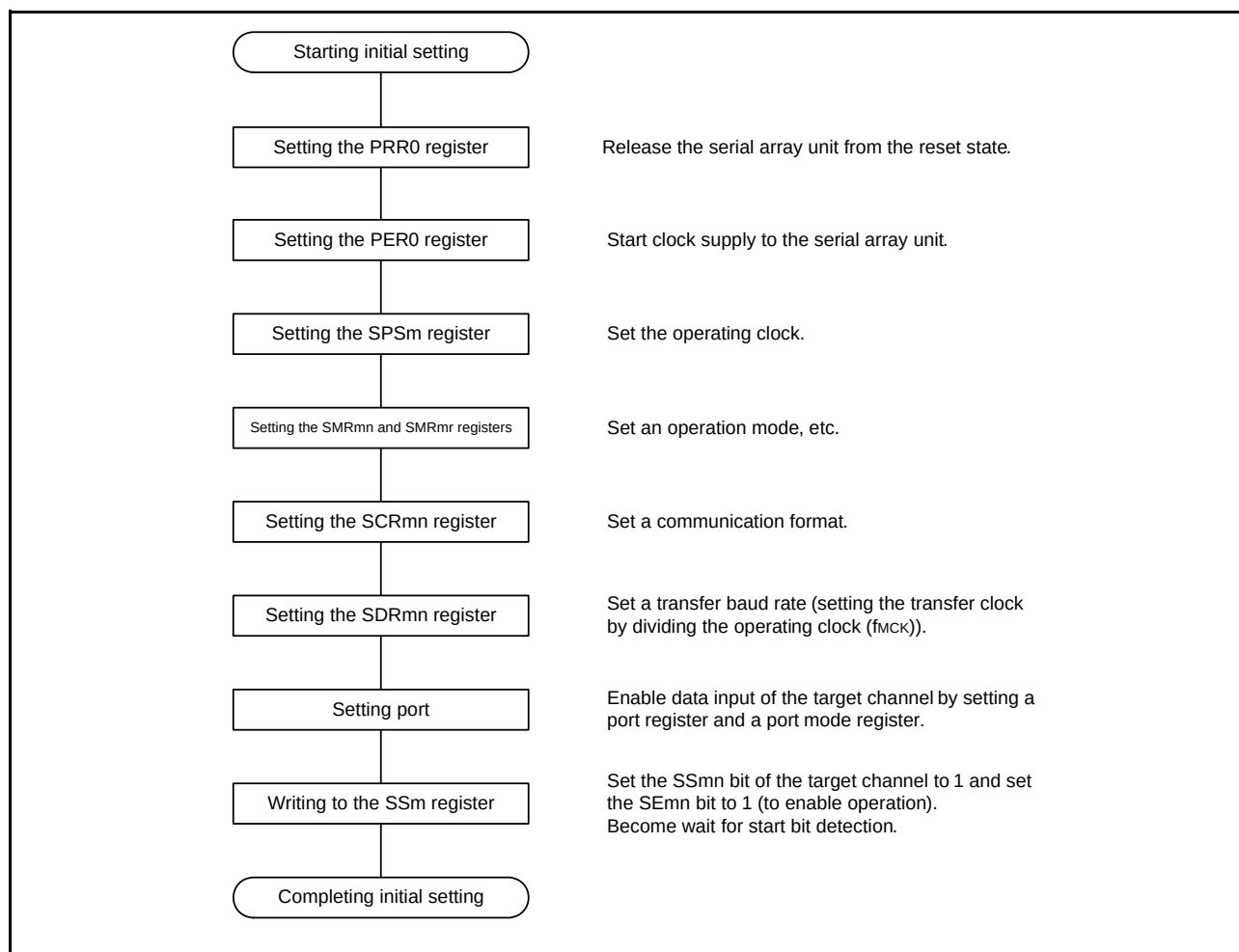
Remark 2. : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 13 - 86 Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fmck clock cycles have elapsed.

Figure 13 - 87 Procedure for Stopping UART Reception

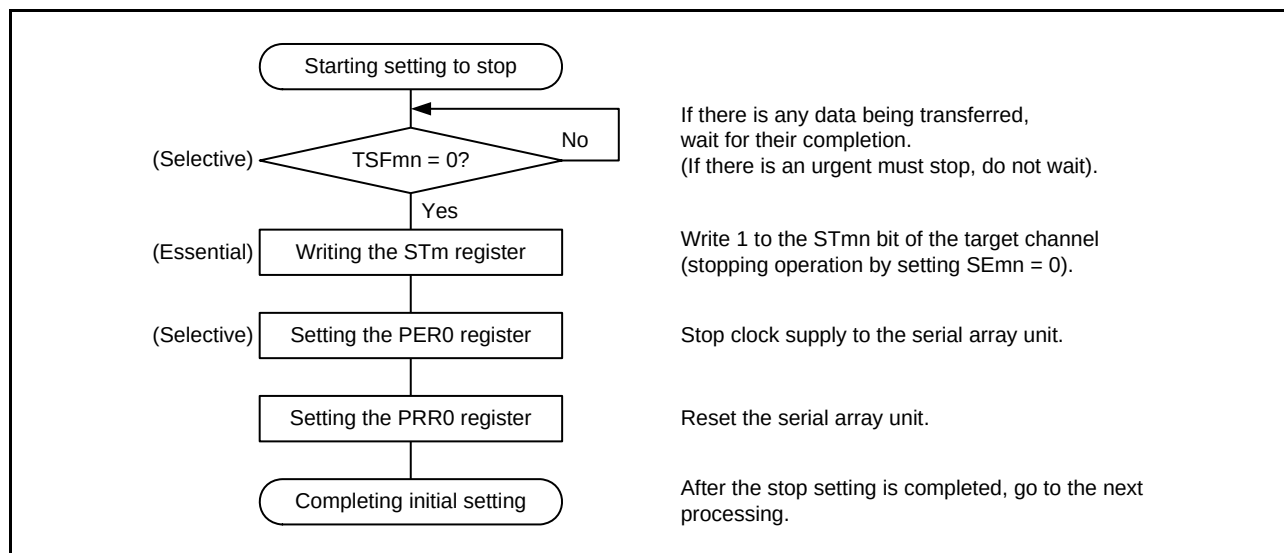
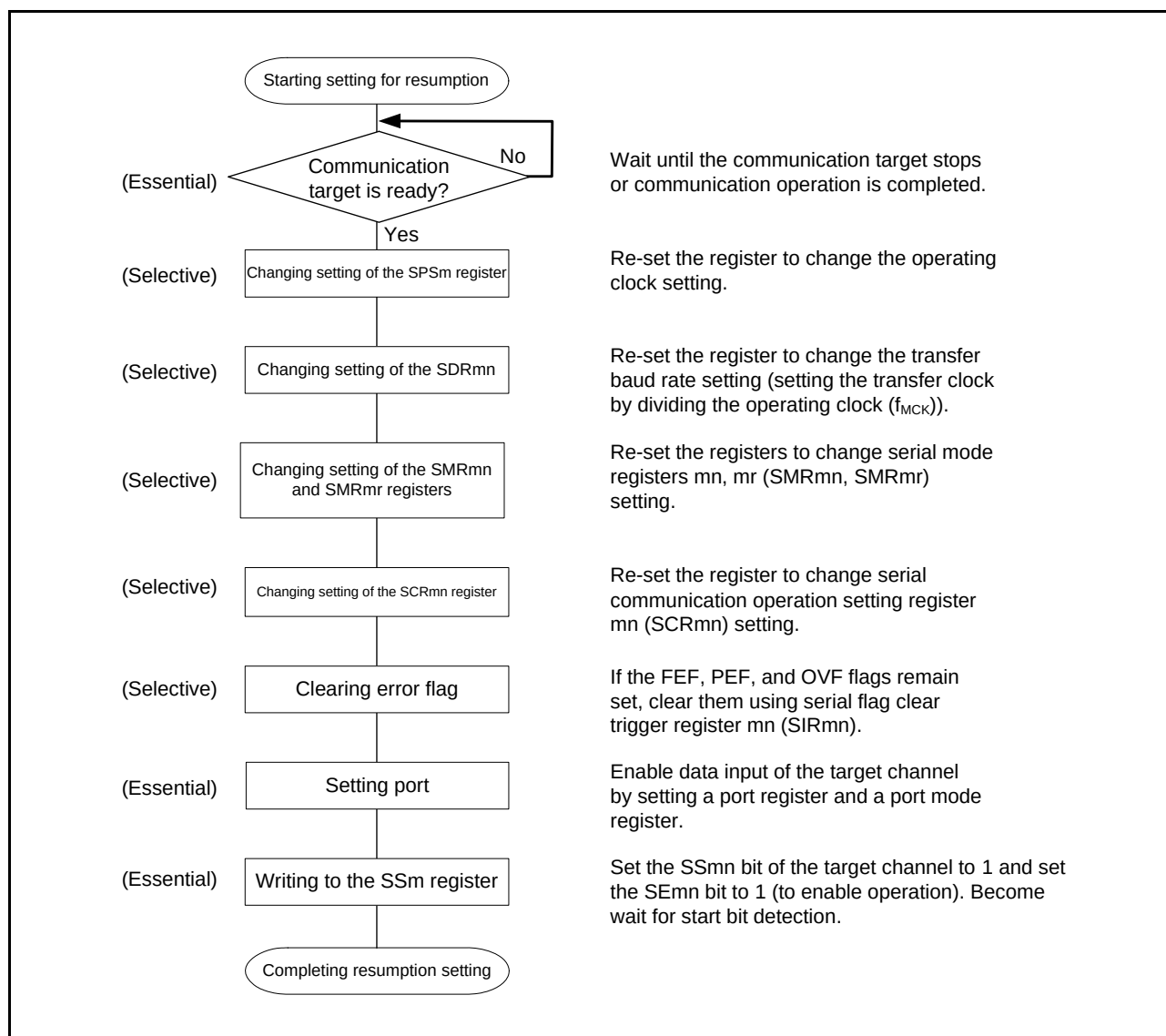


Figure 13 - 88 Procedure for Resuming UART Reception

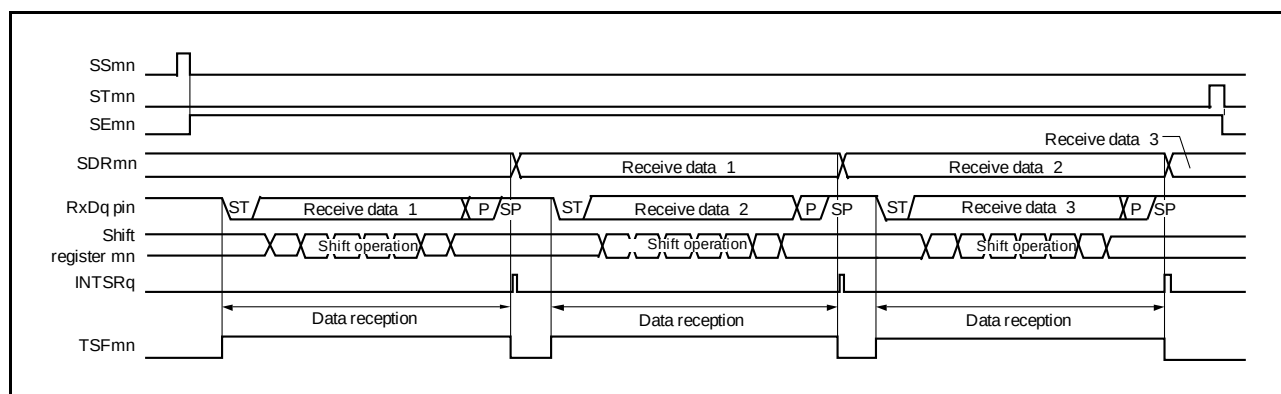


Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fmck clocks have elapsed.

Remark If PRR0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then proceed initialization instead of restarting the communication.

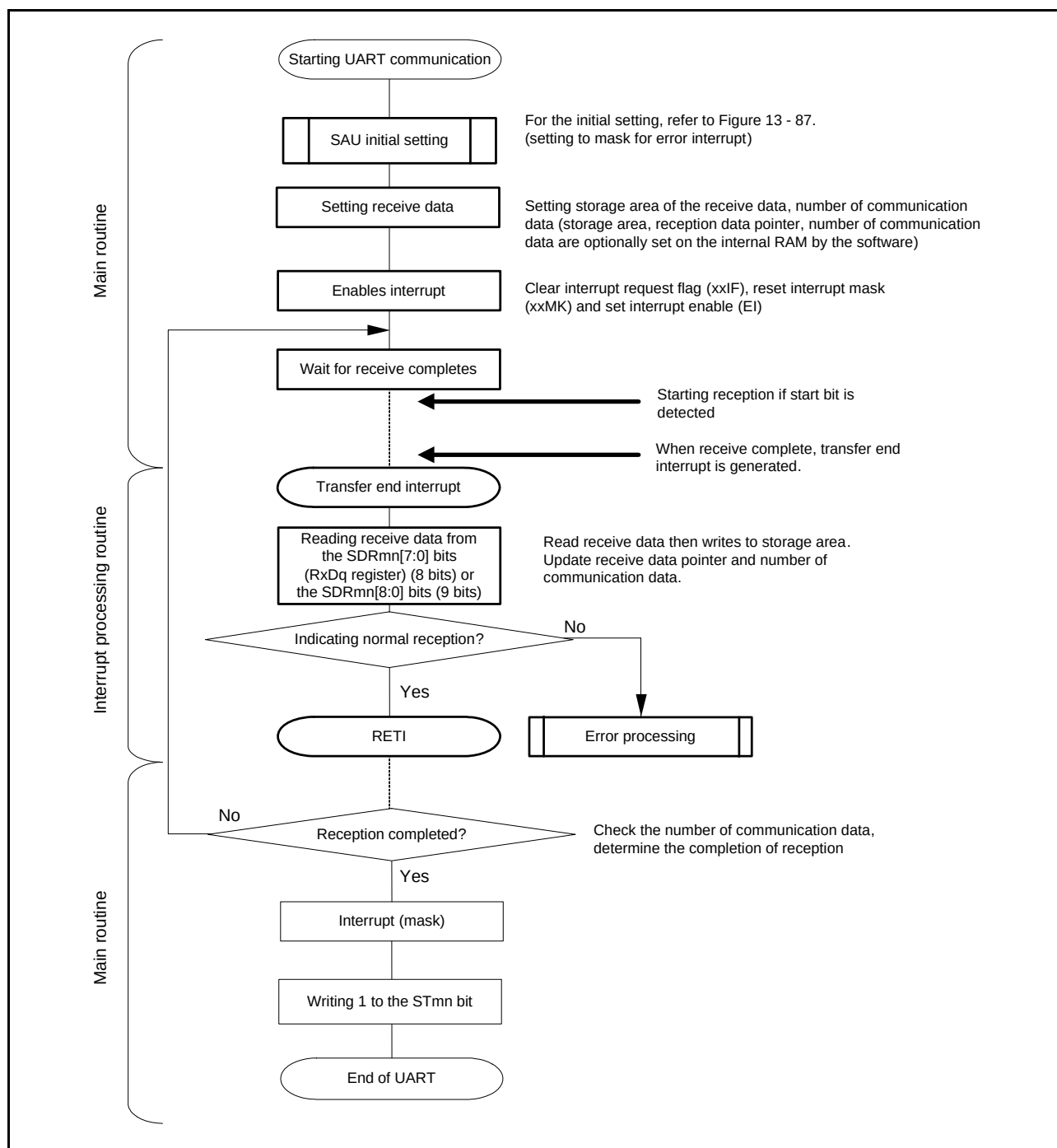
(3) Processing flow

Figure 13 - 89 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n - 1), q: UART number (q = 0 to 2)

Figure 13 - 90 Flowchart of UART Reception



13.6.3 SNOOZE Mode

The SNOOZE mode enables the UART to handle reception on detection of an input on the RxD0 pin in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to handle reception without CPU operation. UART0 only supports the SNOOZE mode.

When using UART0 in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 13 - 93 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)** and **Figure 13 - 95 Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1).**)

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPS0 register and bits 15 to 9 of the SDR01 register with reference to **Table 13 - 4**.
- Set the EOC01 and SSEC0 bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWC0 bit of serial standby control register 0 (SSC0) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SS01 bit of serial channel start register 0 (SS0) to 1.
- A UART0 starts reception in SNOOZE mode on detecting input of the start bit on the RxD0 pin following a transition of the CPU to the STOP mode.

Caution 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for fCLK.

When the medium-speed on-chip oscillator clock is selected, use the medium-speed on-chip oscillator trimming register (MIOTRM) to correct the accuracy of the oscillation frequency.

Caution 2. The maximum transfer rate in the SNOOZE mode is 115.2 kbps (when FWKUP = 1, fCLK = fIH (32 MHz)). When FWKUP is set to 1, fCLK cannot be set to a value other than fIH = 32 MHz.

Caution 3. When SWC0 = 1, UART0 can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- When after the SWC0 bit has been set to 1, the reception operation is started before the STOP mode is entered
- When the reception operation is started while another function is in the SNOOZE mode
- When after having returned from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWC0 bit is returned to 0

Caution 4. If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OVF01 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, and OVF01 flags before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxD0) of the SDR01 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxD0 signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxD0 pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Table 13 - 4 Baud Rate Setting for UART Reception in SNOOZE Mode when Starting of the High-Speed On-Chip Oscillator is at Normal Speed (FWKUP = 0)

Baud Rate	High-Speed On-Chip Oscillator (f _{IH})	Operating Clock (f _{MCK})	SDR01[15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ± 1% Note	f _{CLK} /2 ⁵	106	1.45%	-1.67%
	24 MHz ± 1% Note	f _{CLK} /2 ⁵	79	1.77%	-1.37%
9600 bps	32 MHz ± 1% Note	f _{CLK} /2 ⁴	106	1.45%	-1.67%
	24 MHz ± 1% Note	f _{CLK} /2 ⁴	79	1.77%	-1.37%

Table 13 - 5 Baud Rate Setting for UART Reception in SNOOZE Mode when Starting of the High-Speed On-Chip Oscillator is at High Speed (FWKUP = 1)

Baud Rate	High-Speed On-Chip Oscillator (f _{IH})	Operating Clock (f _{MCK})	SDR01[15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ± 1% Note	f _{CLK} /2 ⁵	106	1.45%	-1.67%
9600 bps		f _{CLK} /2 ⁴	106	1.45%	-1.67%
19200 bps		f _{CLK} /2 ³	106	1.45%	-1.67%
31250 bps		f _{CLK} /2 ³	65	1.05%	-2.06%
38400 bps		f _{CLK} /2 ²	106	1.45%	-1.67%
76800 bps		f _{CLK} /2	106	1.45%	-1.67%
115200 bps		f _{CLK} /2	70	1.93%	-1.21%

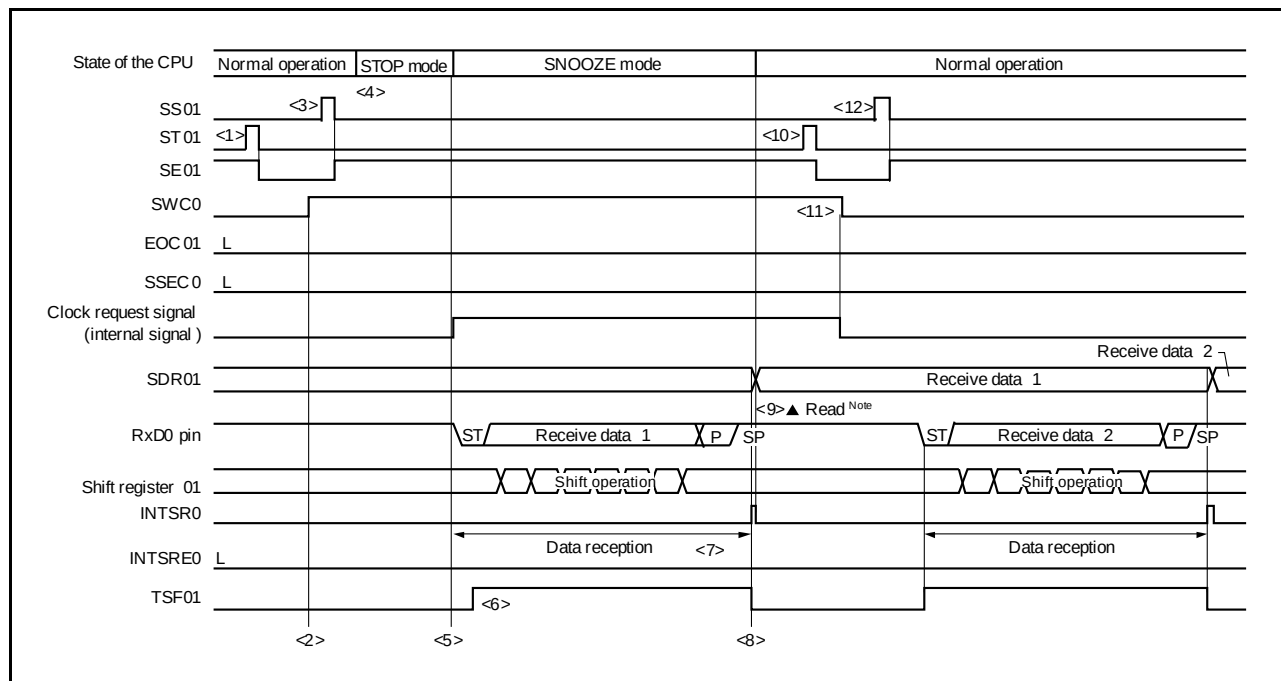
- Note** When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.
- In the case of f_{IH} ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
 - In the case of f_{IH} ±2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOC01 = 0, SSEC0 = 0/1)

Because of the setting of EOC01 = 0, even though a communication error occurs, an error interrupt (INTSRE0) is not generated, regardless of the setting of the SSEC0 bit. A transfer end interrupt (INTSR0) will be generated.

Figure 13 - 91 Timing Chart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1)



Note Read the received data when SWC0 = 1.

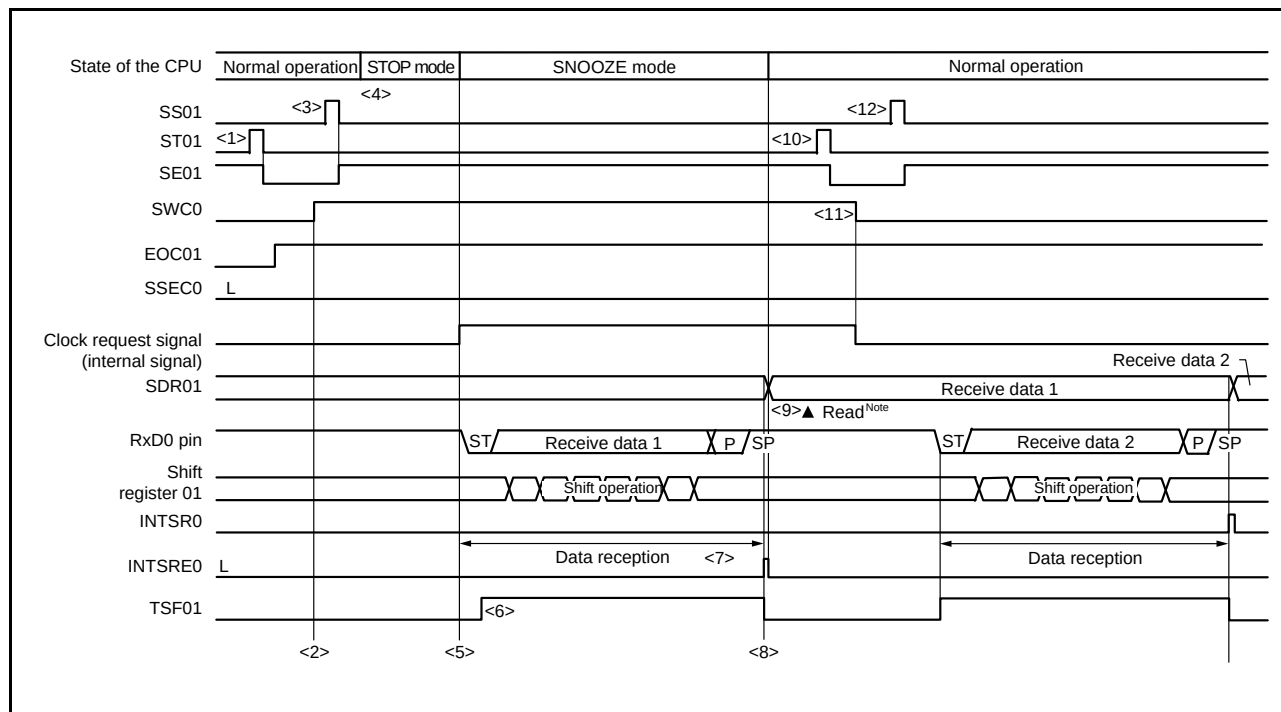
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST01 bit to 1 (the SE01 bit is cleared and the operation stops). After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).

Remark <1> to <12> in the figure correspond to <1> to <12> in Figure 13 - 93 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0).

(2) SNOOZE mode operation (EOC01 = 1, SSEC0 = 0: Error interrupt (INTSRE0) generation is enabled)

Because EOC01 = 1 and SSEC0 = 0, an error interrupt (INTSRE0) is generated when a communication error occurs.

Figure 13 - 92 Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 0)

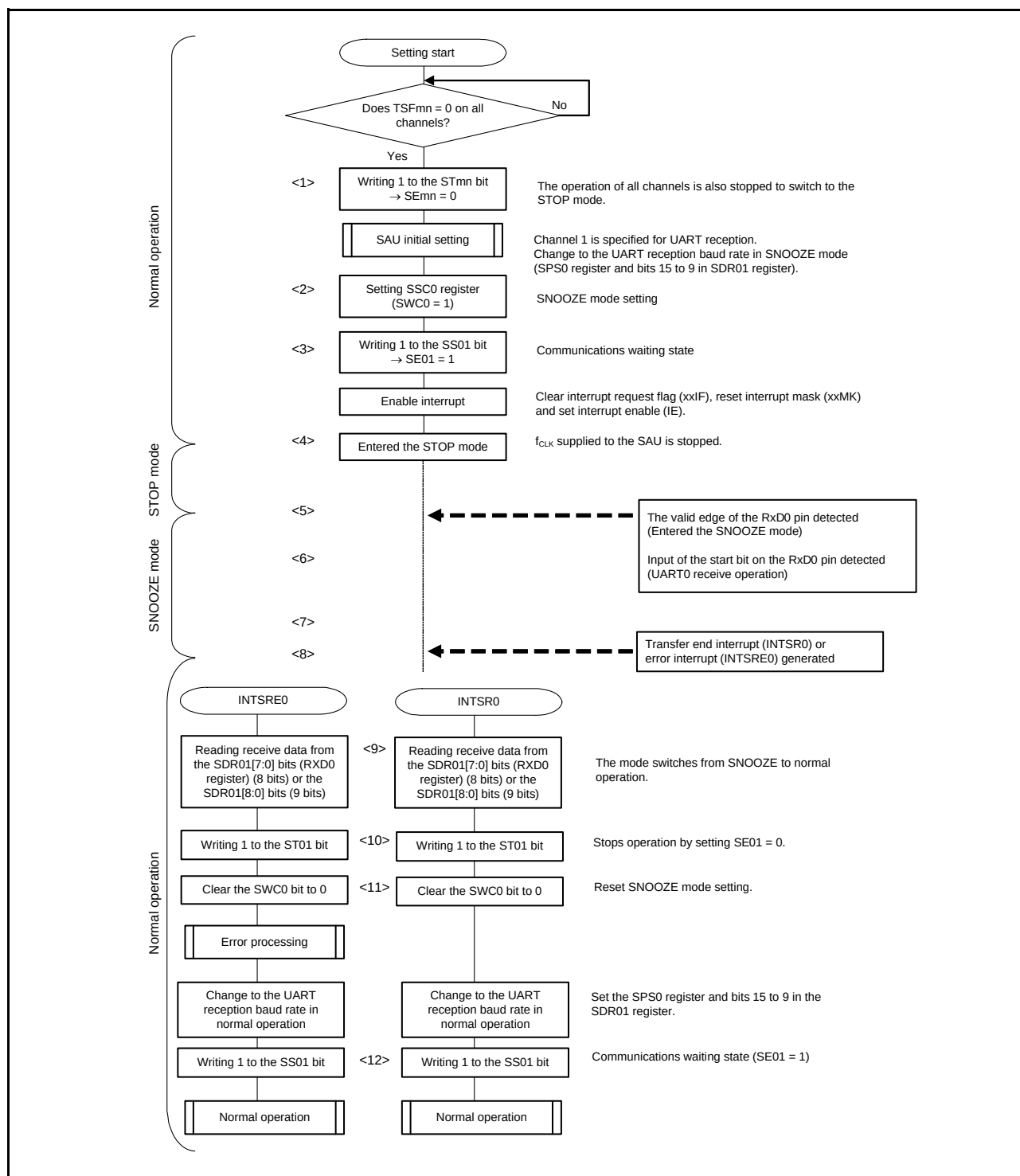


Note Read the received data when SWC0 = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST01 bit to 1 (the SE01 bit is cleared and the operation stops). After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).

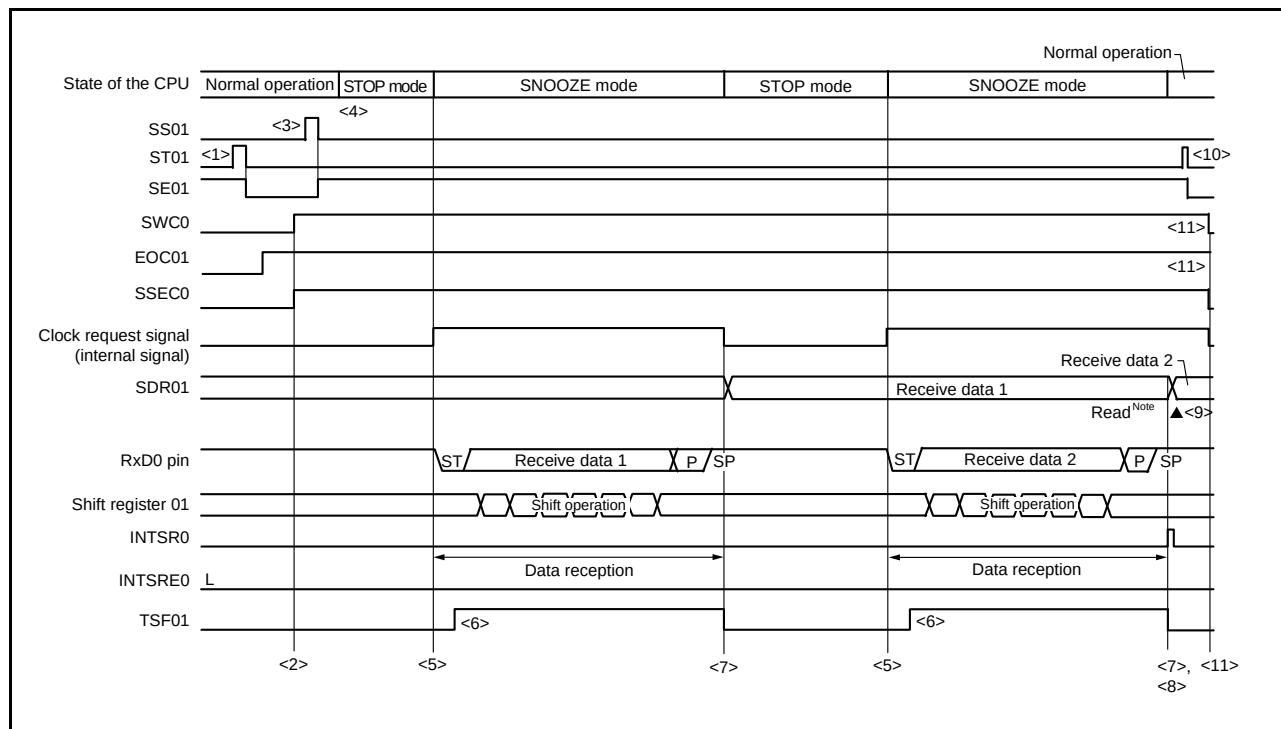
Remark <1> to <12> in the figure correspond to <1> to <12> in Figure 13 - 93 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0).

Figure 13 - 93 Flowchart of SNOOZE Mode Operation (EOC01 = 0, SSEC0 = 0/1 or EOC01 = 1, SSEC0 = 0)



- (3) SNOOZE mode operation (EOC01 = 1, SSEC0 = 1: Error interrupt (INTSRE0) generation is stopped)
Because EOC01 = 1 and SSEC0 = 1, an error interrupt (INTSRE0) is not generated when a communication error occurs.

Figure 13 - 94 Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1)



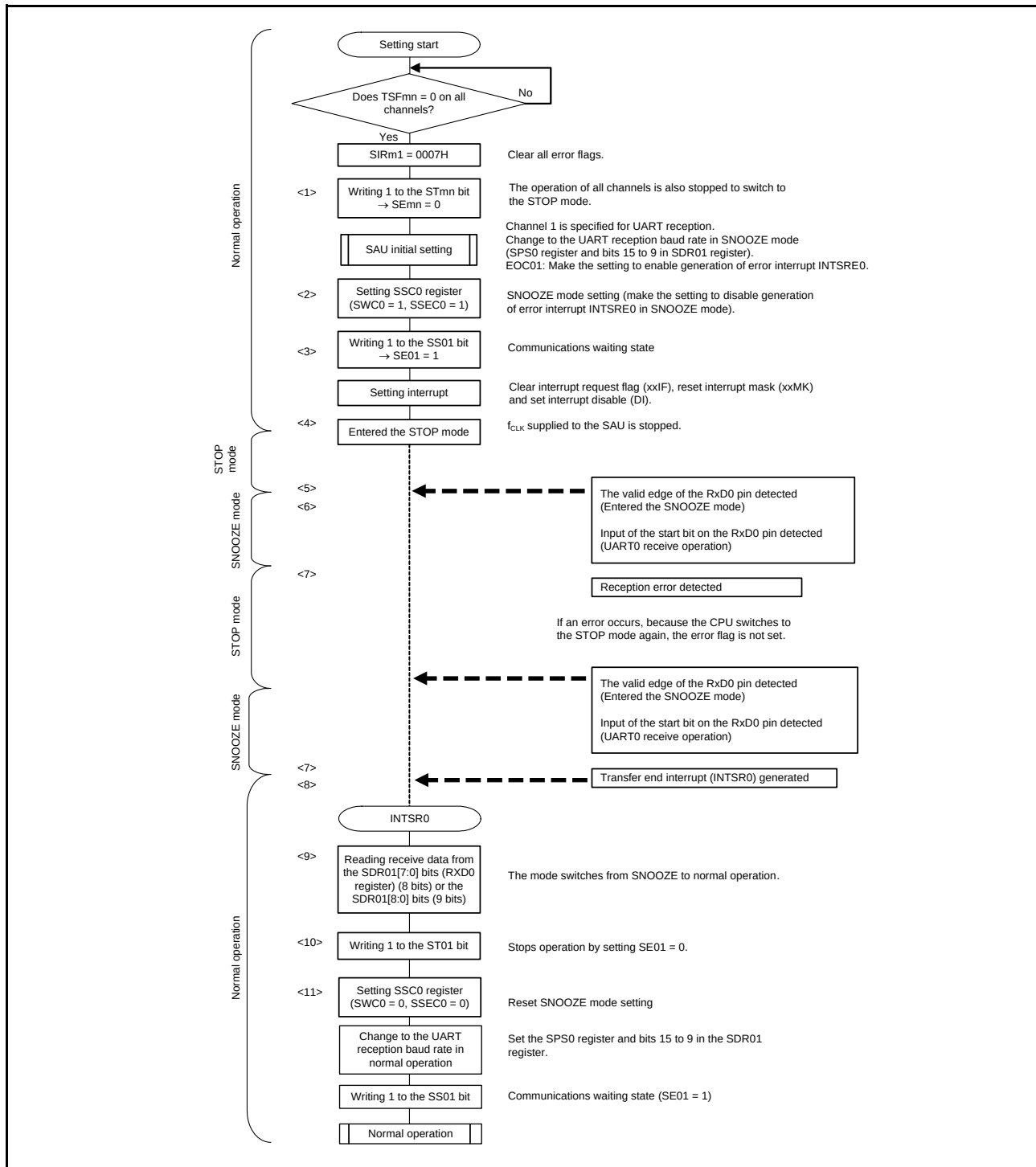
Note Read the received data when SWC0 = 1.

Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the ST01 bit to 1 (the SE01 bit is cleared and the operation stops). After the receive operation completes, also clear the SWC0 bit to 0 (SNOOZE mode release).

Caution 2. If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OV01 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, and OV01 flags before setting the SWC0 bit to 1 and read the value in SDR01[7:0] (RxD0 register) (8 bits) or SDR01[8:0] (9 bits).

Remark <1> to <11> in the figure correspond to <1> to <11> in Figure 13 - 95 Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1).

Figure 13 - 95 Flowchart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1)



Caution If a parity error, framing error, or overrun error occurs while the SSEC0 bit is set to 1, the PEF01, FEF01, or OVFO1 flag is not set and an error interrupt (INTSRE0) is not generated. Therefore, when the setting of SSEC0 = 1 is made, clear the PEF01, FEF01, and OVFO1 flags before setting the SWCm bit to 1 and read the value in SDR01[7:0] (RxD0 register) (8 bits) or SDR01[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 13 - 94 Timing Chart of SNOOZE Mode Operation (EOC01 = 1, SSEC0 = 1).

Remark 2. 16-pin products: m = 0; n = 0, 1
20- to 25-pin products: m = 0; n = 0 to 3
36- to 48-pin products: m = 0, 1; n = 0 to 3

13.6.4 Calculating Baud Rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART2) communication can be calculated by the following expressions.

$$(\text{Baud rate}) = \{\text{Operating clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9]+1) \div 2 [\text{bps}]$$

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The operating clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 13 - 6 Selection of Operating Clock For UART

SMRmn Register	SPSm Register								Operating Clock (fCLK) ^{Note}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		fCLK = 32 MHz
0	x	x	x	x	0	0	0	0	fCLK	32 MHz
	x	x	x	x	0	0	0	1	fCLK/2	16 MHz
	x	x	x	x	0	0	1	0	fCLK/2 ²	8 MHz
	x	x	x	x	0	0	1	1	fCLK/2 ³	4 MHz
	x	x	x	x	0	1	0	0	fCLK/2 ⁴	2 MHz
	x	x	x	x	0	1	0	1	fCLK/2 ⁵	1 MHz
	x	x	x	x	0	1	1	0	fCLK/2 ⁶	500 kHz
	x	x	x	x	0	1	1	1	fCLK/2 ⁷	250 kHz
	x	x	x	x	1	0	0	0	fCLK/2 ⁸	125 kHz
	x	x	x	x	1	0	0	1	fCLK/2 ⁹	62.5 kHz
	x	x	x	x	1	0	1	0	fCLK/2 ¹⁰	31.25 kHz
	x	x	x	x	1	0	1	1	fCLK/2 ¹¹	15.63 kHz
	x	x	x	x	1	1	0	0	fCLK/2 ¹²	7.81 kHz
	x	x	x	x	1	1	0	1	fCLK/2 ¹³	3.91 kHz
	x	x	x	x	1	1	1	0	fCLK/2 ¹⁴	1.95 kHz
	x	x	x	x	1	1	1	1	fCLK/2 ¹⁵	977 Hz
1	0	0	0	0	x	x	x	x	fCLK	32 MHz
	0	0	0	1	x	x	x	x	fCLK/2	16 MHz
	0	0	1	0	x	x	x	x	fCLK/2 ²	8 MHz
	0	0	1	1	x	x	x	x	fCLK/2 ³	4 MHz
	0	1	0	0	x	x	x	x	fCLK/2 ⁴	2 MHz
	0	1	0	1	x	x	x	x	fCLK/2 ⁵	1 MHz
	0	1	1	0	x	x	x	x	fCLK/2 ⁶	500 kHz
	0	1	1	1	x	x	x	x	fCLK/2 ⁷	250 kHz
	1	0	0	0	x	x	x	x	fCLK/2 ⁸	125 kHz
	1	0	0	1	x	x	x	x	fCLK/2 ⁹	62.5 kHz
	1	0	1	0	x	x	x	x	fCLK/2 ¹⁰	31.25 kHz
	1	0	1	1	x	x	x	x	fCLK/2 ¹¹	15.63 kHz
	x	x	x	x	1	1	0	0	fCLK/2 ¹²	7.81 kHz
	x	x	x	x	1	1	0	1	fCLK/2 ¹³	3.91 kHz
	x	x	x	x	1	1	1	0	fCLK/2 ¹⁴	1.95 kHz
	x	x	x	x	1	1	1	1	fCLK/2 ¹⁵	977 Hz
Other than above									Setting prohibited	

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. x: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART2) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$(\text{Baud rate error}) = (\text{Calculated baud rate value}) \div (\text{Target baud rate}) \times 100 - 100 [\%]$$

Here is an example of setting a UART baud rate at fCLK = 32 MHz.

UART Baud Rate (Target Baud Rate)	fCLK = 32 MHz			
	Operating Clock (fMCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	fCLK/2 ⁹	103	300.48 bps	+0.16%
600 bps	fCLK/2 ⁸	103	600.96 bps	+0.16%
1200 bps	fCLK/2 ⁷	103	1201.92 bps	+0.16%
2400 bps	fCLK/2 ⁶	103	2403.85 bps	+0.16%
4800 bps	fCLK/2 ⁵	103	4807.69 bps	+0.16%
9600 bps	fCLK/2 ⁴	103	9615.38 bps	+0.16%
19200 bps	fCLK/2 ³	103	19230.8 bps	+0.16%
31250 bps	fCLK/2 ³	63	31250.0 bps	±0.0%
38400 bps	fCLK/2 ²	103	38461.5 bps	+0.16%
76800 bps	fCLK/2	103	76923.1 bps	+0.16%
153600 bps	fCLK	103	153846 bps	+0.16%
312500 bps	fCLK	50	313725.5 bps	+0.39%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART2) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **13.6.4 (1) Baud rate calculation expression.**)

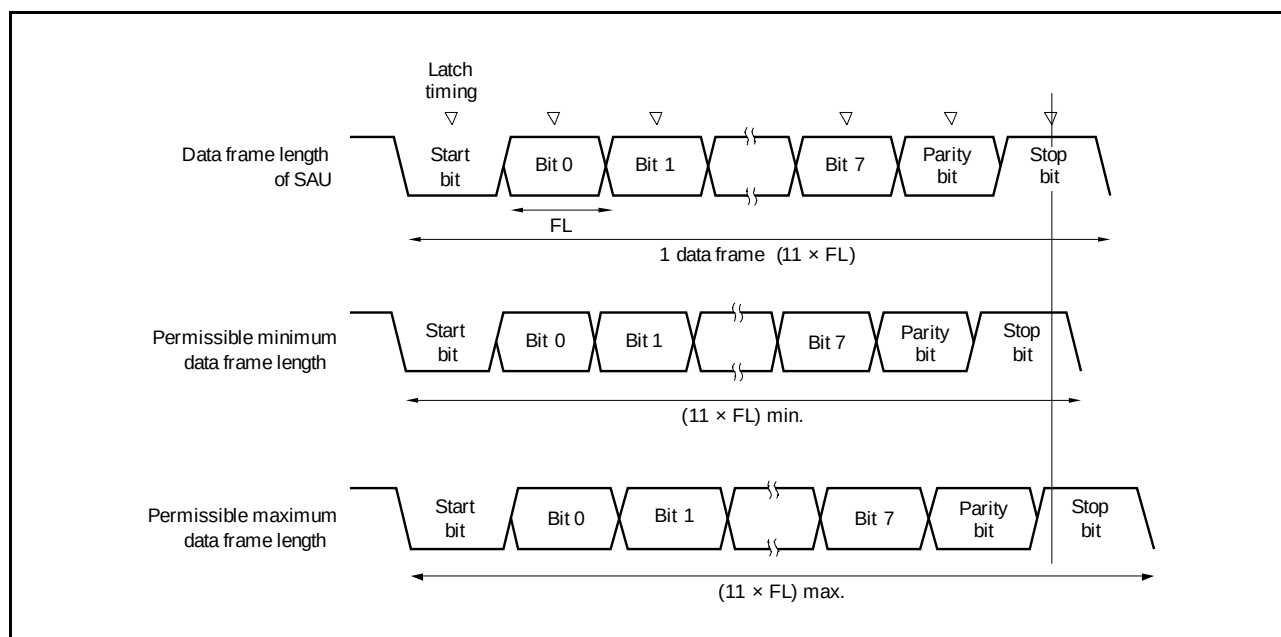
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 13 - 96 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 13 - 96**, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

13.6.5 Procedure for Processing Errors that Occurred during UART (UART0 to UART2) Communication

The procedure for processing errors that occurred during UART (UART0 to UART2) communication is described in **Table 13 - 7** and **Table 13 - 8**.

Table 13 - 7 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	→ The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 13 - 8 Processing Procedure in Case of Framing Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	→ The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1.	→ The SEMn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1.	→ The SEMn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

13.7 LIN Communication Operation

13.7.1 LIN Transmission

Of UART transmission, UART2 of the 30-, 32-, 36-, 40-, 44-, and 48-pin products support LIN communication.
For LIN transmission, channel 0 of unit 1 is used.

UART	UART0	UART1	UART2
Support of LIN communication	Not supported	Not supported	Supported
Target channel	—	—	Channel 0 of SAU1
Pins used	—	—	TxD2
Interrupt	—	—	INTST2
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	8 bits		
Transfer rate ^{Note}	Max. $f_{MCK}/6$ [bps] (SDR10[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit		
Data direction	LSB first		

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for Ta = -40 to +105°C**. In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

Remark f_{MCK} : Operating clock frequency of target channel
 f_{CLK} : System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

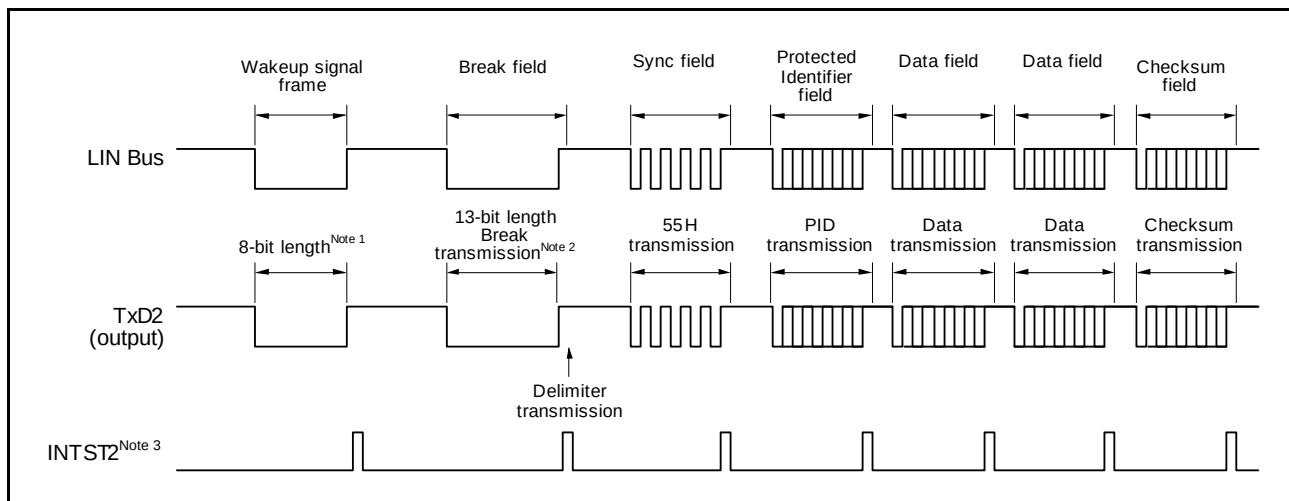
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within $\pm 15\%$, communication can be established.

Figure 13 - 97 outlines a transmission operation of LIN.

Figure 13 - 97 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

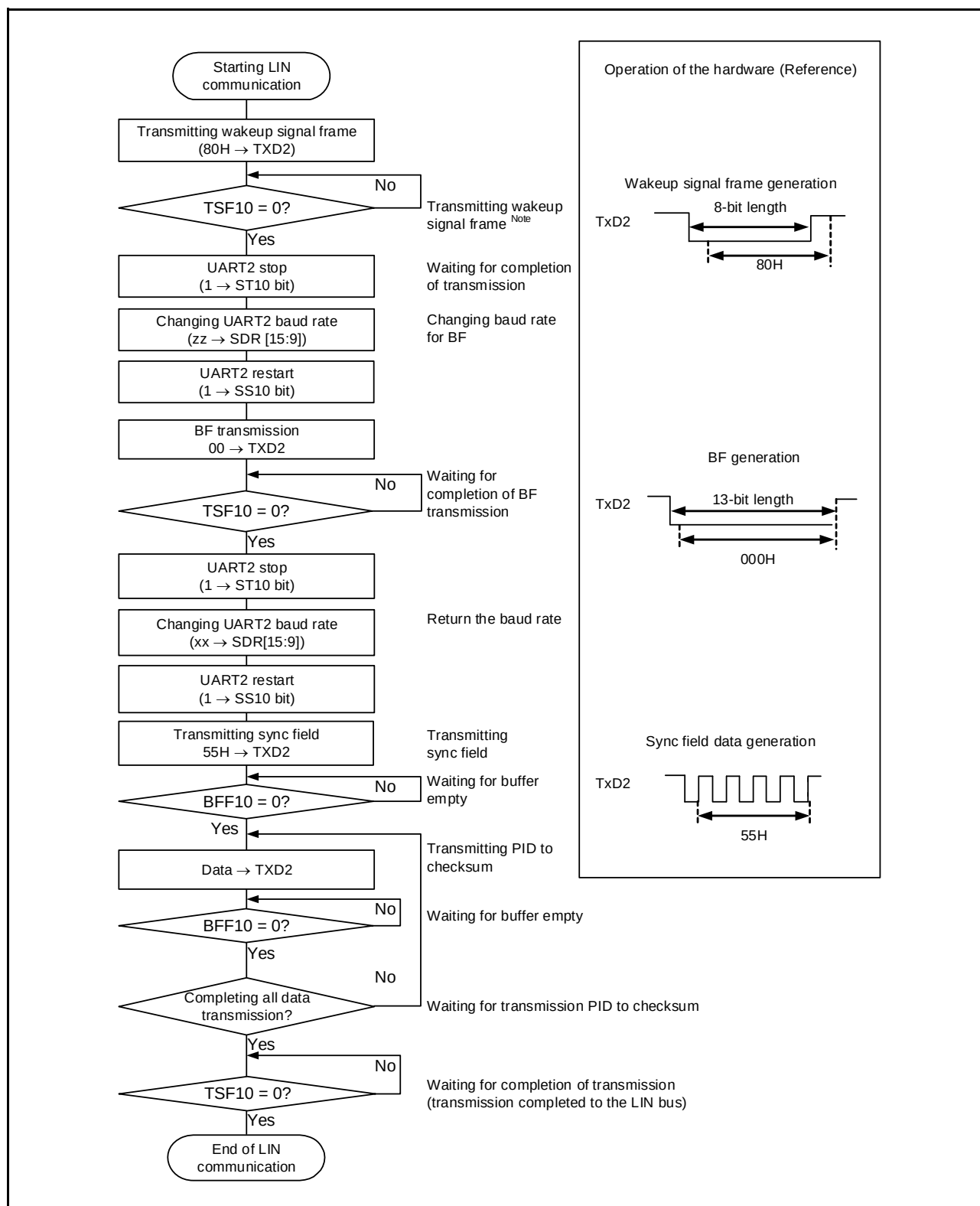
$$(\text{Baud rate of break field}) = 9/13 \times N$$

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST2 is output upon completion of transmission. INTST2 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 13 - 98 Flowchart for LIN Transmission



Note This is only required if the LIN-bus is being started from sleep mode.

Remark This flow assumes that the initial setting of the UART is completed and transmission is enabled.

13.7.2 LIN Reception

Of UART reception, UART2 of the 30-, 32-, 36-, 40-, 44-, and 48-pin products support LIN communication.
For LIN reception, channel 1 of unit 1 is used.

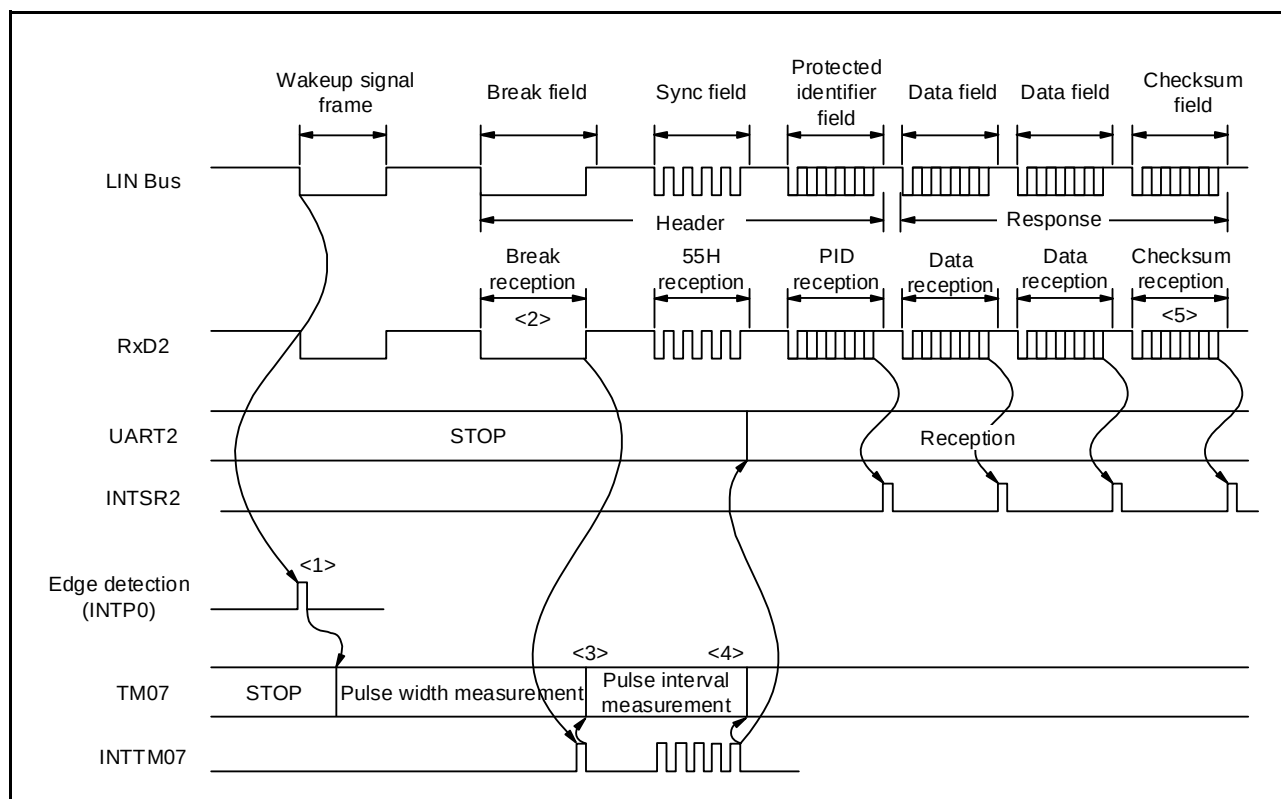
UART	UART0	UART1	UART2
Support of LIN communication	Not supported	Not supported	Supported
Target channel	—	—	Channel 1 of SAU1
Pins used	—	—	RxD2
Interrupt	—	—	INTSR2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	—	—	INTSRE2
Error detection flag	- Framing Error detection flag (FEF11) - Overrun Error detection flag (OVF11)		
Transfer data length	8 bits		
Transfer rate ^{Note}	Max. $f_{MCK}/6$ [bps] (SDR11[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit (The parity bit is not checked.)		
Stop bit	Check the first bit		
Data direction	LSB first		

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$** .

Remark f_{MCK} : Operating clock frequency of target channel
 f_{CLK} : System clock frequency

Figure 13 - 99 outlines a reception operation of LIN.

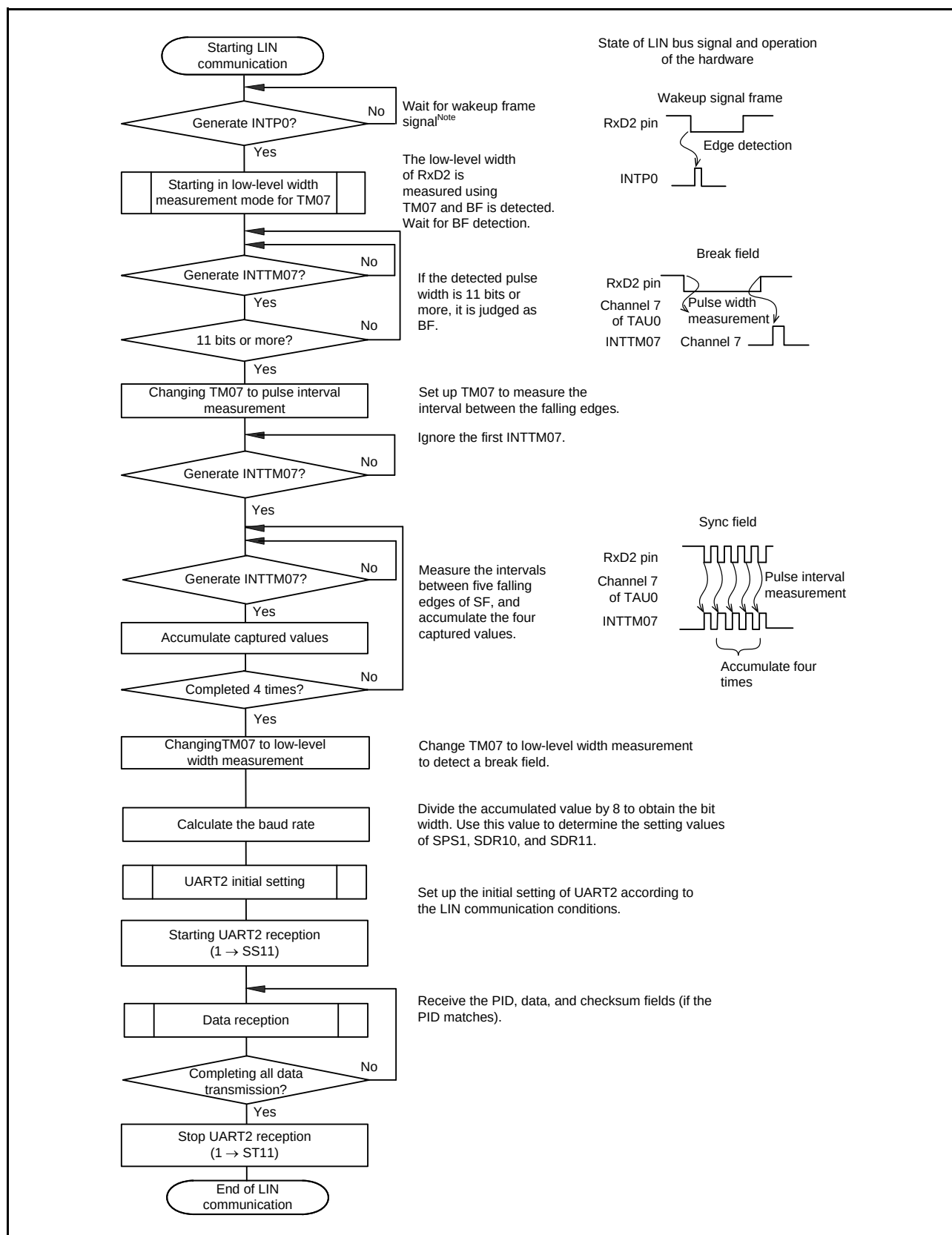
Figure 13 - 99 Reception Operation of LIN



Here is the flow of reception processing.

- <1> The wakeup signal is detected by detecting an edge on the INTP0 interrupt pin. When the wakeup signal is detected, set TM07 to the pulse width measurement function to measure the low-level width of the BF signal. Then wait for BF signal reception.
- <2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.
- <3> When the BF signal has been received normally, change TM07 to pulse interval measurement and measure the interval between the falling edges of the RxD2 signal in the Sync field four times. For details, see **7.8.4 Operation for input pulse interval measurement**.
- <4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART2 once and adjust (re-set) the baud rate.
- <5> The checksum field should be distinguished by software. In addition, processing to initialize UART2 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 13 - 100 Flowchart of LIN Reception



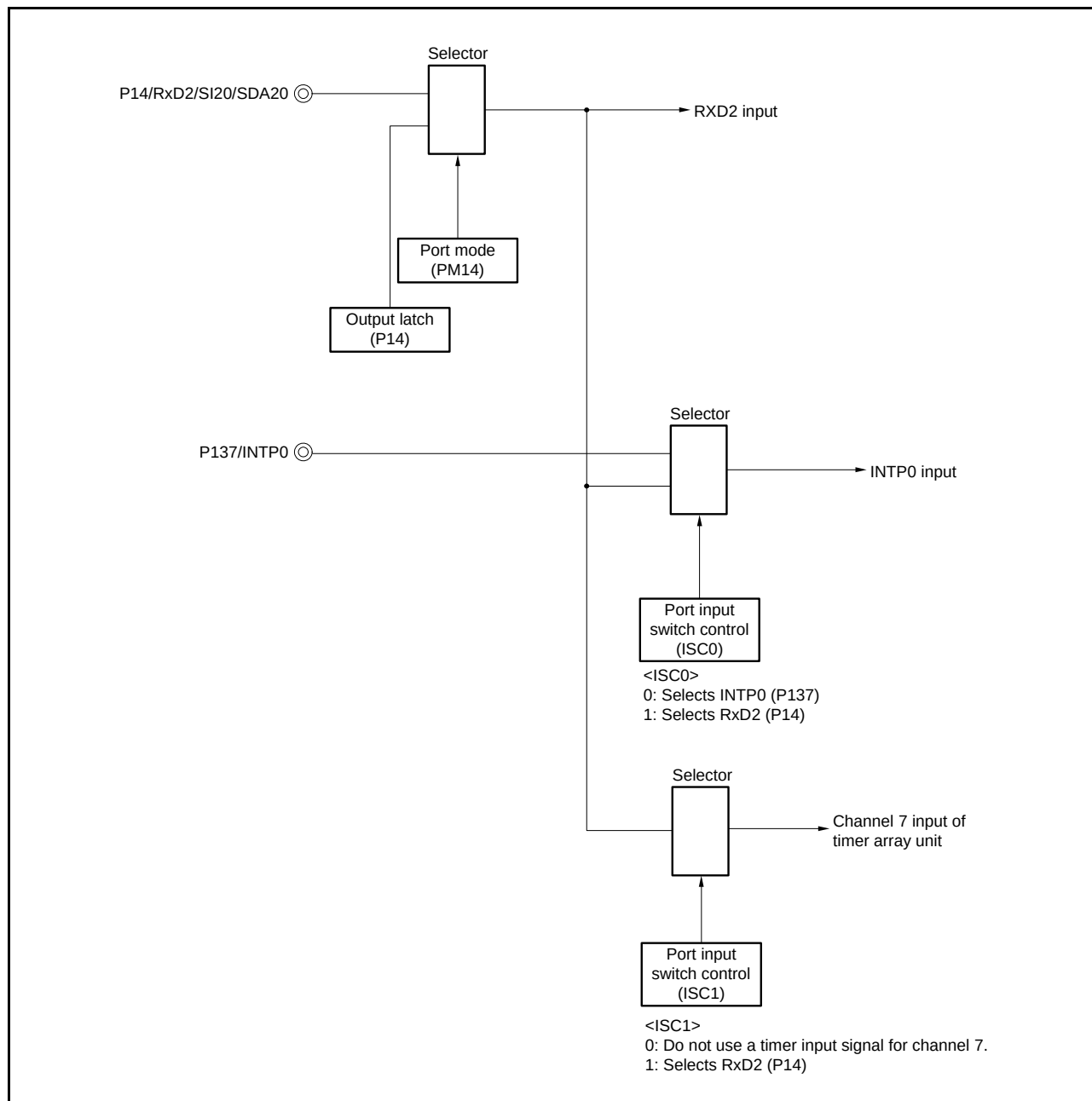
Note This is only required if the LIN bus is in sleep mode.

Figure 13 - 101 and **Figure 13 - 102** show the configuration of ports used for LIN reception.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.

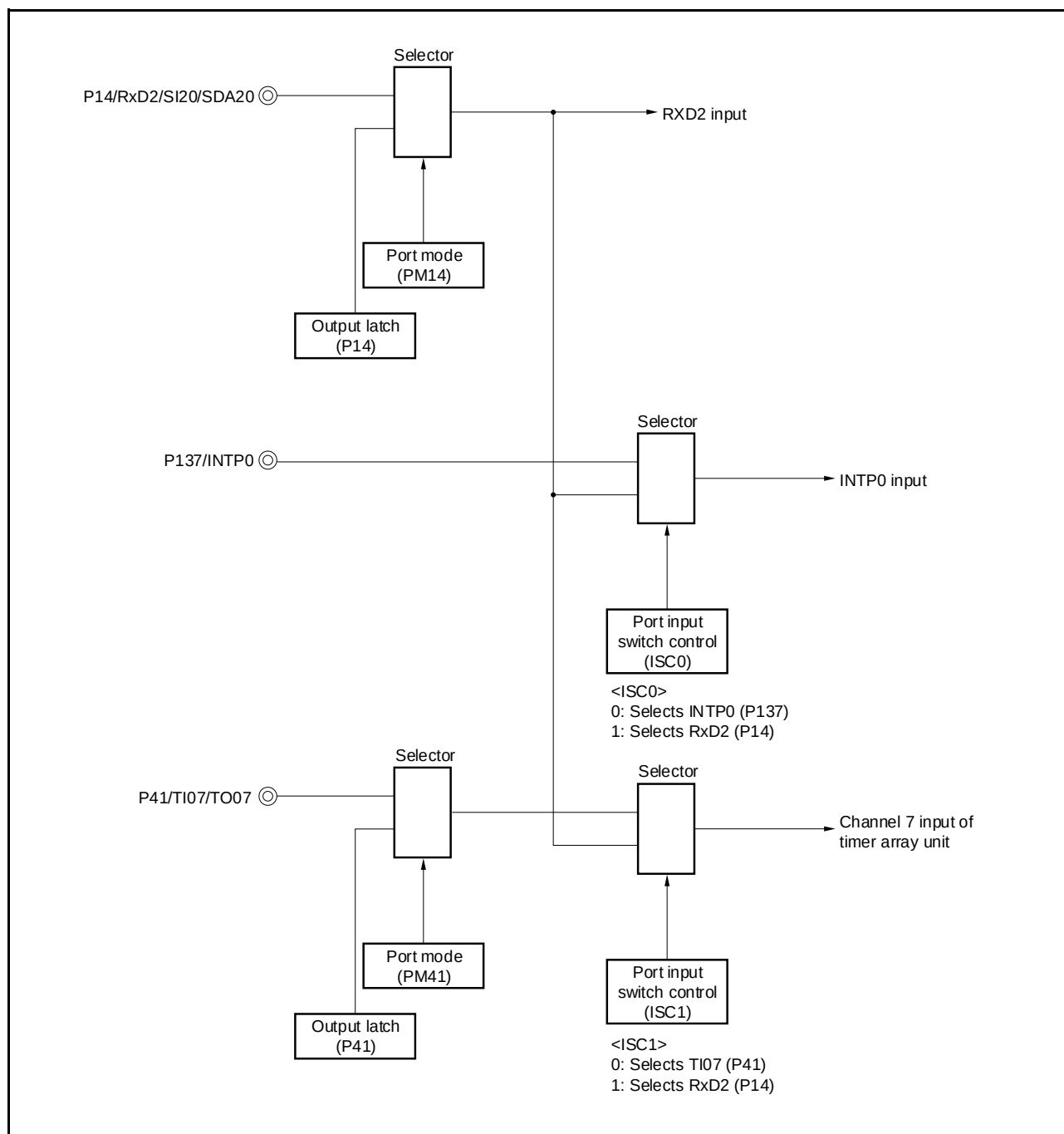
By using the port input switching control (the ISC0 and ISC1 bits), the signal input to the reception port (RxD2) can be used as an external interrupt (INTP0) or sent to the timer array unit without additional external connections.

Figure 13 - 101 Port Configuration for Reception of LIN in 30-, 32-, 36-, and 40-pin Products



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13 - 22**.)

Figure 13 - 102 Port Configuration for Reception of LIN in 44- and 48-pin Products



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13 - 22.**)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit; Baud rate error detection, break field detection.
Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect a baud rate error. (The interval of the edge input to RxD2 is measured in the capture mode.)
To measure the low-level width to detect the break field (BF).
- Channels 0 and 1 (UART2) of serial array unit 1 (SAU1)

13.8 Operation of Simplified I²C (IIC00, IIC01, IIC11, IIC20, and IIC21) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I²C bus line

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function **Note** and ACK detection function
- Data length of 8 bits

(When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)

- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See **13.8.3 (2) Processing flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

Channels 0, 1, and 3 of SAU0 and channel 0 and 1 of SAU1 support the simplified I²C (IIC00, IIC01, IIC11, IIC20, and IIC21).

<16-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		IIC11

<20-, 24-, and 25-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

<30- and 32-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	—		—

<36-, 40-, and 44-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

<48-pin products>

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I ² C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21

Simplified I²C (IIC00, IIC01, IIC11, IIC20, and IIC21) handles the following four types of communications.

- Address field transmission (See **13.8.1.**)
- Data transmission (See **13.8.2.**)
- Data reception (See **13.8.3.**)
- Stop condition generation (See **13.8.4.**)

13.8.1 Address Field Transmission

Address field transmission is a transmission operation that first executes in I²C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I ² C	IIC00	IIC01	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error detection flag	ACK error detection flag (PEFmn)				
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)				
Transfer rate ^{Note 2}	Max.f _{MCK} /4 [Hz] (SDR _{mn} [15:9] = 1 or more) f _{MCK} : Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I ² C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)				
Data level	Non-reverse output (default: high level)				
Parity bit	No parity bit				
Stop bit	Appending 1 bit (for ACK transmission/reception timing)				
Data direction	MSB first				

Note 1. To handle communication via simplified I²C, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POM_{xx} = 1) with the port output mode register (POM_{xx}). See **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Alternate Function** for details.

To communicate with an external device operating at a different voltage through IIC00 and IIC20, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POM_{xx} = 1) also for the clock input/output pins (SCL00 and SCL20). See **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers** for details.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for T_A = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

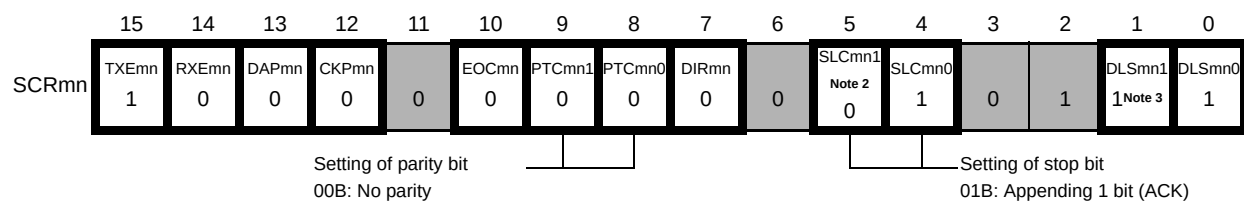
(1) Register setting

Figure 13 - 103 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01, IIC11, IIC20, IIC21)

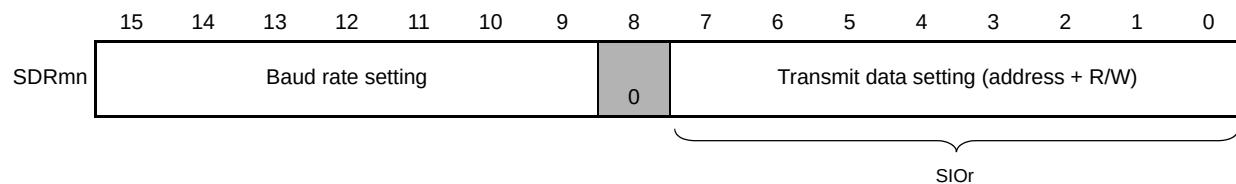
(a) Serial mode register mn (SMRmn)



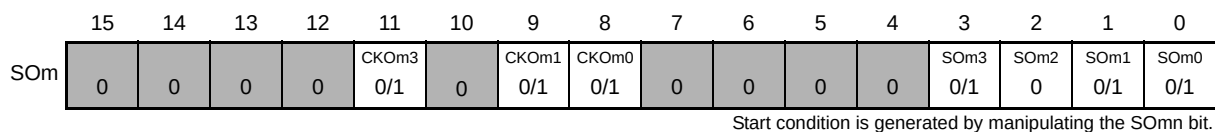
(b) Serial communication operation setting register mn (SCRmn)



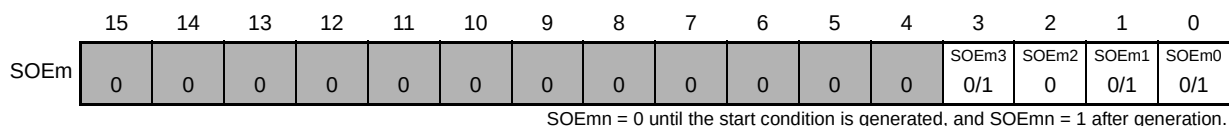
(c) Serial data register mn (SDRmn) (lower 8 bits: SIO_r)



(d) Serial output register m (SOM_m)



(e) Serial output enable register m (SOEm_m)



- (f) Serial channel start register m (SSm) ... Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

SSmn = 0 until the start condition is generated, and SSmn = 1 after generation.

Note 1. This bit is only present in the SMR00, SMR03, and SMR11 registers.

Note 2. This bit is only present in the SCR00 and SCR10 registers.

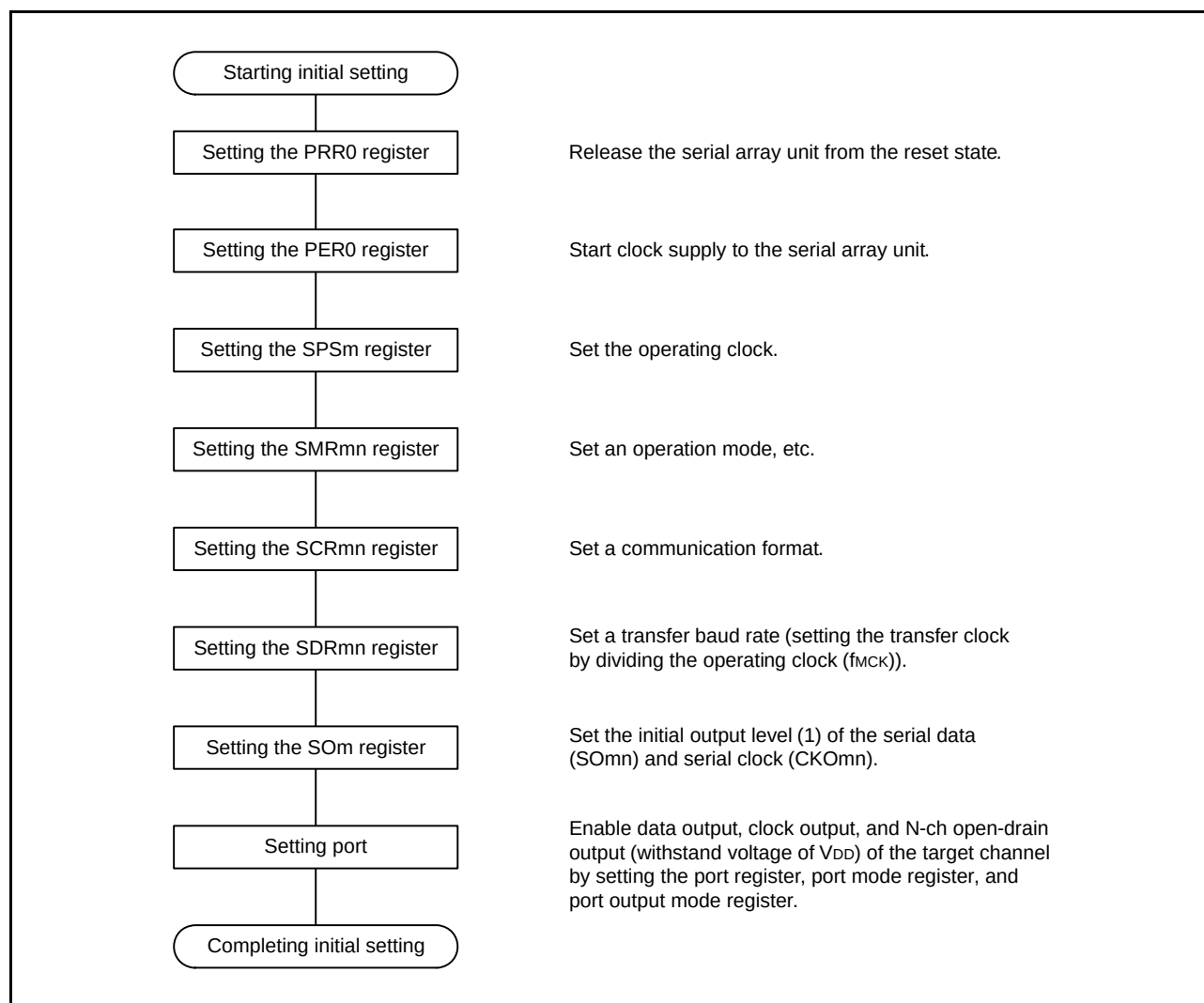
Note 3. This bit is only present in the SCR00 register, and is fixed to 1 in the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the IIC mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

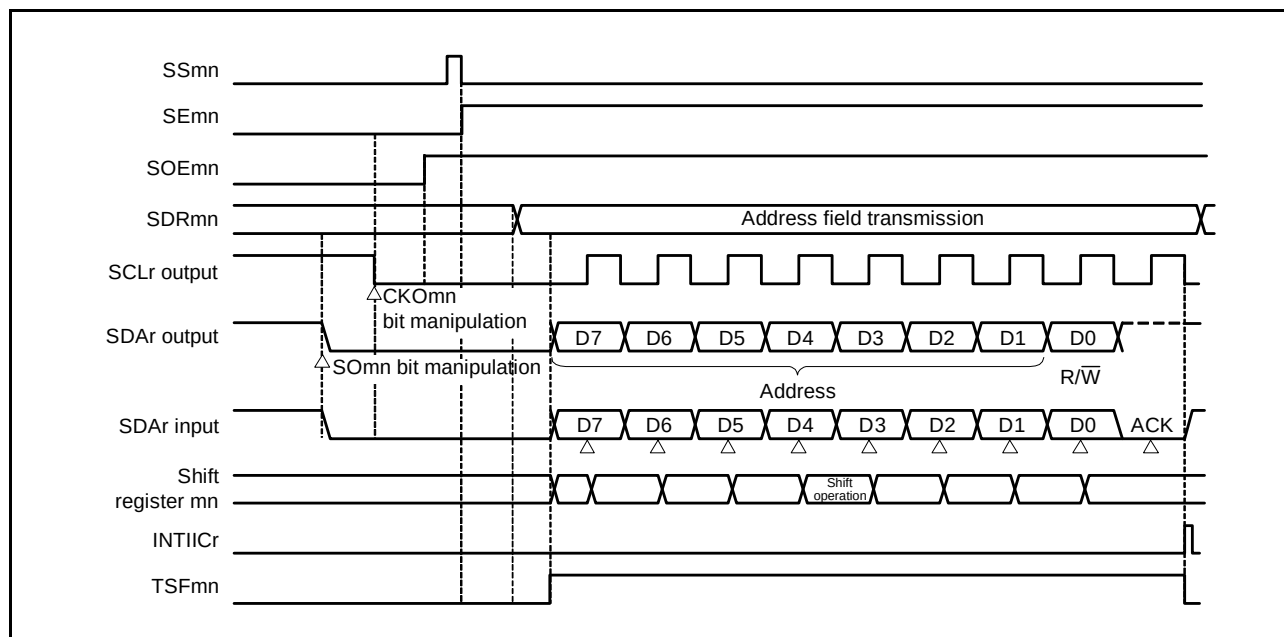
(2) Operation procedure

Figure 13 - 104 Initial Setting Procedure for Simplified I²C Address Field Transmission



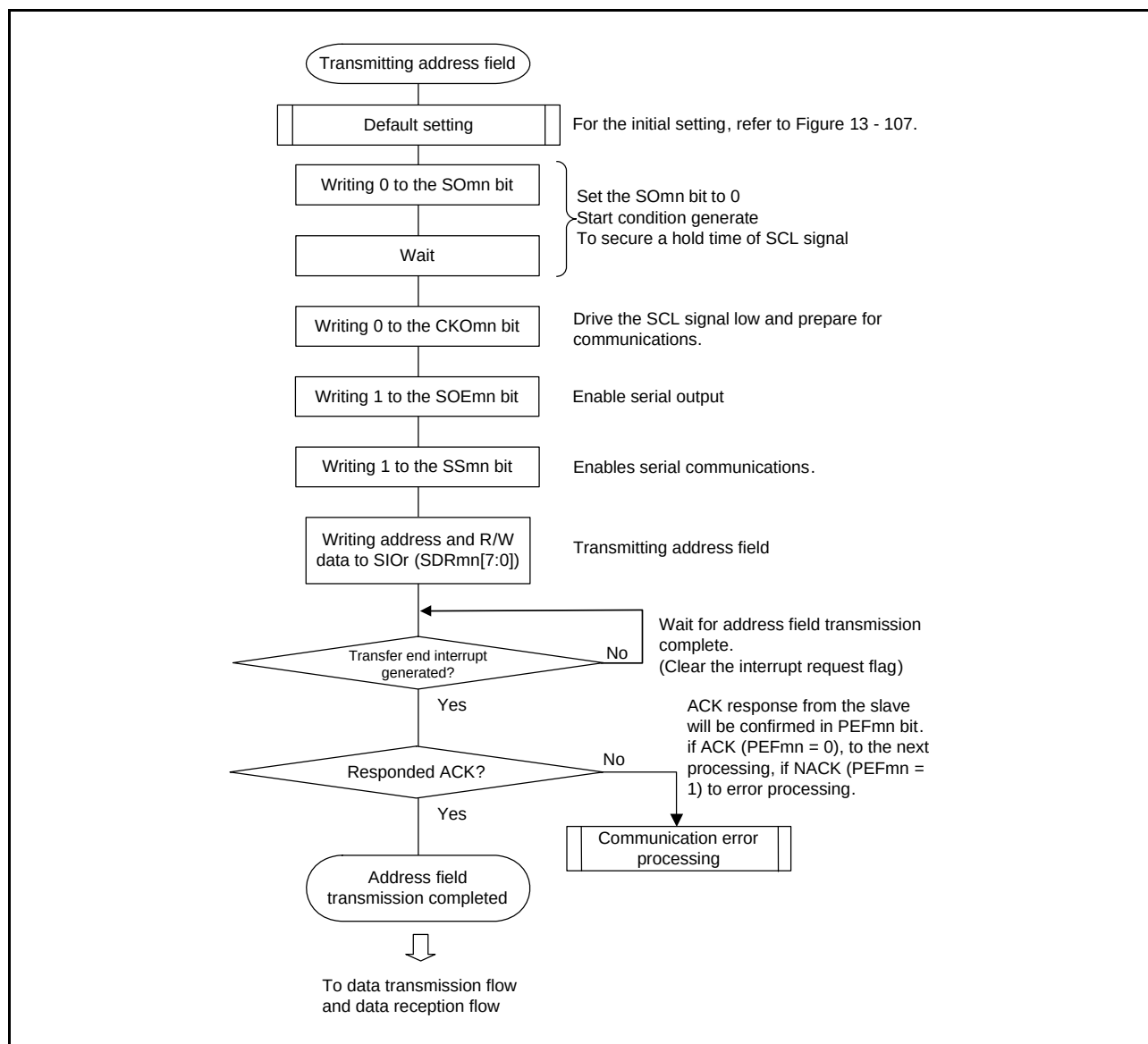
(3) Processing flow

Figure 13 - 105 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 106 Flowchart of Simplified I²C Address Field Transmission



13.8.2 Data Transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I ² C	IIC00	IIC01	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error detection flag	ACK error flag (PEFmn)				
Transfer data length	8 bits				
Transfer rate ^{Note 2}	Max.f _{MCK} /4 [Hz] (SDRmn[15:9] = 1 or more) f _{MCK} : Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I ² C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)				
Data level	Non-reverse output (default: high level)				
Parity bit	No parity bit				
Stop bit	Appending 1 bit (for ACK reception timing)				
Data direction	MSB first				

Note 1. To handle communication via simplified I²C, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Alternate Function**. To communicate with an external device operating at a different voltage through IIC00 and IIC20, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POMxx = 1) also for the clock input/output pins (SCL00 and SCL20). For details, see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for T_A = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

(1) Register setting

Figure 13 - 107 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01, IIC11, IIC20, IIC21)

(a) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0Note 1	0	SISmn0 0Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0Note 2	SLCmn0 1	0	1	DLSmn1 1Note 3	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIO_r) ... During data transmission/reception, valid only lower 8-bits (SIO_r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting ^{Note 4}							0	Transmit data setting							
									SIO _r							

(d) Serial output register m (SOM) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	CKOm3 0/1Note 5	0	CKOm1 0/1Note 5	CKOm0 0/1Note 5	0	0	0	0	SOM3 0/1Note 5	SOM2 0	SOM1 0/1Note 5	SOM0 0/1Note 5

(e) Serial output enable register m (SOEm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 1	SOEm2 0	SOEm1 1	SOEm0 1

(f) Serial channel start register m (SSm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, and SMR11 registers.

Note 2. This bit is only present in the SCR00 and SCR10 registers.

Note 3. This bit is only present in the SCR00 register, and is fixed to 1 in the other registers.

Note 4. Because the setting is completed by address field transmission, setting is not required.

Note 5. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21), mn = 00, 01, 03, 10, 11

Remark 2. ☒: Setting is fixed in the IIC mode, ☐: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Processing flow

Figure 13 - 108 Timing Chart of Data Transmission

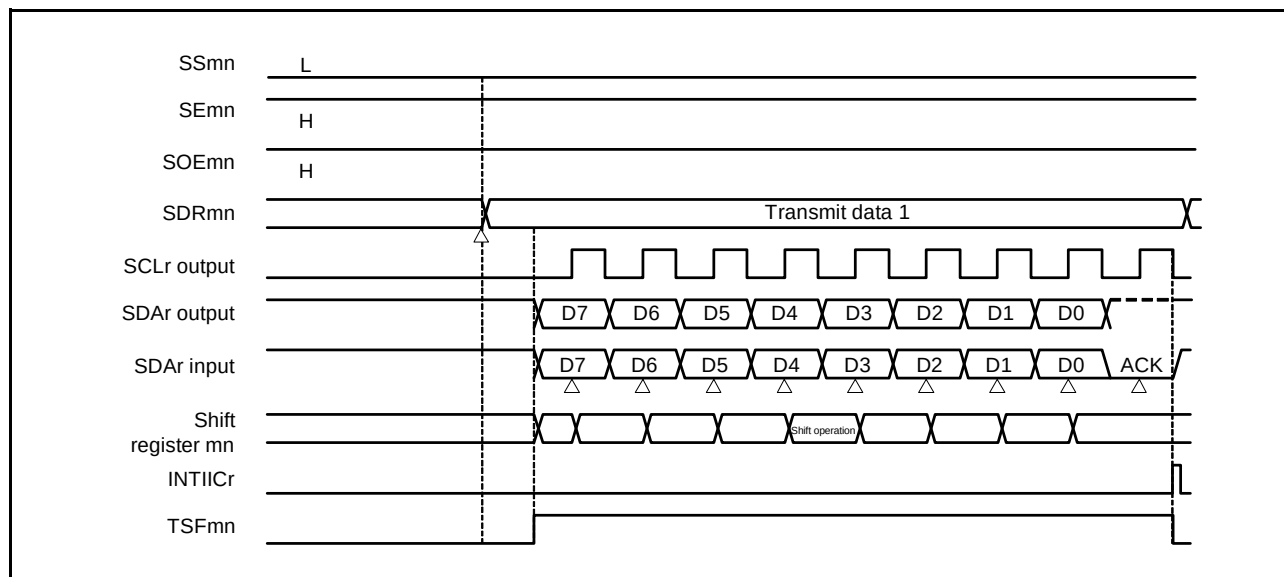
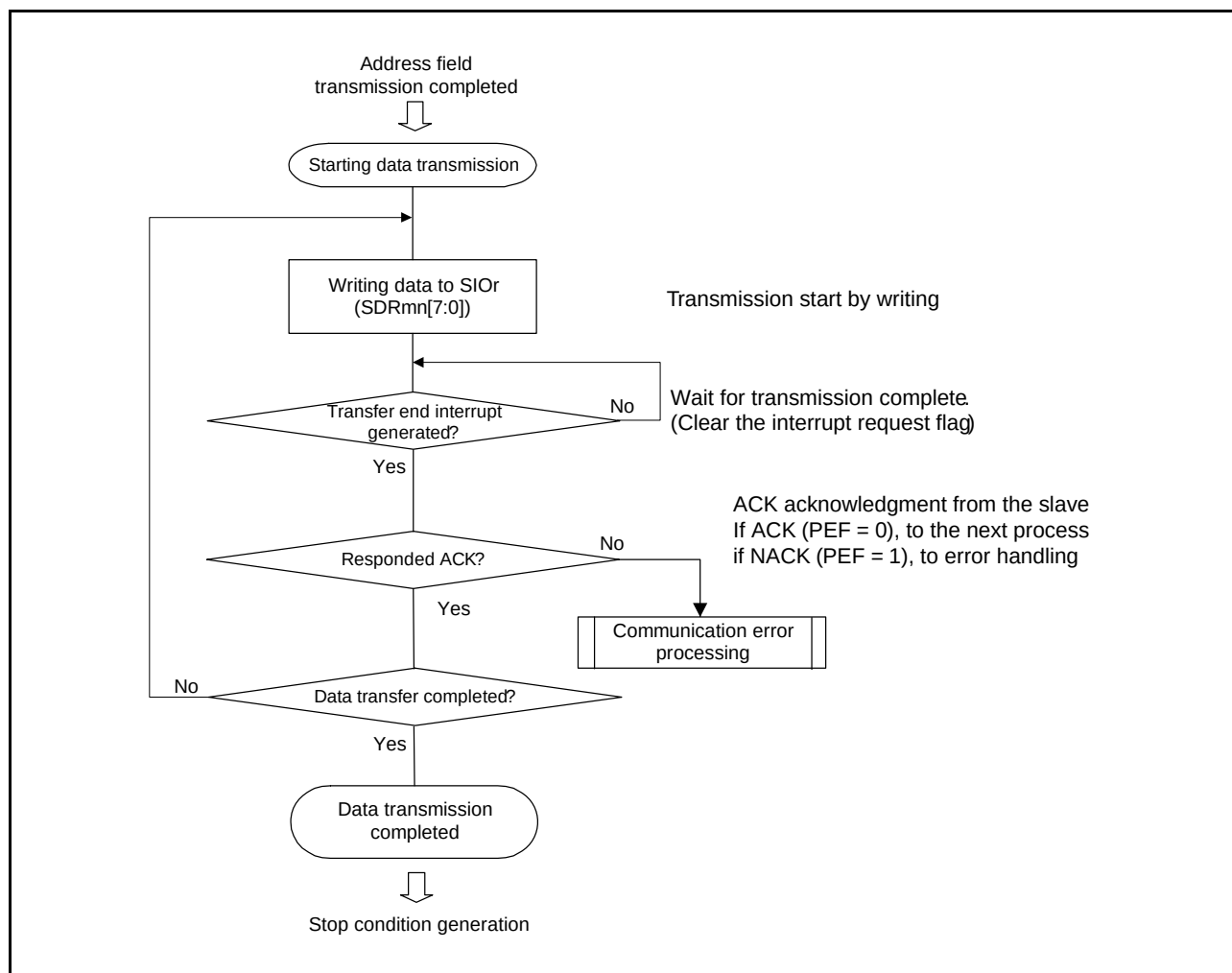


Figure 13 - 109 Flowchart of Simplified I²C Data Transmission



13.8.3 Data Reception

Data reception is an operation to receive data from the target for transfer (slave) after transmission of an address field. After all data are received from the slave, a stop condition is generated and the bus is released.

Simplified I ² C	IIC00	IIC01	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error detection flag	Overrun error detection flag (OVFmn) only				
Transfer data length	8 bits				
Transfer rate ^{Note 2}	Max.f _{MCK} /4 [Hz] (SDRmn[15:9] = 1 or more) f _{MCK} : Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I ² C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)				
Data level	Non-reverse output (default: high level)				
Parity bit	No parity bit				
Stop bit	Appending 1 bit (ACK transmission)				
Data direction	MSB first				

Note 1. To handle communication via simplified I²C, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Alternate Function**. To communicate with an external device operating at a different voltage through IIC00 and IIC20, set the N-ch open-drain output (withstand voltage of V_{DD}) mode (POMxx = 1) also for the clock input/output pins (SCL00 and SCL20). For details, see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 34 Electrical Characteristics for T_A = -40 to +105°C**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

(1) Register setting

Figure 13 - 110 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

(a) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0						STSmn 0Note 1		SISmn0 0Note 1				MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0		EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0		SLCmn1 0Note 2	SLCmn0 1			DLSmn1 1Note 3	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIO_r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting ^{Note 4}							0	Dummy transmit data setting (FFH)							
									SIO _r							

(d) Serial output register m (SOM_m) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM _m	0	0	0	0	CKOm3 0/1Note 5	0	CKOm1 0/1Note 5	CKOm0 0/1Note 5	0	0	0	0	SOM3 0/1Note 5	SOM2 0	SOM1 0/1Note 5	SOM0 0/1Note 5

(e) Serial output enable register m (SOEm_m) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm _m	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 0/1	SOEm2 0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm_m) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm _m	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, and SMR11 registers.

Note 2. This bit is only present in the SCR00 and SCR10 registers.

Note 3. This bit is only present in the SCR00 and SCR01 registers, and is fixed to 1 in the other registers.

Note 4. Because the setting is completed by address field transmission, setting is not required.

Note 5. The value varies depending on the communication data during communication operation.

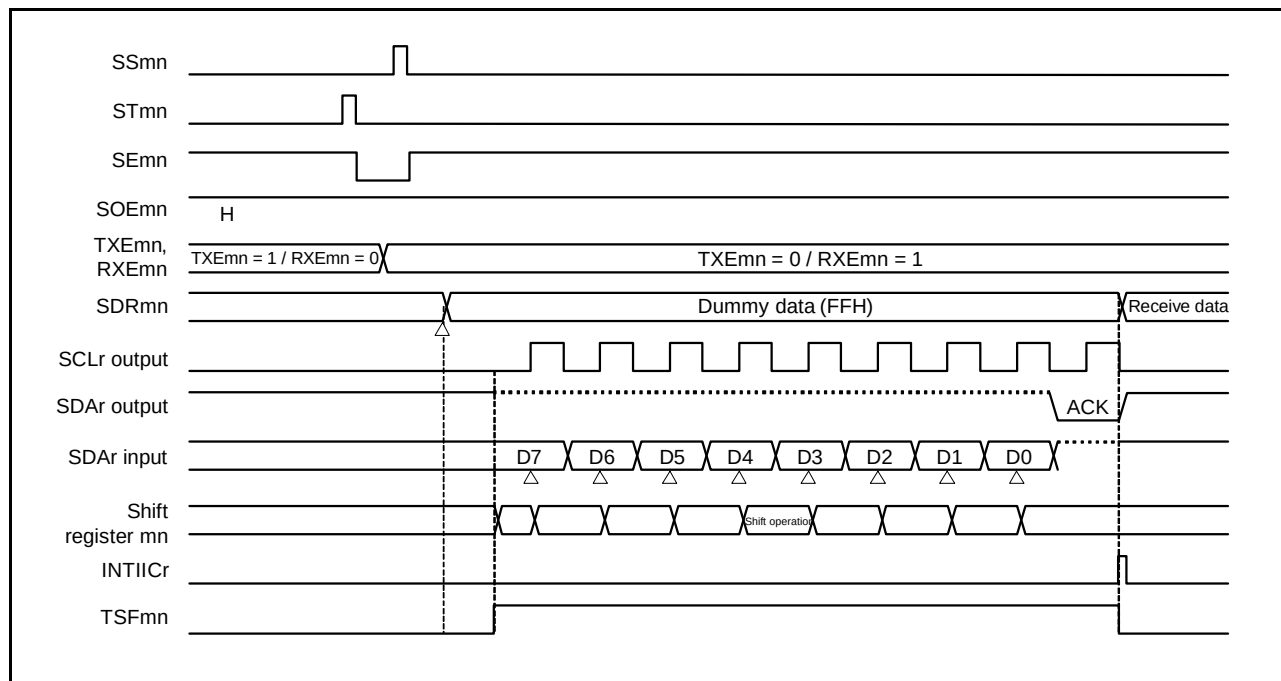
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Remark 2. : Setting is fixed in the IIC mode, : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user.

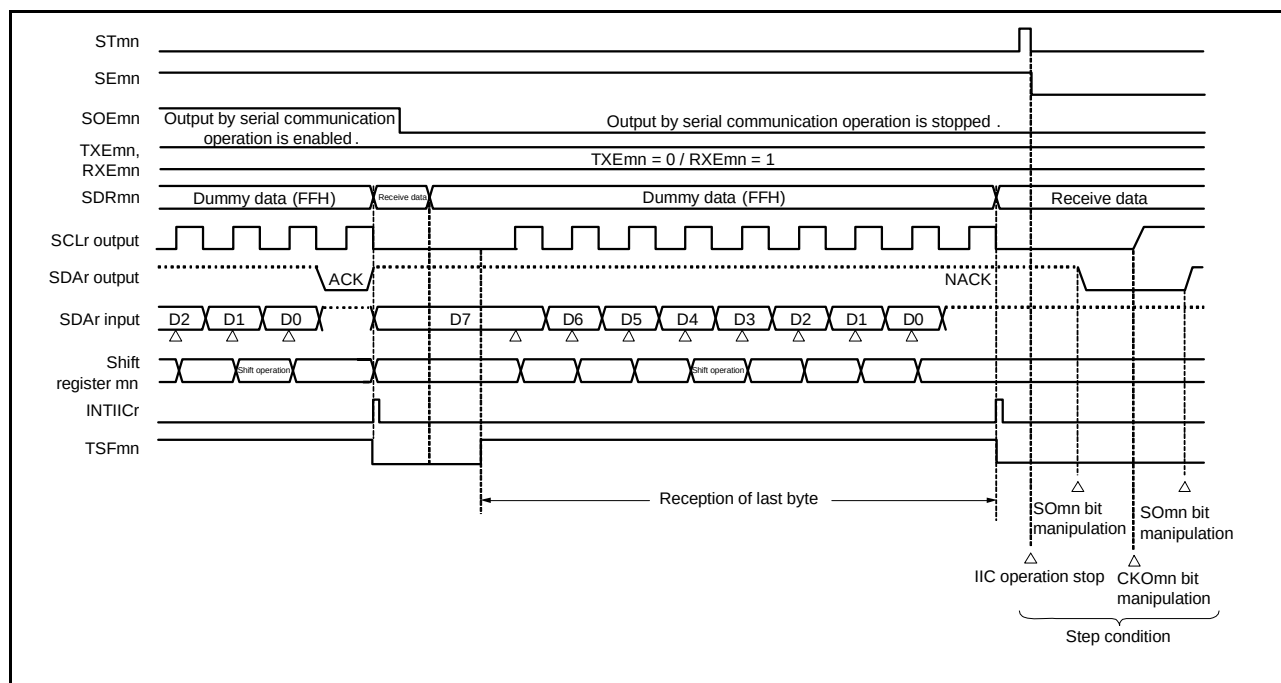
(2) Processing flow

Figure 13 - 111 Timing Chart of Data Reception

(a) When starting data reception

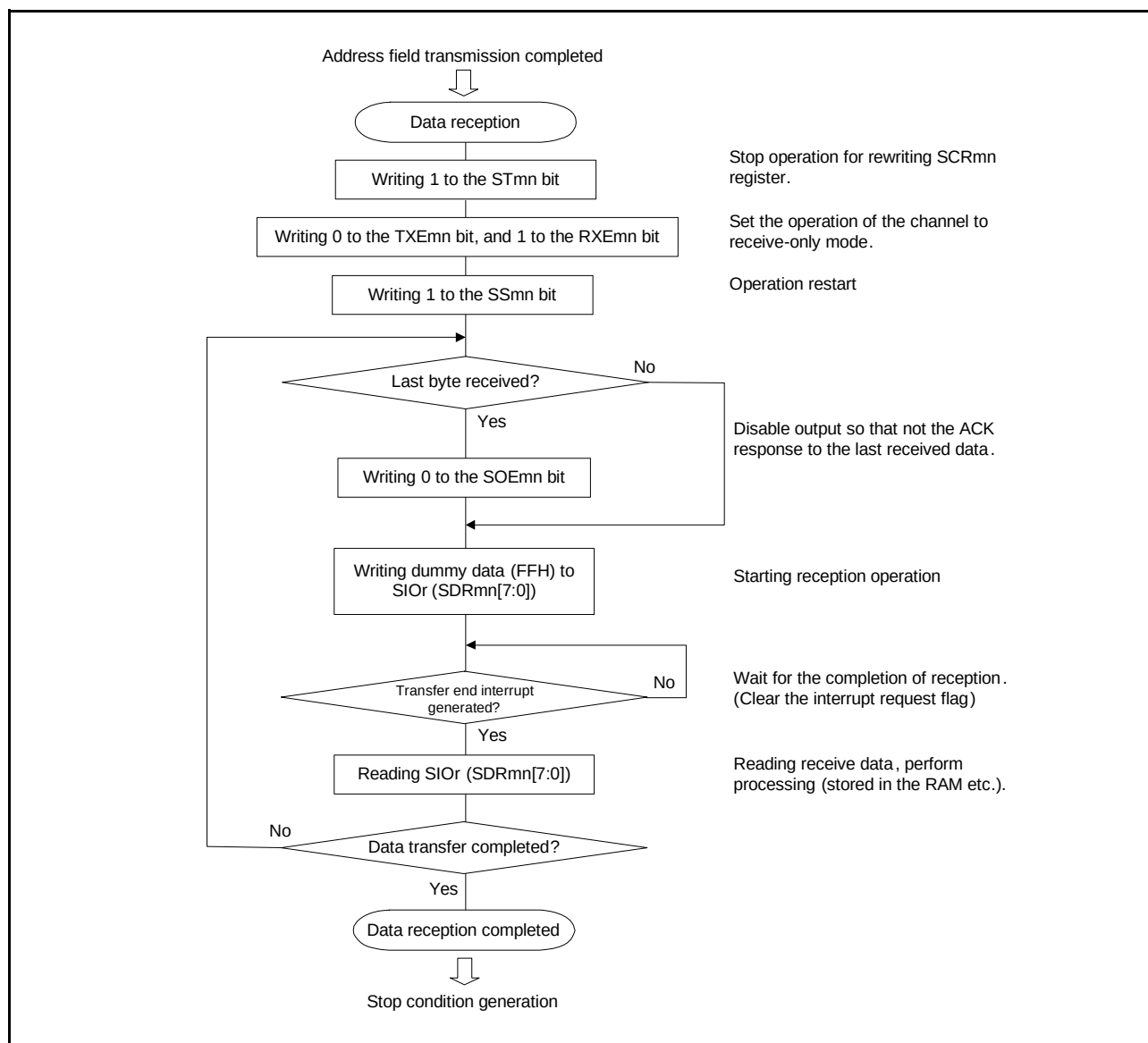


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Figure 13 - 112 Flowchart of Data Reception



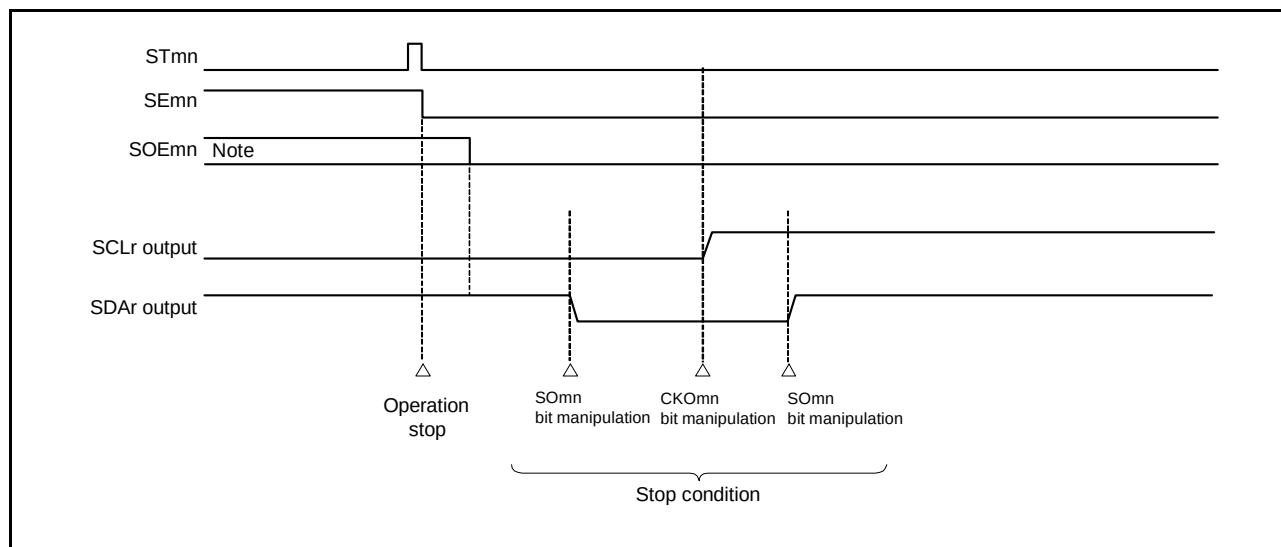
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

13.8.4 Stop Condition Generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

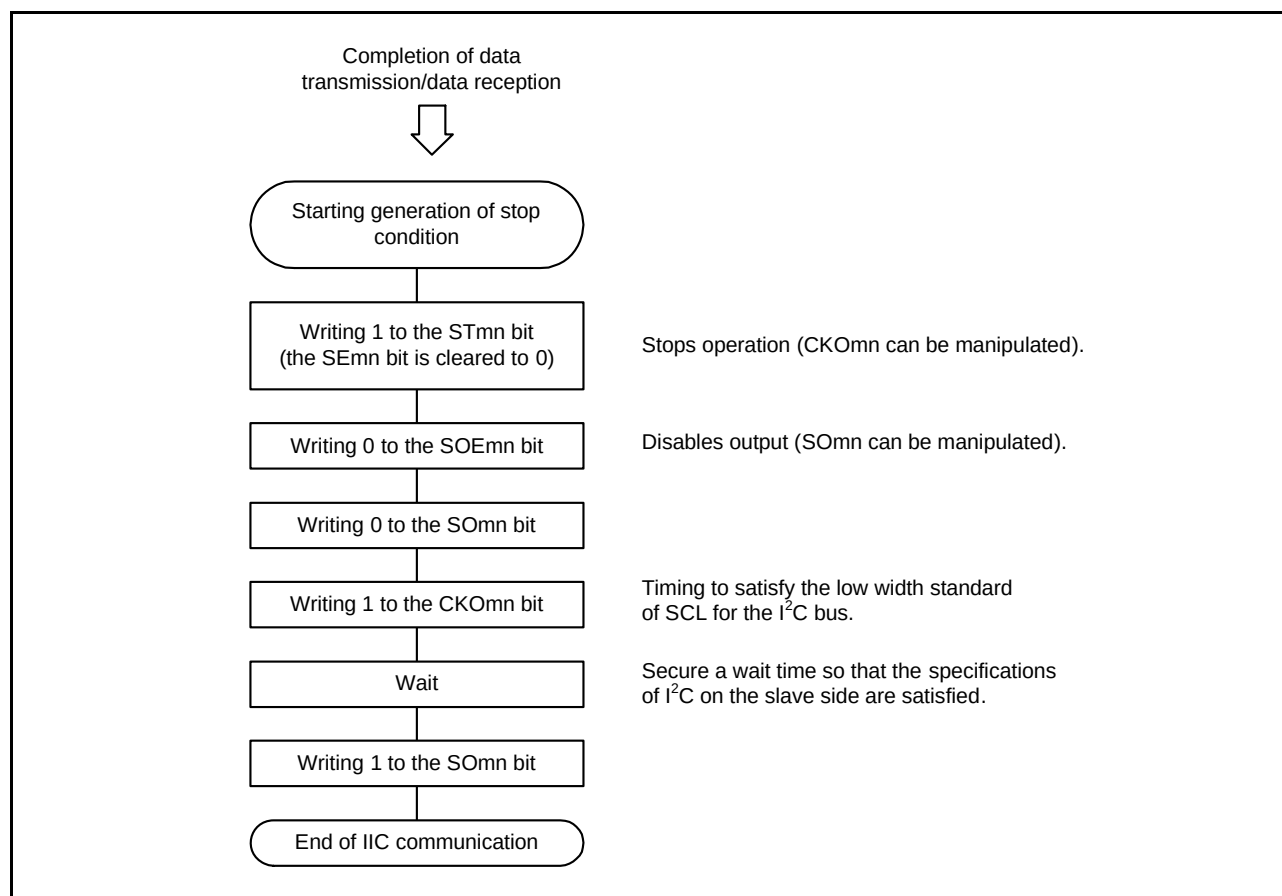
(1) Processing flow

Figure 13 - 113 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 13 - 114 Flowchart of Stop Condition Generation



13.8.5 Calculating Transfer Rate

The transfer rate for simplified I²C (IIC00, IIC01, IIC11, IIC20, IIC21) communication can be calculated by the following expressions.

$$(\text{Transfer rate}) = \{\text{Operating clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

Caution SDRmn[15:9] must not be set to 00000000B. Set SDRmn[15:9] to 0000001B or greater.
The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

The operating clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 13 - 9 Selection of Operating Clock For Simplified I²C

SMRmn Register	SPSm Register								Operating Clock (fMCK) ^{Note}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		fCLK = 32 MHz
0	x	x	x	x	0	0	0	0	fCLK	32 MHz
	x	x	x	x	0	0	0	1	fCLK/2	16 MHz
	x	x	x	x	0	0	1	0	fCLK/2 ²	8 MHz
	x	x	x	x	0	0	1	1	fCLK/2 ³	4 MHz
	x	x	x	x	0	1	0	0	fCLK/2 ⁴	2 MHz
	x	x	x	x	0	1	0	1	fCLK/2 ⁵	1 MHz
	x	x	x	x	0	1	1	0	fCLK/2 ⁶	500 kHz
	x	x	x	x	0	1	1	1	fCLK/2 ⁷	250 kHz
	x	x	x	x	1	0	0	0	fCLK/2 ⁸	125 kHz
	x	x	x	x	1	0	0	1	fCLK/2 ⁹	62.5 kHz
	x	x	x	x	1	0	1	0	fCLK/2 ¹⁰	31.25 kHz
	x	x	x	x	1	0	1	1	fCLK/2 ¹¹	15.63 kHz
1	0	0	0	0	x	x	x	x	fCLK	32 MHz
	0	0	0	1	x	x	x	x	fCLK/2	16 MHz
	0	0	1	0	x	x	x	x	fCLK/2 ²	8 MHz
	0	0	1	1	x	x	x	x	fCLK/2 ³	4 MHz
	0	1	0	0	x	x	x	x	fCLK/2 ⁴	2 MHz
	0	1	0	1	x	x	x	x	fCLK/2 ⁵	1 MHz
	0	1	1	0	x	x	x	x	fCLK/2 ⁶	500 kHz
	0	1	1	1	x	x	x	x	fCLK/2 ⁷	250 kHz
	1	0	0	0	x	x	x	x	fCLK/2 ⁸	125 kHz
	1	0	0	1	x	x	x	x	fCLK/2 ⁹	62.5 kHz
	1	0	1	0	x	x	x	x	fCLK/2 ¹⁰	31.25 kHz
	1	0	1	1	x	x	x	x	fCLK/2 ¹¹	15.63 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. x: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), mn = 00, 01, 03, 10, 11

Here is an example of setting an I²C transfer rate where $f_{MCK} = f_{CLK} = 32 \text{ MHz}$.

I ² C Transfer Mode (Desired Transfer Rate)	fCLK = 32 MHz			
	Operating Clock (fMCK)	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	fCLK/2	79	100 kHz	0.0%
400 kHz	fCLK	41	380 kHz	5.0% Note
1 MHz	fCLK	18	0.84 MHz	16.0% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

13.8.6 Procedure for Processing Errors that Occurred during Simplified I²C (IIC00, IIC01, IIC11, IIC20, and IIC21) Communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC11, IIC20, IIC21) communication is described in **Table 13 - 10** and **Table 13 - 11**.

Table 13 - 10 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 13 - 11 Processing Procedure in Case of ACK Error in Simplified I²C Mode

Software Manipulation	State of the Hardware	Remark
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1.	The SEMn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	The slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates a stop condition.		
Creates a start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1.	The SEMn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), r: IIC number (r = 00, 01, 11, 20, 21),
mn = 00, 01, 03, 10, 11

Section 14 Serial Interface IICA (IICA)

The number of the serial Interface IICA channels depends on the product.

	24-, 25-, 30-, 32-, 36-, 40-, 44-, and 48-pin products
Number of serial interface channels	1

14.1 Functions of Serial Interface IICA

The serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. The operating power can be reduced in this mode.

(2) I²C bus mode (multi-master supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I²C bus format and the master device can send start conditions, addresses, transfer directions, acknowledges (ACK), data, and stop conditions to the slave devices, via the serial data bus. The slave device automatically detects these states and data by hardware. This function can simplify the part of application program that controls the I²C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or the local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1).

The all address match function is enabled by setting the SVADISn bit of the ICCTLn1 register to 1, allowing any received address is to be determined as a matched address.

Figure 14 - 1 shows a block diagram of serial interface IICA.

Remark n = 0

Figure 14 - 1 Block Diagram of Serial Interface IICA0

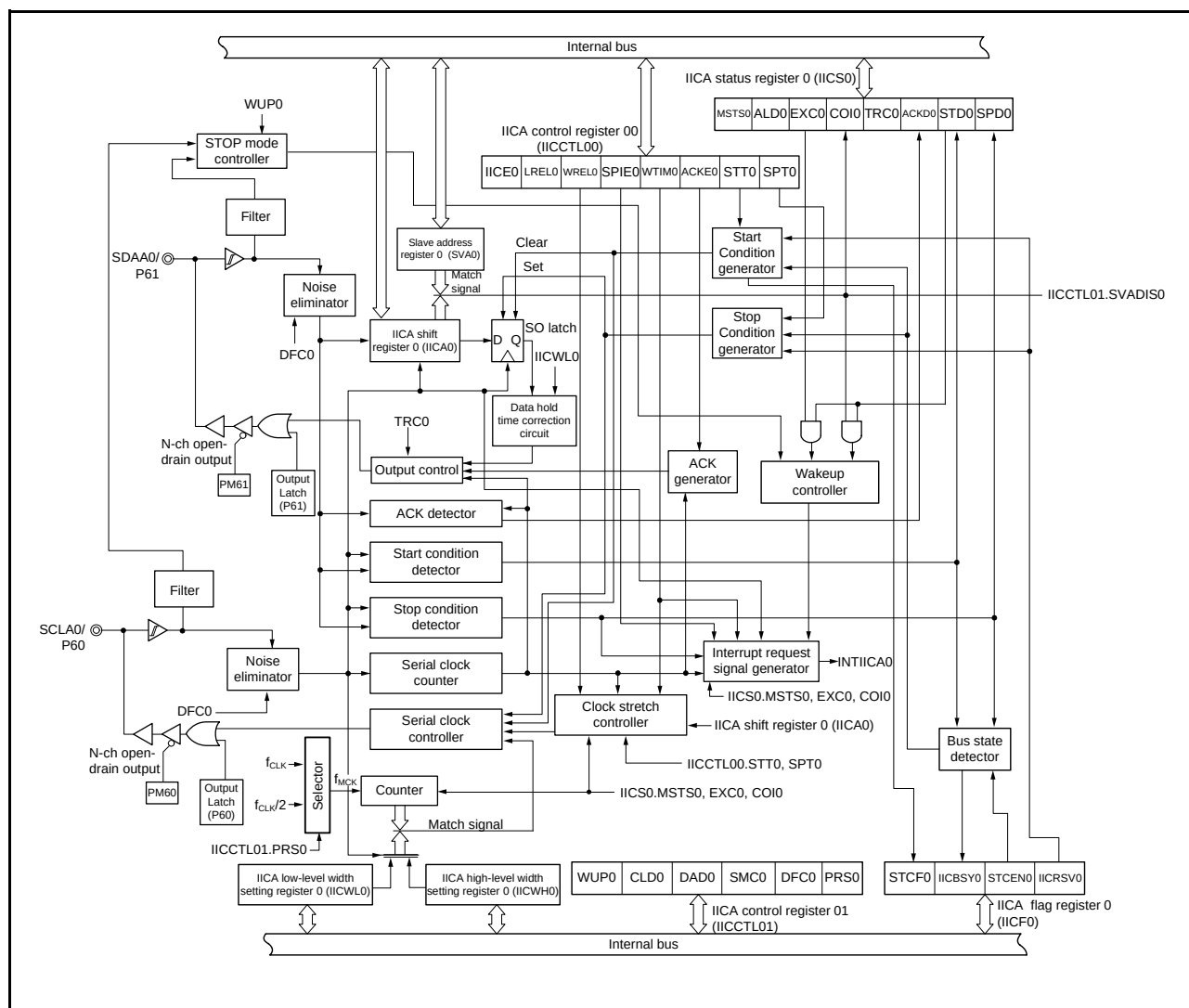
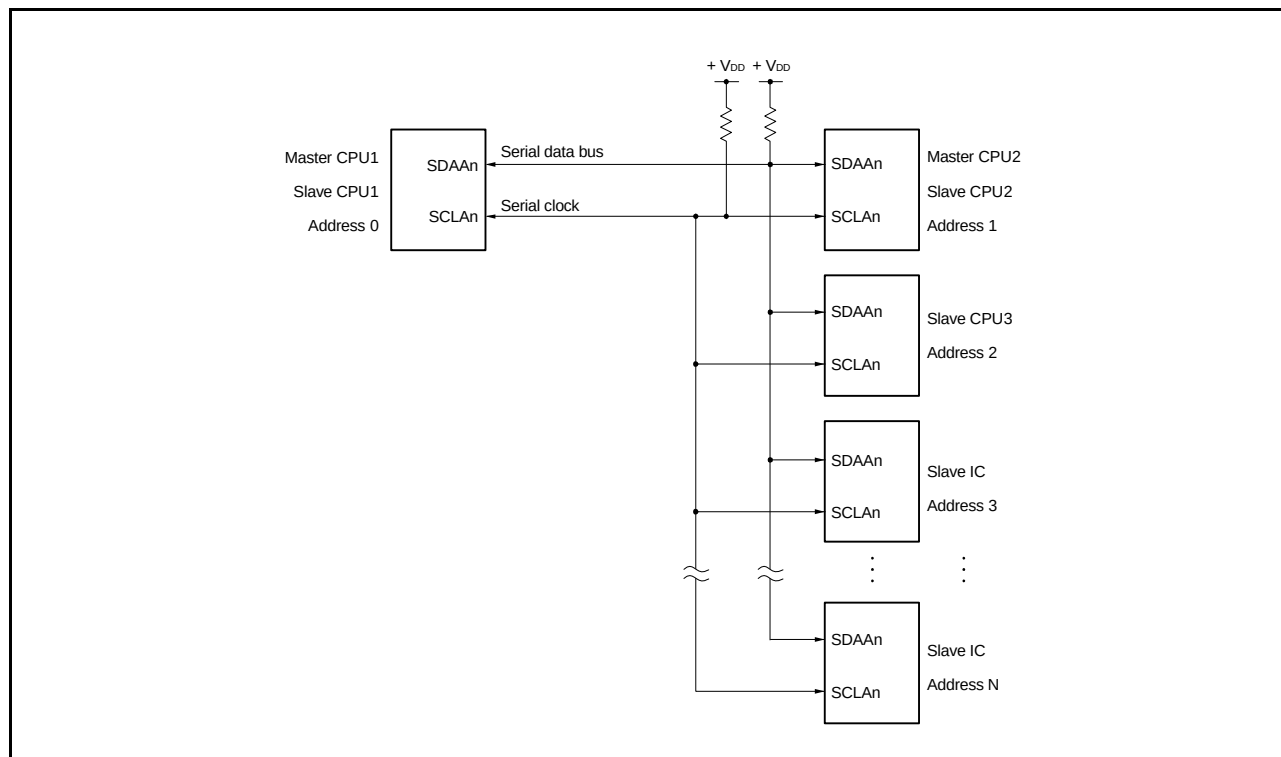


Figure 14 - 2 shows a serial bus configuration example.

Figure 14 - 2 Example of the Serial Bus Configuration Using the I²C Bus



Remark n = 0

14.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 14 - 1 Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register n (IICAn) Slave address register n (SVAn)
Control registers	Peripheral enable register 0 (PER0) Peripheral reset control register 0 (PRR0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (IICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode register 6 (PM6) Port register 6 (P6)

Remark n = 0

(1) IICA shift register n (IICAn)

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing to and reading from the IICAn register.

Release serial interface IICA from the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period.

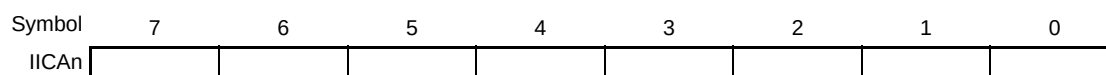
The IICAn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 3 Format of IICA Shift Register n (IICAn)

Address: FFF50H (IICA0)

After reset: 00H

R/W: R/W



Caution 1. Do not write data to the IICAn register during data transfer.

Caution 2. Write to or read from the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (STTn) is set to 1.

Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0

(2) Slave address register n (SVAn)

This register holds seven bits (A6, A5, A4, A3, A2, A1, and A0) of the local address when in slave mode.

The SVAn register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected).

The value of this register following a reset is 00H.

Figure 14 - 4 Format of Slave Address Register n (SVAn)

Address: F0234H (SVA0)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAAn pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIICAn) when the received address matches the address value set to the slave address register n (SVAn), when any address is received while the all address match function is enabled, or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clock cycles that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICAn).

An I²C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

Remark WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)

SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

(7) Serial clock controller

In master mode, this circuit generates the serial clock, which is output via the SCLAn pin.

(8) Clock stretch controller

This circuit controls the timing of clock stretching.

Remark n = 0

- (9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate or detect each state.

- (10) Data hold time correction circuit

This circuit generates the hold time for data after the falling edge of the serial clock.

- (11) Start condition generator

This circuit generates a start condition when the STTn bit is set to 1.

However, while communication reservations are disabled (IICRSVn bit = 1) and the bus is busy (IICBSYn bit = 1), start condition requests are ignored and the STCFn bit is set to 1.

- (12) Stop condition generator

This circuit generates a stop condition when the SPTn bit is set to 1.

- (13) Bus state detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus state cannot be detected immediately after the IICA operation is enabled, the initial state is set by the STCENn bit.

Remark 1. STTn bit: Bit 1 of IICA control register n0 (IICCTLn0)

SPTn bit: Bit 0 of IICA control register n0 (IICCTLn0)

IICRSVn bit: Bit 0 of IICA flag register n (IICFn)

IICBSYn bit: Bit 6 of IICA flag register n (IICFn)

STCFn bit: Bit 7 of IICA flag register n (IICFn)

STCENn bit: Bit 1 of IICA flag register n (IICFn)

Remark 2. n = 0

14.3 Registers for Controlling Serial Interface IICA

The following registers are used to control serial interface IICA.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- IICA control register n0 (IICCTLn0)
- IICA status register n (IICSn)
- IICA flag register n (IICFn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)
- Port mode registers (PMxx)
- Port registers (Pxx)

Remark n = 0

14.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

When serial interface IICAn is to be used, be sure to set bit 4 (IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H

After reset: 00H

R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICAnEN	Control of supply of an input clock to the serial interface IICAn
0	Stops supply of an input clock. - The SFRs used by the serial interface IICAn cannot be written. - When an SFR used by the serial interface IICAn is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the serial interface IICAn can be read and written.

Caution 1. When setting serial interface IICA, make sure that the setting of the IICAnEN bit is 1 before setting the following registers. If IICAnEN = 0, the values of the registers which control the serial interface IICA are cleared to their initial values, and writing to them is ignored (except for port mode register 6 (PM6) and port register 6 (P6)).

- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)

Caution 2. The functions that are mounted depend on the product. For details on the PER0 register, see the description in Section 6 Clock Generator.

Caution 3. While the operation of a peripheral function is enabled, do not switch the setting of the corresponding bit in the PER0 register.

When switching the setting by the PER0 register, each peripheral function assigned to the PER0 register must be stopped.

Remark n = 0

14.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Use bit 4 (IICA0RES) of the PRR0 register to control reset or release from the reset for the serial interface IICA0.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 14 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H

After reset: 00H

R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES

IICAnRES	Control resetting of the serial interface IICAn
0	Serial interface IICAn is released from the reset state.
1	Serial interface IICAn is in the reset state. • The SFRs for use with the serial interface IICAn are initialized.

Caution 1. Be sure to set bits 7, 6, and 1 to 0.

Caution 2. The functions that are mounted depend on the product. For details on the PRR0 register, see the description in Section 21 Reset Function.

Remark n = 0

14.3.3 IICA control register n0 (IICCTLn0)

This register is used to enable or disable the I²C operations, set the timing of clock stretching, and set other I²C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that bits SPIEn, WTIMn, and ACKEn must be set while the setting of IICEn is 0 or this module is in the clock stretch state. These bits can be set at the same time as setting the IICEn bit 1. The value of this register following a reset is 00H.

Remark n = 0

Figure 14 - 7 Format of IICA Control Register n0 (IICCTLn0) (1/5)

Address: F0230H (IICCTL00)

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I ² C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) ^{Note} . Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
- Cleared by instruction - Reset	Set by instruction

Note The IICA status register n (IICSn), the STCFn and IICBSYn bits of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Caution If the operation of I²C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I²C (IICEn = 1).

Remark n = 0

Figure 14 - 7 Format of IICA Control Register n0 (IICCTLn0) (2/5)

LRELn Notes 1, 2	Exit from communications				
0	Normal operation				
1	IICA exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn				
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match, extension code reception, or address reception with the all address match function enabled occurs after the start condition.					
<table border="1"> <thead> <tr> <th>Condition for clearing (LRELn = 0)</th><th>Condition for setting (LRELn = 1)</th></tr> </thead> <tbody> <tr> <td> • Automatically cleared after execution • Reset </td><td> • Set by instruction </td></tr> </tbody> </table>		Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)	• Automatically cleared after execution • Reset	• Set by instruction
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)				
• Automatically cleared after execution • Reset	• Set by instruction				

WRELn Notes 1, 2	Release from the clock stretch state				
0	The interface is not released from the clock stretch state.				
1	The interface is released from the clock stretch state. After release from the clock stretch state, this bit is automatically cleared to 0.				
When the WRELn bit is set (for release from the clock stretch state) during the clock stretch period at the ninth clock pulse in the transmission state (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).					
<table border="1"> <thead> <tr> <th>Condition for clearing (WRELn = 0)</th><th>Condition for setting (WRELn = 1)</th></tr> </thead> <tbody> <tr> <td> • Automatically cleared after execution • Reset </td><td> • Set by instruction </td></tr> </tbody> </table>		Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)	• Automatically cleared after execution • Reset	• Set by instruction
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)				
• Automatically cleared after execution • Reset	• Set by instruction				

SPIEnNote 1	Enable/disable generation of interrupt request when stop condition is detected				
0	Disable				
1	Enable				
If the WUPn bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIEn = 1.					
<table border="1"> <thead> <tr> <th>Condition for clearing (SPIEn = 0)</th><th>Condition for setting (SPIEn = 1)</th></tr> </thead> <tbody> <tr> <td> • Cleared by instruction • Reset </td><td> • Set by instruction </td></tr> </tbody> </table>		Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)	• Cleared by instruction • Reset	• Set by instruction
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)				
• Cleared by instruction • Reset	• Set by instruction				

Note 1. The setting of this bit has no effect while the setting of IICEn is 0.

Note 2. Reading the LRELn and WRELn bits always returns 0.

Caution If the operation of I²C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I²C (IICEn = 1).

Remark n = 0

Figure 14 - 7 Format of IICA Control Register n0 (IICCTLn0) (3/5)

WTIMn Note 1	Control of clock stretching and interrupt request generation
0	An interrupt request is generated on the falling edge of the eighth clock cycle. Master mode: After the output of eight clock pulses, the clock output is set to the low level and clock stretching is set. Slave mode: After the input of eight clock pulses, the clock is set to the low level and clock stretching is set for the master device.
1	An interrupt request is generated on the falling edge of the ninth clock cycle. Master mode: After the output of nine clock pulses, the clock output is set to the low level and clock stretching is set. Slave mode: After the input of nine clock pulses, the clock is set to the low level and clock stretching is set for the master device.
An interrupt is generated on the falling edge of the ninth clock cycle during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. In master mode, clock stretching is inserted at the falling edge of the ninth clock cycle during address transfer. For a slave device that has received a local address, clock stretching is inserted at the falling edge of the ninth clock cycle after an acknowledge (ACK) is issued. However, if the slave device has received an extension code, clock stretching is inserted at the falling edge of the eighth clock cycle. When an address is received while the all address match function is enabled, clock stretching is inserted at the falling edge of the eighth clock cycle.	
Condition for clearing (WTIMn = 0)	
Condition for setting (WTIMn = 1)	
- Cleared by instruction - Reset	
Set by instruction	

ACKEn Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.
Condition for clearing (ACKEn = 0)	
Condition for setting (ACKEn = 1)	
- Cleared by instruction - Reset	
Set by instruction	

Note 1. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 2. The set value is invalid during address transfer and if the code is not an extension code, and the all address match function is disabled.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Remark n = 0

Figure 14 - 7 Format of IICA Control Register n0 (IICCTLn0) (4/5)

STTn Notes 1, 2	Start condition trigger				
0	Do not generate a start condition.				
1	When bus is released (in standby state, when IICBSYn = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: - When communication reservation function is enabled (IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. - When communication reservation function is disabled (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated. In the clock stretch state (for a master device): Generates a restart condition after release from the clock stretch state.				
Cautions concerning set timing - For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception. - For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock. - Cannot be set to 1 at the same time as stop condition trigger (SPTn). - Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.					
<table border="1"> <thead> <tr> <th>Condition for clearing (STTn = 0)</th><th>Condition for setting (STTn = 1)</th></tr> </thead> <tbody> <tr> <td> - Cleared by setting the STTn bit to 1 while communication reservation is prohibited. - Cleared by loss in arbitration - Cleared after start condition is generated by master device - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset </td><td> - Set by instruction </td></tr> </tbody> </table>		Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)	- Cleared by setting the STTn bit to 1 while communication reservation is prohibited. - Cleared by loss in arbitration - Cleared after start condition is generated by master device - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset	Set by instruction
Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)				
- Cleared by setting the STTn bit to 1 while communication reservation is prohibited. - Cleared by loss in arbitration - Cleared after start condition is generated by master device - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset	Set by instruction				

Note 1. The signal of this bit is invalid while IICEn is 0.

Note 2. The STTn bit is always read as 0.

Remark 1. IICRSVn: Bit 0 of IIC flag register n (IICFn)
STCFn: Bit 7 of IIC flag register n (IICFn)

Remark 2. n = 0

Figure 14 - 7 Format of IICA Control Register n0 (IICCTLn0) (5/5)

SPTn ^{Note}	Stop condition trigger				
0	Stop condition is not generated.				
1	Stop condition is generated (termination of master device's transfer).				
Cautions concerning set timing - For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception. - For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the clock stretch period that follows output of the ninth clock. - Cannot be set to 1 at the same time as start condition trigger (STTn). - The SPTn bit can be set to 1 only when in master mode. - When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the clock stretch period that follows output of eight clock pulses, note that a stop condition will be generated during the high-level period of the ninth clock after release from the clock stretch state. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clock pulses, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock. - Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.					
<table border="1"> <thead> <tr> <th>Condition for clearing (SPTn = 0)</th><th>Condition for setting (SPTn = 1)</th></tr> </thead> <tbody> <tr> <td> - Cleared by loss in arbitration - Automatically cleared after stop condition is detected - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset </td><td> - Set by instruction </td></tr> </tbody> </table>		Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)	- Cleared by loss in arbitration - Automatically cleared after stop condition is detected - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset	Set by instruction
Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)				
- Cleared by loss in arbitration - Automatically cleared after stop condition is detected - Cleared by LRELn = 1 (exit from communications) - When IICEn = 0 (operation stop) - Reset	Set by instruction				

Note The SPTn bit is always read as 0.

Caution When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission state), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and the interface is released from the clock stretch state, after which the TRCn bit is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn bit is 1 (transmission state) by writing to the IICA shift register n.

Remark 1. Bit 0 (SPTn) becomes 0 when it is read after data setting.

Remark 2. n = 0

14.3.4 IICA status register n (IICS_n)

This register indicates the state of the I²C.

The IICS_n register can only be read by a 1-bit or 8-bit memory manipulation instruction while the setting of STT_n is 1 or this module is in the clock stretch state. The value of this register following a reset is 00H.

Caution Reading the IICS_n register while the address match wakeup function is enabled (WUP_n = 1) in STOP mode is prohibited. When the WUP_n bit is changed from 1 to 0 (stopping wakeup operation), regardless of the INTIICAn interrupt request signal, a change in the state is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICS_n register after the interrupt has been detected.

Remark STT_n: Bit 1 of IICA control register n0 (IICCTLn0)

WUP_n: Bit 7 of IICA control register n1 (IICCTLn1)

Figure 14 - 8 Format of IICA Status Register n (IICS_n) (1/4)

Address: FFF51H (IICS0)

After reset: 00H

R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n
MSTS _n	Master status check flag							
0	Slave state or communications standby state							
1	Master communications state							
Condition for clearing (MSTS _n = 0)					Condition for setting (MSTS _n = 1)			
• When a stop condition is detected • When ALD_n = 1 (arbitration loss) • Cleared by LREL_n = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset					• When a start condition is generated			
ALD _n	Detection of arbitration loss							
0	This state means either that no arbitration is in progress or that the arbitration result was a “win”.							
1	This state indicates the arbitration result was a “loss”. The MSTS _n bit is cleared.							
Condition for clearing (ALD _n = 0)					Condition for setting (ALD _n = 1)			
• Automatically cleared after the IICSn register is read^{Note} • When the IICEn bit changes from 1 to 0 (operation stop) • Reset					• When the arbitration result is a “loss”.			

Note The ALD_n bit is also cleared when a 1-bit memory manipulation instruction is executed for another bit in the IICS_n register. Therefore, when using the ALD_n bit, read the data of this bit before the data of the other bits.

Remark 1. LREL_n: Bit 6 of IICA control register n0 (IICCTLn0)

IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 14 - 8 Format of IICA Status Register n (IICSn) (2/4)

EXCn	Detection of extension code reception	
0	Extension code was not received.	
1	Extension code was received. Or, the all address match function is enabled.	
Condition for clearing (EXCn = 0)		Condition for setting (EXCn = 1)
- When a start condition is detected - When a stop condition is detected - Cleared by LRELn = 1 (exit from communications) - When the IICEn bit changes from 1 to 0 (operation stop) - Reset		- When the higher four bits of the received address data is either "0000" or "1111" (set at the rising edge of the eighth clock). - When an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses	
0	Addresses do not match.	
1	Addresses match. Or, the all address match function is enabled.	
Condition for clearing (COIn = 0)		Condition for setting (COIn = 1)
- When a start condition is detected - When a stop condition is detected - Cleared by LRELn = 1 (exit from communications) - When the IICEn bit changes from 1 to 0 (operation stop) - Reset		- When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock). - When an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)

IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 14 - 8 Format of IICA Status Register n (IICSn) (3/4)

TRCn	Detection of transmission/reception state
0	Reception state (non-transmission state). The SDAAn line is set for high impedance.
1	Transmission state. The value in the SON latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte's ninth clock).
Condition for clearing (TRCn = 0)	
<Both master and slave> • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Cleared by WRELn = 1^{Note} (release from the clock stretch state) • When the ALDn bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS_n, EXC_n, COIn = 0) <Master> • When 1 is output to the first byte's LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When 0 is input to the first byte's LSB (transfer direction specification bit)	
Condition for setting (TRCn = 1)	
<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)	

Note When bit 3 (TRCn) of the IICA status register n (IICS_n) is set to 1 (transmission state), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and the interface is released from the clock stretch state, after which the TRCn bit is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn bit is 1 (transmission state) by writing to the IICA shift register n.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 14 - 8 Format of IICA Status Register n (IICSn) (4/4)

ACKDn	Detection of acknowledge (ACK)	
0	Acknowledge was not detected.	
1	Acknowledge was detected.	
Condition for clearing (ACKDn = 0)		Condition for setting (ACKDn = 1)
- When a stop condition is detected - At the rising edge of the next byte's first clock - Cleared by LRELn = 1 (exit from communications) - When the IICEn bit changes from 1 to 0 (operation stop) - Reset		After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock

STDn	Detection of start condition	
0	Start condition was not detected.	
1	Start condition was detected. This indicates that the address transfer period is in effect.	
Condition for clearing (STDn = 0)		Condition for setting (STDn = 1)
- When a stop condition is detected - At the rising edge of the next byte's first clock following address transfer - Cleared by LRELn = 1 (exit from communications) - When the IICEn bit changes from 1 to 0 (operation stop) - Reset		When a start condition is detected

SPDn	Detection of stop condition	
0	Stop condition was not detected.	
1	Stop condition was detected. The master device's communication is terminated and the bus is released.	
Condition for clearing (SPDn = 0)		Condition for setting (SPDn = 1)
- At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition - When the WUPn bit changes from 1 to 0 - When the IICEn bit changes from 1 to 0 (operation stop) - Reset		When a stop condition is detected

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)

IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

14.3.5 IICA flag register n (IICFn)

This register sets the operation mode of I²C and indicates the state of the I²C bus.

The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I²C bus status flag (IICBSYn) bits are read-only.

The IICRSVn bit can be used to enable or disable the communication reservation.

The STCENn bit can be used to set the initial value of the IICBSYn bit.

The IICRSVn and STCENn bits can be written only when the operation of I²C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). The IICFn register is read-only while the operation of the I²C is enabled. The value of this register following a reset is 00H.

Figure 14 - 9 Format of IICA Flag Register n (IICFn) (1/2)

Address: FFF52H (IICF0)

After reset: 00H

R/W: R/WNote

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag
Condition for clearing (STCFn = 0)	
• Cleared by STTn = 1 • When IICEn = 0 (operation stop) • Reset	
Condition for setting (STCFn = 1)	
• Generating start condition unsuccessful and the STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).	

IICBSYn	I ² C bus status flag
0	Bus released state (initial communications state when STCENn = 1)
1	Bus communications state (initial communications state when STCENn = 0)
Condition for clearing (IICBSYn = 0)	
• When a stop condition is detected • When IICEn = 0 (operation stop) • Reset	
Condition for setting (IICBSYn = 1)	
• When a start condition is detected • Setting of the IICEn bit when STCENn = 0	

Note Bits 7 and 6 are read-only.

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Remark 2. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 3. n = 0

Figure 14 - 9 Format of IICA Flag Register n (IICFn) (2/2)

STCENn	Initial start enable trigger	
0	After operation is enabled (IICEn = 1), enable generation of a start condition upon detection of a stop condition.	
1	After operation is enabled (IICEn = 1), enable generation of a start condition without detecting a stop condition.	
Condition for clearing (STCENn = 0)		Condition for setting (STCENn = 1)
• Cleared by instruction • When a start condition is detected • Reset		• Set by instruction

IICRSVn	Communication reservation function disable bit	
0	Enable communication reservation	
1	Disable communication reservation	
Condition for clearing (IICRSVn = 0)		Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset		• Set by instruction

Caution 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

Caution 2. As the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state when STCENn = 1. When generating the first start condition (STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to the IICRSVn bit only when the operation is stopped (IICEn = 0).

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Remark 2. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 3. n = 0

14.3.6 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I²C and detect the states of the SCLAn and SDAAn pins.

The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only.

Set the IICCTLn1 register, except the WUPn bit, while operation of I²C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0). The value of this register following a reset is 00H.

Figure 14 - 10 Format of IICA Control Register n1 (IICCTLn1) (1/3)

Address: F0231H (IICCTL01)

After reset: 00H

R/W: R/W>Note 1

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	SVADISn	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three cycles of f_{MCK} after setting (1) the WUPn bit. See **Figure 14 - 22 Flow When Setting WUPn = 1**.

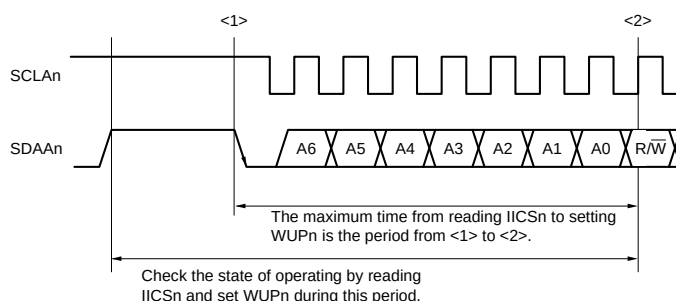
Clear (0) the WUPn bit after the address has matched, an address has been received while the all address match function is enabled, or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. The interface must be released from the clock stretch state and transmit data must be written after the WUPn bit has been cleared (0).

The interrupt timing when the address has matched, when an address has been received while the all address match function is enabled, or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. A delay of the difference of sampling by the clock will occur. Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIEn bit is set to 1.

Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
Cleared by instruction (after address match, address reception with the all address match function enabled, or extension code reception)	Set by instruction (when the MSTSn, EXCn, and COIn bits are 0, and the STDn bit also 0 (communication not entered))Note 2

Note 1. Bits 5 and 4 are read-only.

Note 2. The state of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



Remark n = 0

Figure 14 - 10 Format of IICA Control Register n1 (IICCTLn1) (2/3)

SVADISn	Address match disabling flag	
0	Disables the all address match function.	
1	Enables the all address match function.	
When SVADISn = 1, IICA considers any address as address match, and performs the same operation as that when an extension code is received.		
Therefore, IICSn.COIn is set to 1, and IICSn.EXCn is set to 1.		
For details about extension code reception, see 14.5.11 Extension code .		

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)	
0	The SCLAn pin was detected at low level.	
1	The SCLAn pin was detected at high level.	
Condition for clearing (CLDn = 0)		Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICEn = 0 (operation stop) • Reset		• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)	
0	The SDAAn pin was detected at low level.	
1	The SDAAn pin was detected at high level.	
Condition for clearing (DADn = 0)		Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICEn = 0 (operation stop) • Reset		• When the SDAAn pin is at high level

SMCn	Operation mode switching	
0	Operates in standard mode (fastest transfer rate: 100 kbps).	
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).	

Caution 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.).

Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to 1 only when the f_{CLK} exceeds 20 MHz.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock.

The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (min.)

Fast mode plus: f_{CLK} = 10 MHz (min.)

Normal mode: f_{CLK} = 1 MHz (min.)

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

Figure 14 - 10 Format of IICA Control Register n1 (IICCTLn1) (3/3)

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set (1) or cleared (0).	
PRSn	IICA operation clock (fmck)
0	Selects fCLK ($1 \text{ MHz} \leq f_{\text{CLK}} \leq 20 \text{ MHz}$).
1	Selects fCLK/2 ($20 \text{ MHz} < f_{\text{CLK}}$).

Caution 1. The fastest operation frequency of the IICA operation clock (fmck) is 20 MHz (max.).

Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to 1 only when the fCLK exceeds 20 MHz.

Caution 2. Note the minimum fCLK operation frequency when setting the transfer clock.

The minimum fCLK operation frequency for serial interface IICA is determined according to the mode.

Fast mode: fCLK = 3.5 MHz (min.)

Fast mode plus: fCLK = 10 MHz (min.)

Normal mode: fCLK = 1 MHz (min.)

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

14.3.7 IICA low-level width setting register n (IICWLn)

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

The IICWLn register can be set by an 8-bit memory manipulation instruction.

Set the IICWLn register while operation of I²C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

The value of this register following a reset is FFH.

For details about setting the IICWLn register, see **14.4.2 Setting transfer clock by using IICWLn and IICWHn registers**. The data hold time is one-quarter of the time set by the IICWLn register.

Figure 14 - 11 Format of IICA Low-Level Width Setting Register n (IICWLn)

Address: F0232H (IICWL0)

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

14.3.8 IICA high-level width setting register n (IICWHn)

This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

The IICWHn register can be set by an 8-bit memory manipulation instruction.

Set the IICWHn register while operation of I²C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

The value of this register following a reset is FFH.

Figure 14 - 12 Format of IICA High-Level Width Setting Register n (IICWHn)

Address: F0233H (IICWH0)

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1. For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **14.4.2 (1)** and **14.4.2 (2)**, respectively.

Remark 2. n = 0

14.3.9 Registers for controlling the port function multiplexed with the IICA I/O pins

When the IICA serial interface is to be used, set up the port mode register (PMxx) and port register (Pxx) related to the port function multiplexed with the pins of the given IICA channel.

For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICEn bit is 0.

14.4 I²C Bus Mode Functions

14.4.1 Pin configuration

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

- (1) SCLAn....This pin is used for serial clock input and output.

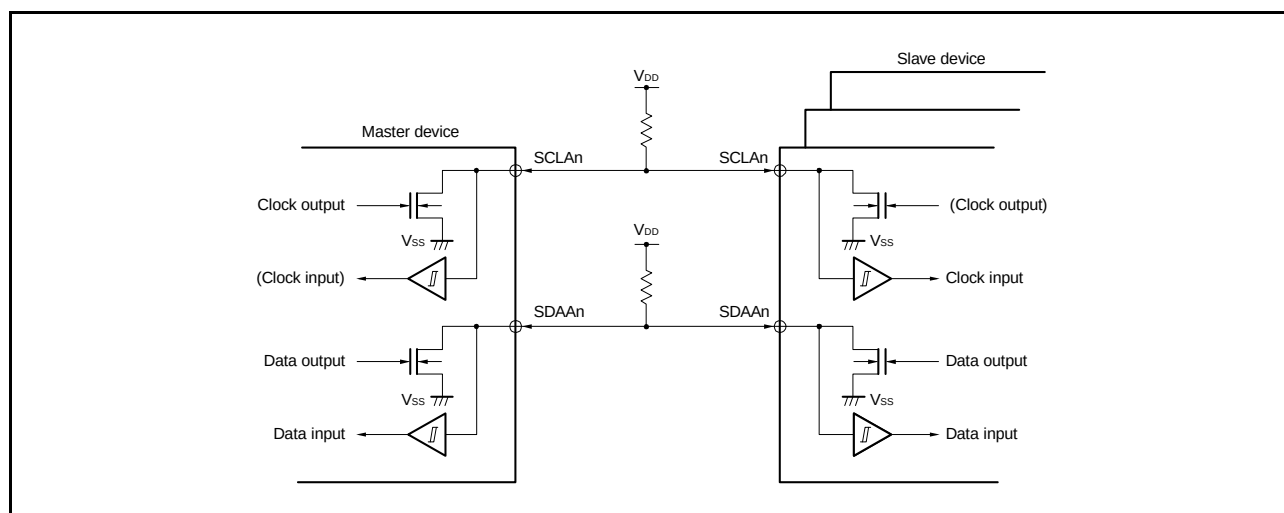
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

- (2) SDAAn....This pin is used for serial data input and output.

This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 14 - 13 Pin Configuration Diagram



Remark n = 0

14.4.2 Setting transfer clock by using IICWL_n and IICWH_n registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{\text{MCK}}}{\text{IICWL} + \text{IICWH} + f_{\text{MCK}} (t_{\text{R}} + t_{\text{F}})}$$

At this time, the optimal setting values of the IICWL_n and IICWH_n registers are as follows.

(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWL}_n = \frac{0.52}{\text{Transfer clock}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.48}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- When the normal mode

$$\text{IICWL}_n = \frac{0.47}{\text{Transfer clock}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.53}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

- When the fast mode plus

$$\text{IICWL}_n = \frac{0.50}{\text{Transfer clock}} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = \left(\frac{0.50}{\text{Transfer clock}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{MCK}}$$

(2) Setting IICWL_n and IICWH_n registers on slave side

(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWL}_n = 1.3 \mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (1.2 \mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- When the normal mode

$$\text{IICWL}_n = 4.7 \mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (5.3 \mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

- When the fast mode plus

$$\text{IICWL}_n = 0.50 \mu\text{s} \times f_{\text{MCK}}$$

$$\text{IICWH}_n = (0.50 \mu\text{s} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{MCK}}$$

Caution 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Set bit 0 (PRSn) of the IICA control register n1 (IICCTL_n1) to 1 only when the f_{CLK} exceeds 20 MHz.

Caution 2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: $f_{\text{CLK}} = 3.5 \text{ MHz (min.)}$

Fast mode plus: $f_{\text{CLK}} = 10 \text{ MHz (min.)}$

Normal mode: $f_{\text{CLK}} = 1 \text{ MHz (min.)}$

(Remarks are listed on the next page.)

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistor and wire load.

Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
 t_F : SDAAn and SCLAn signal falling times
 t_R : SDAAn and SCLAn signal rising times
 f_{MCK} : IICA operation clock frequency

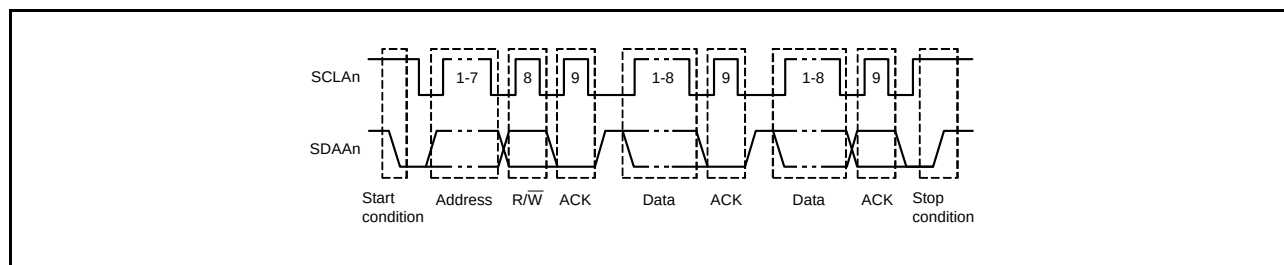
Remark 3. $n = 0$

14.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus.

Figure 14 - 14 shows the transfer timing for the “start condition”, “address”, “data”, and “stop condition” output via the I²C bus's serial data bus.

Figure 14 - 14 I²C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

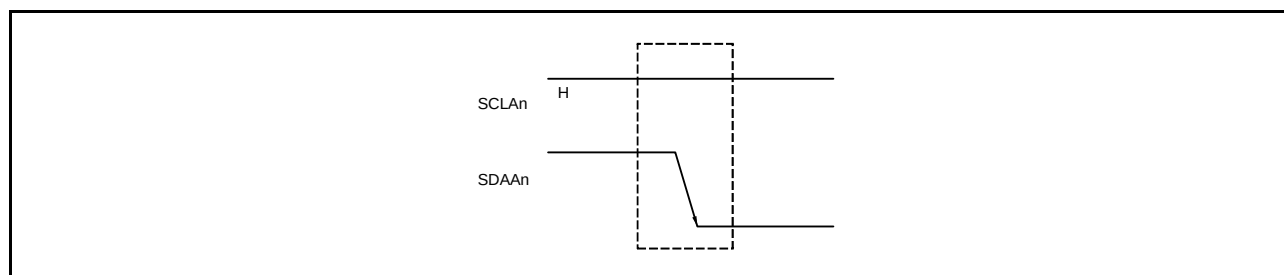
The serial clock (SCLAn) is continuously output by the master device. However, for the slave device, the period over which the SCLAn pin is at the low level can be extended and clock stretching can be inserted.

14.5.1 Start conditions

When the SCLAn pin is at high level, changing the SDAAn pin from the high level to the low level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 14 - 15 Start Conditions



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

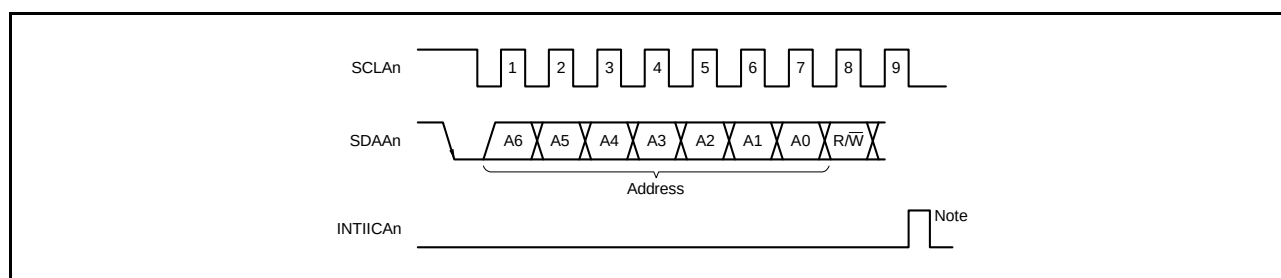
Remark n = 0

14.5.2 Address

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address. The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 14 - 16 Addresses



Note INTIICAn is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **14.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

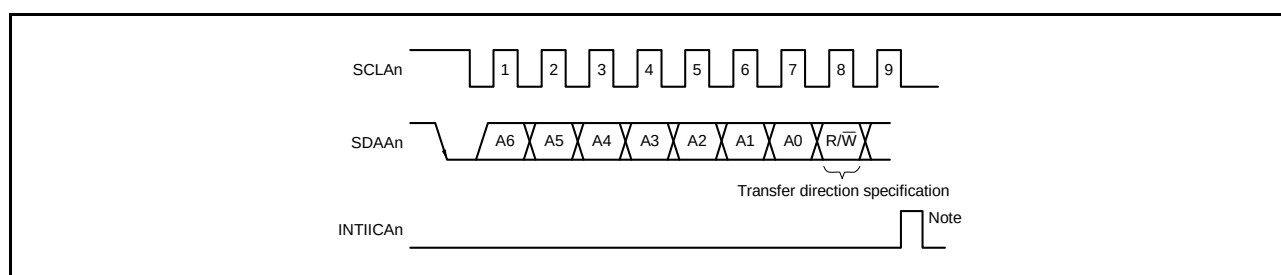
The slave address is assigned to the higher 7 bits of the IICAn register.

14.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 14 - 17 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Remark n = 0

14.5.4 Acknowledge (ACK)

ACK is used to check the state of serial data at the transmission and reception sides.

The reception side returns ACK each time it has received 8-bit data.

The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKDn) of the IICA status register n (IICSn).

When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

- <1> Reception was not performed normally.
- <2> The final data item was received.
- <3> The reception side specified by the address does not exist.

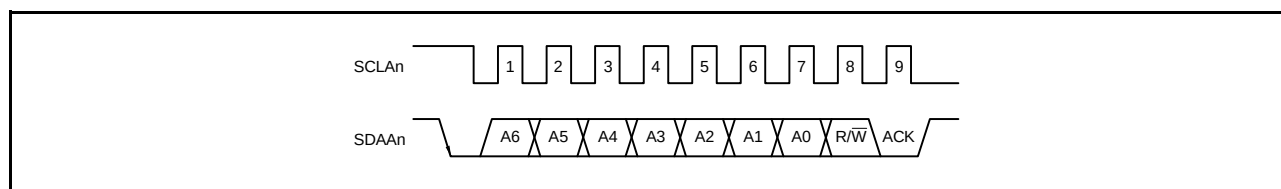
To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set to the value of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0).

If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 14 - 18 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, or when an address is received while the all address match function is enabled, ACK is generated if the ACKEn bit is set to 1 in advance.

How ACK is generated when data is received depends on the setting of the timing of clock stretching as follows.

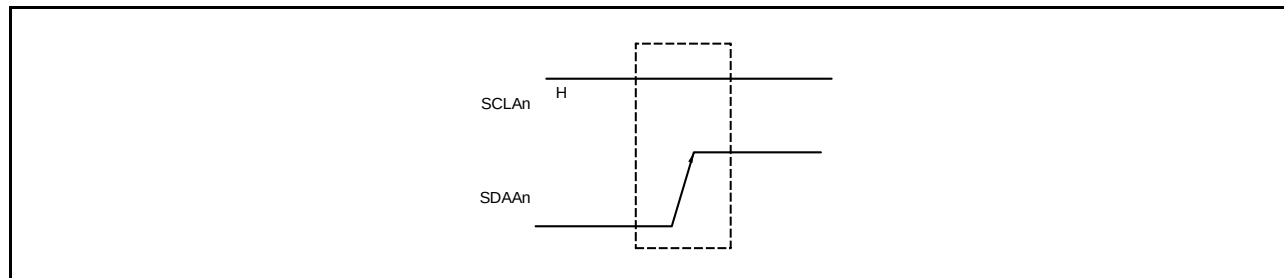
- When 8th cycle clock stretching is selected (bit 3 (WTIMn) of the IICCTLn0 register = 0):
By setting the ACKEn bit to 1 before release from the clock stretch state, ACK is generated at the falling edge of the eighth clock cycle of the SCLAn pin.
- When 9th cycle clock stretching is selected (bit 3 (WTIMn) of the IICCTLn0 register = 1):
ACK is generated if the ACKEn bit is set to 1 in advance.

Remark n = 0

14.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition. A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 14 - 19 Stop Condition



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

Remark n = 0

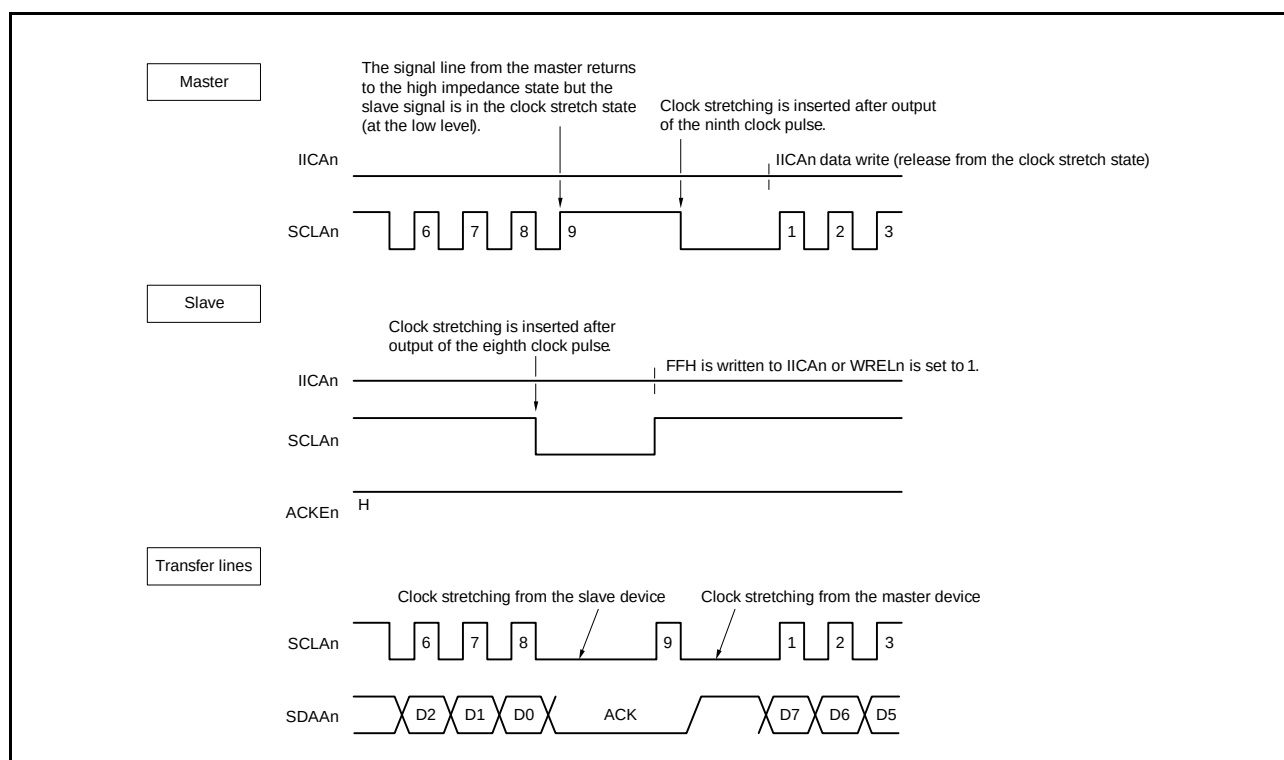
14.5.6 Clock stretching

Clock stretching is used to notify the other party in communications that a device (master or slave) is preparing to transmit or receive data (i.e., the interface is in the clock stretch state).

Setting the SCLAn pin to the low level indicates the clock stretch state to the other party. When clock stretching is released for both the master and slave devices, the next data transfer can start.

Figure 14 - 20 Clock Stretching (1/2)

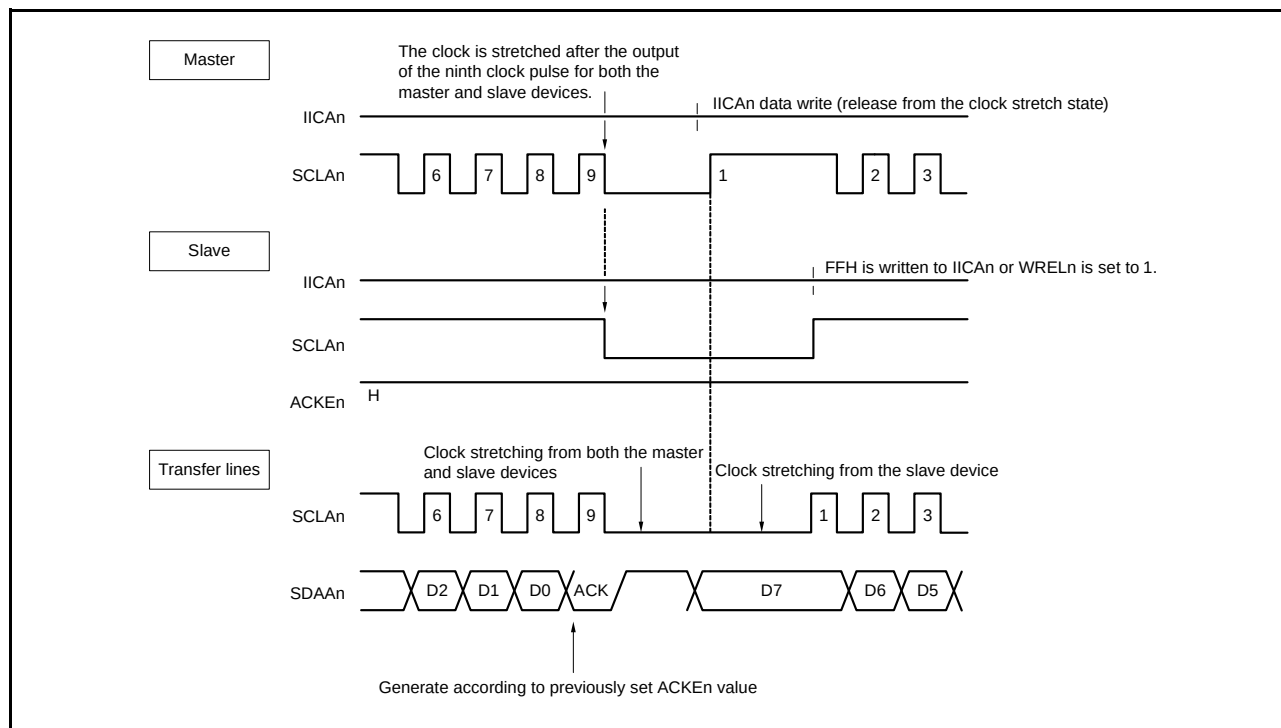
- (1) When clock stretching is set for the ninth and eighth clock cycles for the master and slave devices, respectively (master: transmission, slave: reception, and ACKEn = 1)



Remark n = 0

Figure 14 - 20 Clock Stretching (2/2)

- (2) When clock stretching is set for the ninth clock cycle for both the master and slave devices
(master: transmission, slave: reception, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

Clock stretching is automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0).

Normally, the receiving side releases the clock stretch state when bit 5 (WRELn) of the IICCTLn0 register is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side releases the clock stretch state when data is written to the IICAn register.

The master device can also release the clock stretch state via either of the following methods.

- By setting bit 1 (STTn) of the IICCTLn0 register to 1
- By setting bit 0 (SPTn) of the IICCTLn0 register to 1

Remark n = 0

14.5.7 Release from clock stretching

The I²C interface usually releases the clock stretch state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)^{Note}
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)^{Note}

Note Master only

Executing the above processing for release from clock stretching leads to IICA releasing the clock stretch state after which communications are resumed.

To release the clock stretch state and transmit data (including addresses), write the data to the IICAn register.

To receive data after release from the clock stretch state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.

To generate a restart condition after release from the clock stretch state, set bit 1 (STTn) of the IICCTLn0 register to 1.

To generate a stop condition after release from the clock stretch state, set bit 0 (SPTn) of the IICCTLn0 register to 1.

Execute the processing for release only once for each period in the clock stretch state.

If, for example, data is written to the IICAn register after release from the clock stretch state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.

In addition to the above, communications are stopped if the IICEn bit is cleared to 0 when communications have been aborted, so that the clock stretch state can be released.

If the I²C bus has deadlocked due to noise, the device can exit from communications by setting bit 6 (LRELn) of the IICCTLn0 register to 1, so that the clock stretch state can be released.

Caution If the processing for release from clock stretching is executed when WUPn = 1, the clock stretch state will not be released.

Remark n = 0

14.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and controls clock stretching, as shown in **Table 14 - 2**.

Table 14 - 2 INTIICAn Generation Timing and Control of Clock Stretching

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	gNotes 1, 2	gNote 2	gNote 2	9	8	8
1	gNotes 1, 2	gNote 2	gNote 2	9	9	9

Note 1. The slave device's INTIICAn signal and clock stretching occur at the falling edge of the ninth clock cycle only when there is a match with the address set to the slave address register n (SVAn).
At this point, ACK is generated regardless of the value set to the IICCTLn0 register's bit 2 (ACKEn). For a slave device that has received an extension code, or has received an address while the all address match function is enabled, INTIICAn occurs at the falling edge of the eighth clock.

However, if the address does not match after restart, INTIICAn is generated at the falling edge of the ninth clock cycle, but clock stretching does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn), the all address match function is disabled, and extension code is not received, neither INTIICAn nor clock stretching occurs.

Remark The numbers in the table indicate the pulses of the serial clock signal. Interrupt requests and control of clock stretching are both synchronized with the falling edge of these clock pulses.

(1) During address transmission/reception

- Slave device operation:

The timing of the interrupt and clock stretching depends on the conditions described in **Note 1** and **Note 2** above, regardless of the setting of the WTIMn bit.

- Master device operation:

The interrupt and clock stretching occur at the falling edge of the ninth clock cycle, regardless of the setting of the WTIMn bit.

(2) During data reception

- Master/slave device operation:

The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.

(3) During data transmission

- Master/slave device operation:

The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.

Remark n = 0

(4) Release from clock stretching

The four types of processing for release from clock stretching are as follows.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)^{Note}
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)^{Note}

Note Master only

When 8th cycle clock stretching has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined before release from the clock stretch state.

(5) Detection of stop condition

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

14.5.9 Address match detection method

In I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request signal (INTIICAn) occurs only when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, when an address is received while the all address match function is enabled (IICCTLn1.SVADISn = 1), or when an extension code has been received.

14.5.10 Error detection

In I²C bus mode, the state of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark n = 0

14.5.11 Extension code

- (1) When the higher 4 bits of the receive address are either "0000" or "1111", the extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request signal (INTIICAn) is issued at the falling edge of the eighth clock.

When an address is received while the all address match function is enabled, it is also determined that an extension code has been received.

The local address stored in the slave address register n (SVAn) is not affected.

- (2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer while the SVAn register is set to 11110xx0 or if an address is received while the all address match function is enabled. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match or the all address match function is enabled: EXCn = 1
- Seven bits of data match or the all address match function is enabled: COIn = 1

Remark EXCn: Bit 5 of IICA status register n (IICSn)

COIn: Bit 4 of IICA status register n (IICSn)

- (3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.

If the extension code is received or an address is received with the all address match function enabled during operation as a slave, then the slave device is participating in communication even if its address does not match.

For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 14 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark 1. See the I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2. n = 0

14.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

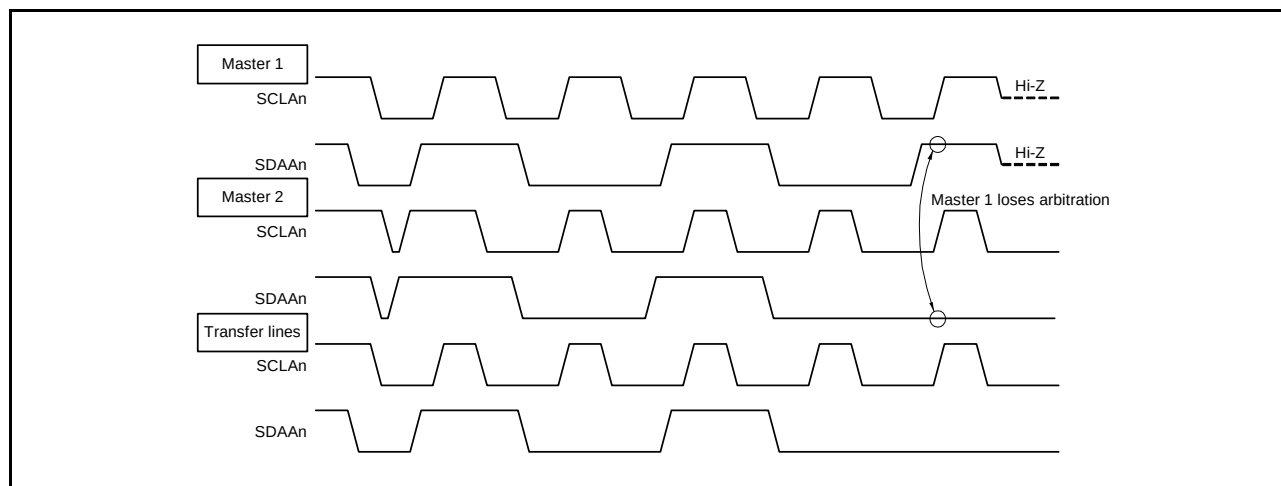
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected by checking ALDn = 1 by software at the timing of the next interrupt request (the eighth or ninth clock cycle, when a stop condition is detected, etc.).

For details of interrupt request timing, see **14.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching**.

Remark STDn: Bit 1 of IICA status register n (IICSn)
STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Figure 14 - 21 Arbitration Timing Example



Remark n = 0

Table 14 - 4 State during Arbitration and Interrupt Request Generation Timing

State during Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIEn = 1) ^{Note 2}
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1) ^{Note 2}
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When the WTIMn bit (bit 3 of IICA control register n0 (IICCTLn0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0, the extension code's slave address is received, and an address is received while the all address match function is enabled, an interrupt request occurs at the falling edge of the eighth clock.

Note 2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0

14.5.13 Wakeup function

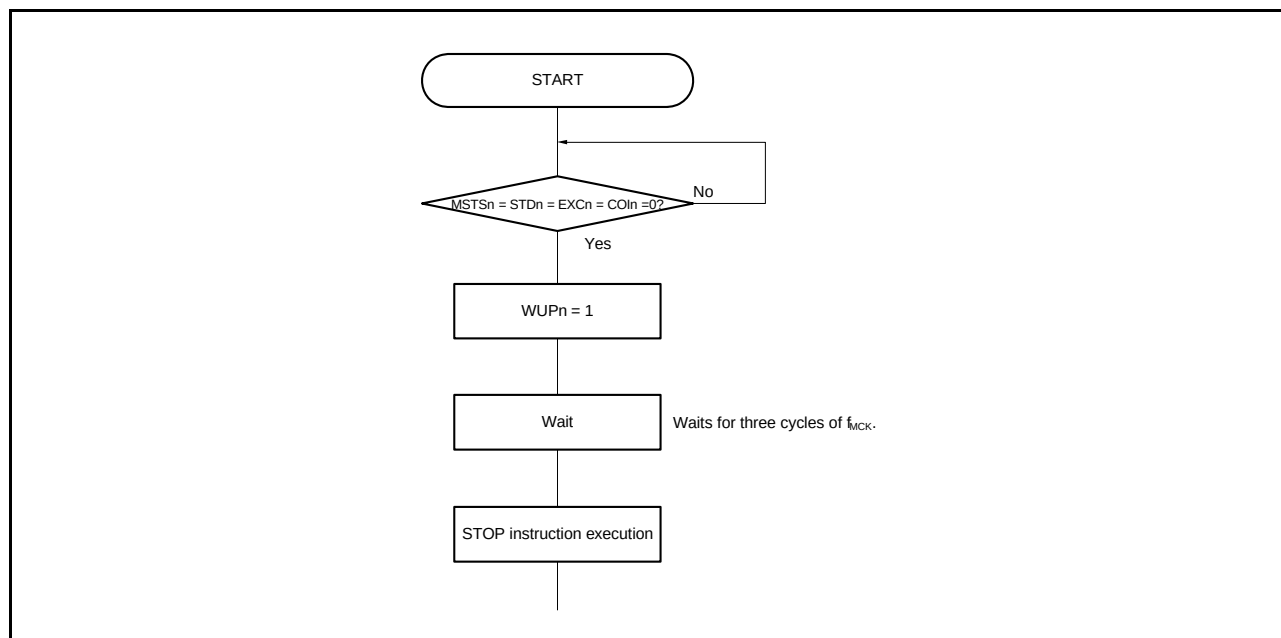
The I²C bus slave function is a function that generates an interrupt request signal (INTIICAn) when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match while the all address match function is disabled.

When a start condition is detected, wakeup standby mode is set. Even a master that has generated a start condition enters the wakeup standby state while transmitting an address because the master may become a slave due to an arbitration loss.

To use the wakeup function in the STOP mode, set the WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) is also generated when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. Operation returns to normal operation by using an instruction to clear (0) the WUPn bit after this interrupt has been generated.

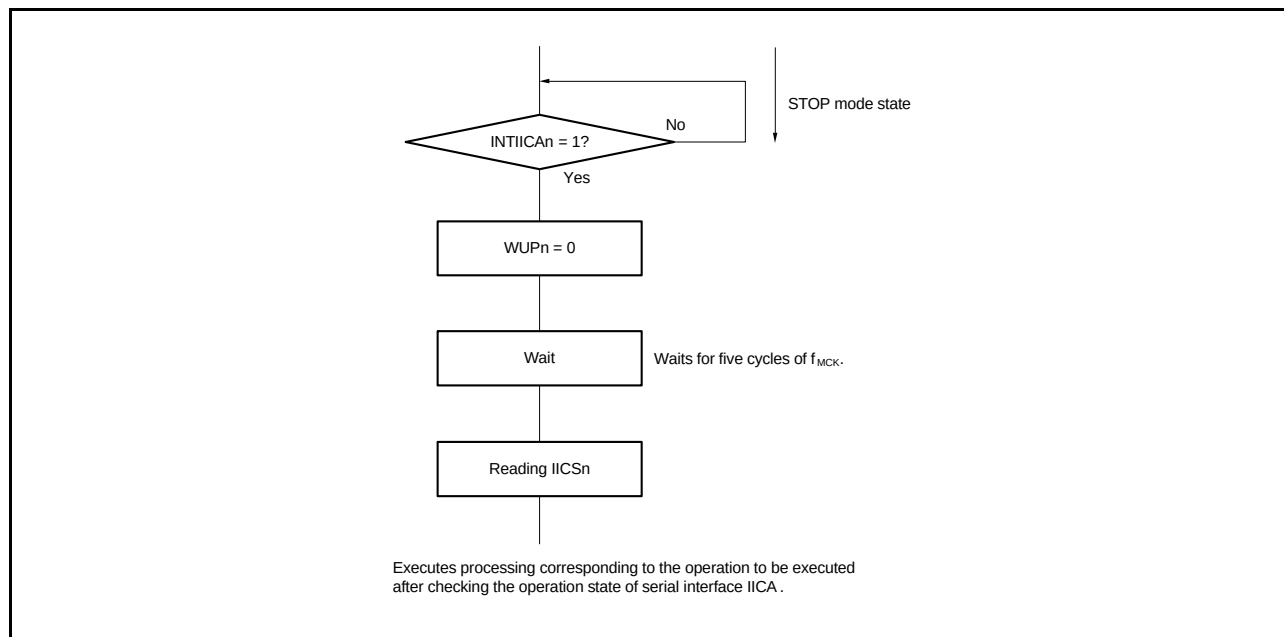
Figure 14 - 22 shows the flow for setting WUPn = 1 and **Figure 14 - 23** shows the flow for setting WUPn = 0 upon an address match (or when the all address match function is enabled).

Figure 14 - 22 Flow When Setting WUPn = 1



Remark n = 0

Figure 14 - 23 Flow When Setting WUPn = 0 upon Address Match (or When the All Address Match Function is Enabled) (Including Extension Code Reception)

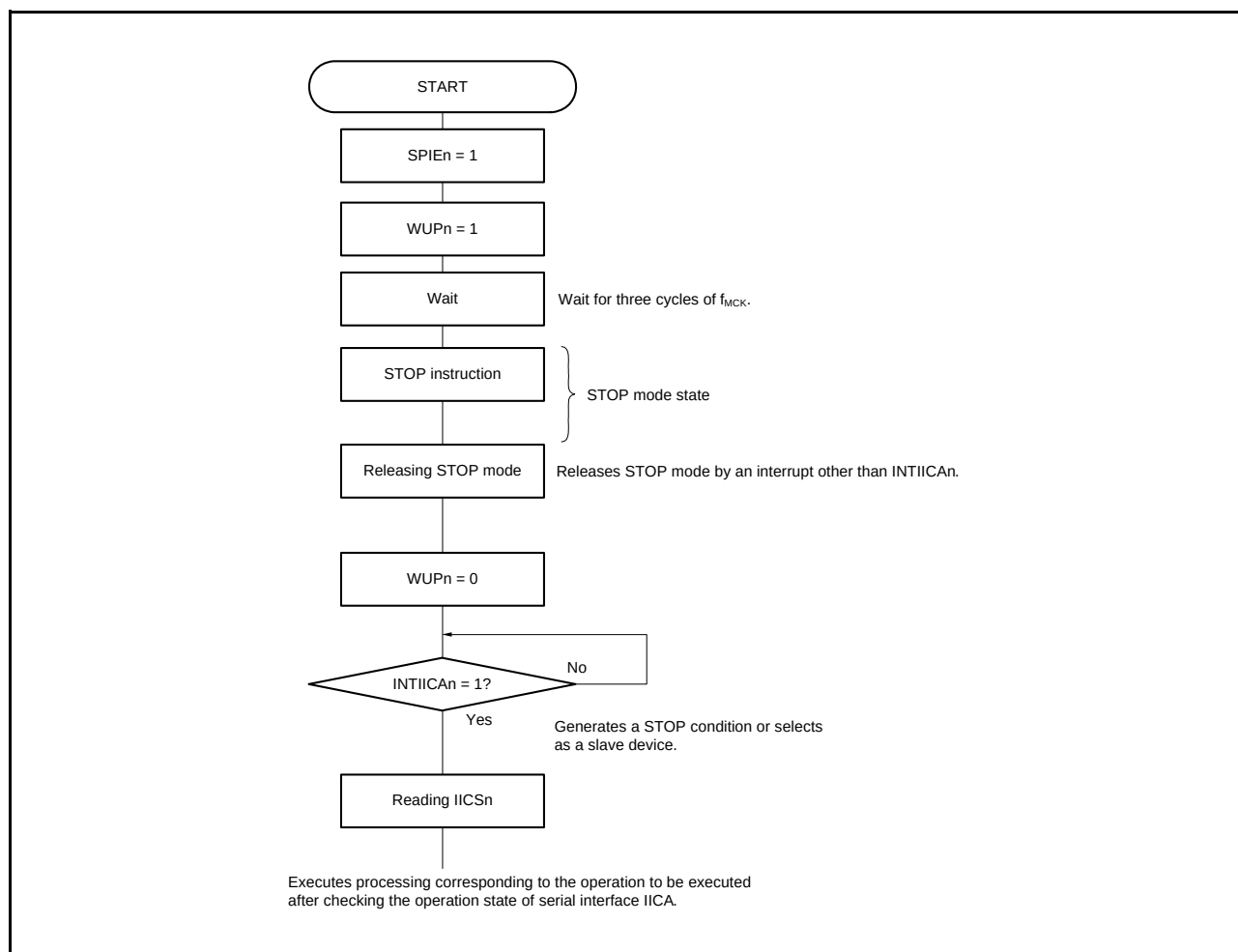


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request signal (INTIICAn) generated from serial interface IICA.

- When operating next IIC communication as master: Flow shown in **Figure 14 - 24**.
- When operating next IIC communication as slave:
 - When released by INTIICAn interrupt: Same as the flow in **Figure 14 - 23**.
 - When released by other than INTIICAn interrupt: Wait for INTIICAn interrupt with WUPn left set to 1.

Remark n = 0

Figure 14 - 24 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn



Remark n = 0

14.5.14 Communication reservation

- (1) When communication reservation function is enabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and exiting from communication)

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used, a start condition is automatically generated and wait state is entered after the bus is released (after a stop condition is detected).

If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.

When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the state of the bus.

- If the bus has been released a start condition is generated
- If the bus has not been released (standby mode) communication reservation

Check whether the communication reservation operates or not by using the MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.

Use software to secure the wait time calculated by the following expression.

Wait time from setting STTn = 1 to checking the MSTSn flag:

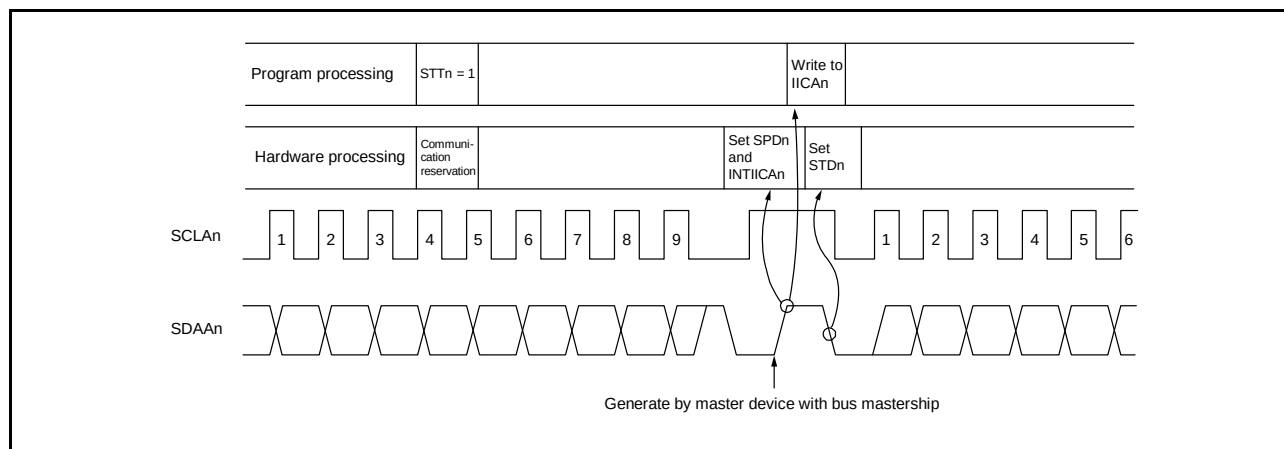
$$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4) / f_{MCK} + t_F \times 2$$

- Remark 1.** IICWLn: IICA low-level width setting register n
 IICWHn: IICA high-level width setting register n
 tF: SDAAn and SCLAn signal falling times
 fMCK: IICA operation clock frequency

- Remark 2.** n = 0

Figure 14 - 25 shows the communication reservation timing.

Figure 14 - 25 Communication Reservation Timing



Remark IICAn: IICA shift register n
STTn: Bit 1 of IICA control register n0 (IICCTLn0)
STDn: Bit 1 of IICA status register n (IICSn)
SPDn: Bit 0 of IICA status register n (IICSn)

Communication reservations are accepted via the timing shown in **Figure 14 - 26**. After bit 1 (STDn) of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.

Figure 14 - 26 Timing for Accepting Communication Reservations

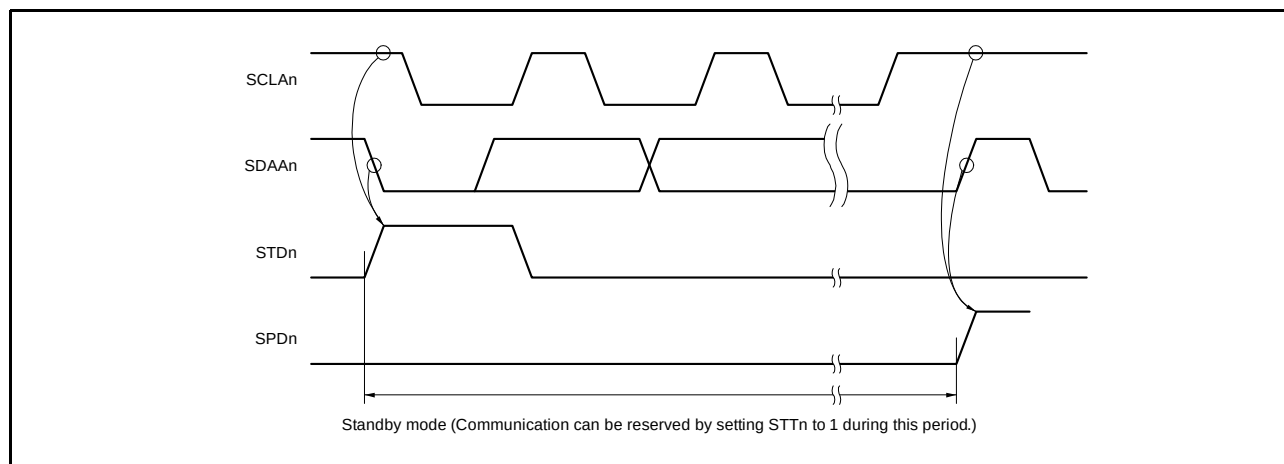


Figure 14 - 27 shows the communication reservation protocol.

Remark n = 0

```

graph TD
    DI[DI] --> SET1[SET1 STTn]
    SET1 --> Define[Define communication reservation]
    Define --> Wait[Wait]
    Wait --> Decision{MSTS n = 0?}
    Decision -- Yes --> EI[EI]
    Decision -- No --> Cancel[Cancel communication reservation]
    Cancel --> MOV[MOV IICAn, #xxH]
    MOV --> EI
    EI --> Decision

```

DI

SET1 STTn
Sets STTn flag (communication reservation)

Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)

Wait
Secures wait time^{Note 1} by software.

(Communication reservation)^{Note 2}

MSTS n = 0?

Yes

No
(Generate start condition)

Cancel communication reservation
Clear user flag

MOV IICAn, #xxH
IICAn write operation

EI

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTS_n: Bit 7 of IICA status register n (IICS_n)
IICAn: IICA shift register n
IICWL_n: IICA low-level width setting register n
IICWH_n: IICA high-level width setting register n
tr: SDAAn and SCLAn signal falling times
fMCK: IICA operation clock frequency

Remark 2. $n = 0$

- (2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The states where the bus is not in use consist of the following two states.

- When arbitration results in neither master nor slave operation
- While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of the IICCTLn0 register to 1 and exiting from communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register). It takes up to 5 cycles of fMCK until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure the time by software.

Remark n = 0

14.5.15 Cautions

(1) When STCENn = 0

Immediately after I²C operation is enabled (IICEn = 1), the bus is recognized as being in a communications state (IICBSYn = 1) regardless of its actual state. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

- <1> Set IICA control register n1 (IICCTLn1).
- <2> Set bit 7 (IICEn) of IICA control register n0 (IICCTLn0) to 1.
- <3> Set bit 0 (SPTn) of the IICCTLn0 register to 1.

(2) When STCENn = 1

Immediately after I²C operation is enabled (IICEn = 1), the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state. To generate the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I²C communications are already in progress

If I²C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the IICA recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code or the all address match function is enabled, ACK is returned, but this interferes with other I²C communications. To avoid this, start the IICA in the following sequence.

- <1> Clear bit 4 (SPIEn) of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
- <2> Set bit 7 (IICEn) of the IICCTLn0 register to 1 to enable the operation of the IICA.
- <3> Wait for detection of the start condition.
- <4> Set bit 6 (LRELn) of the IICCTLn0 register to 1 before ACK is returned (4 to 72 cycles of fMCK after setting the IICEn bit to 1), to forcibly disable detection.

(4) Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n (IICAn) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIEn bit to 1 when the MSTSn bit (bit 7 of the IICA status register n (IICSn)) is detected by software.

Remark n = 0

14.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single-master system

The flowchart when using the RL78/G22 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multi-master system

In the I²C bus multi-master system, whether the bus is released or used cannot be judged by the I²C bus specifications when a device takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/G22 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/G22 loses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

An example of when the RL78/G22 is used as the I²C bus slave is shown below.

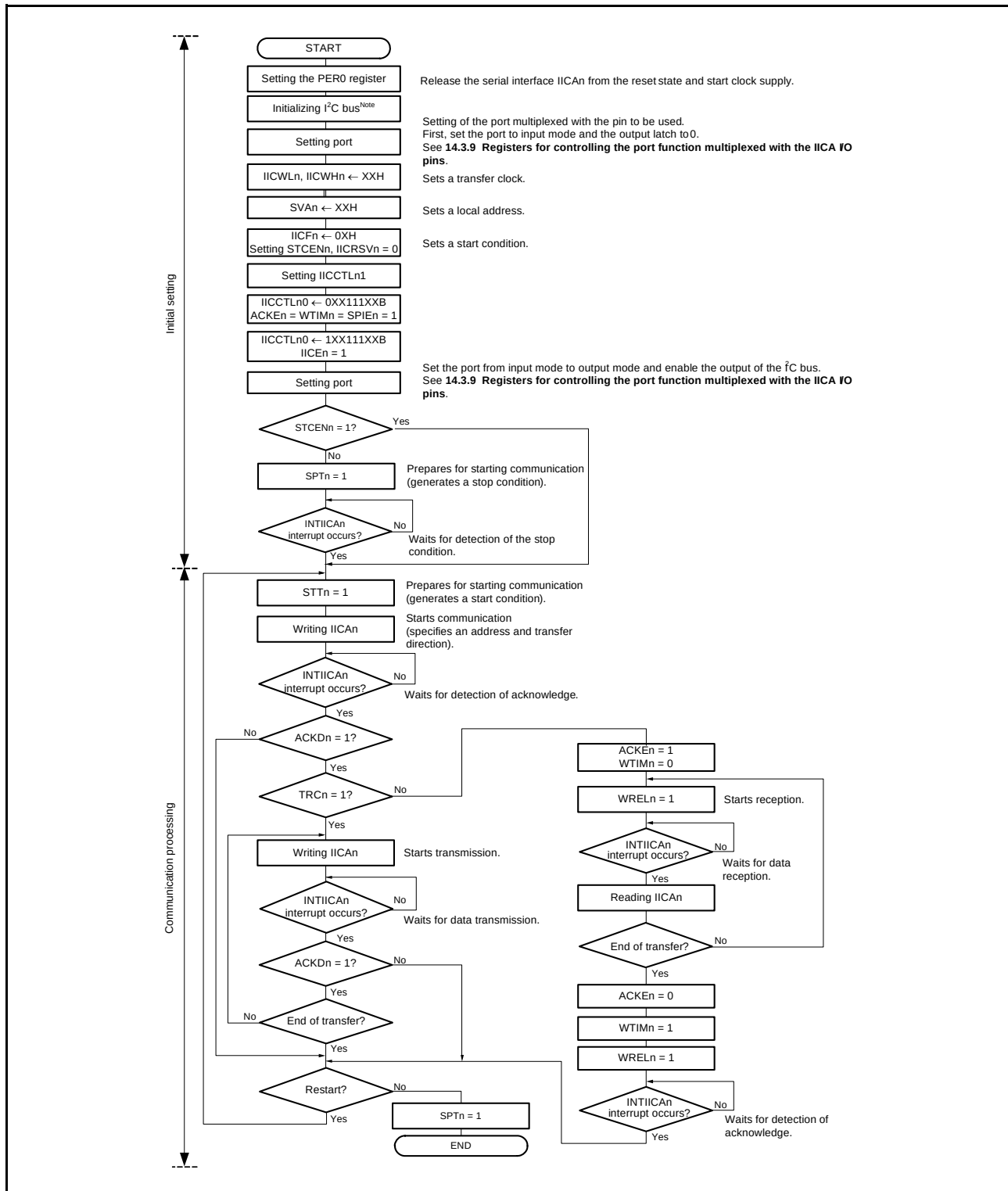
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communications state is judged and the result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark n = 0

(1) Master operation in single-master system

Figure 14 - 28 Master Operation in Single-Master System



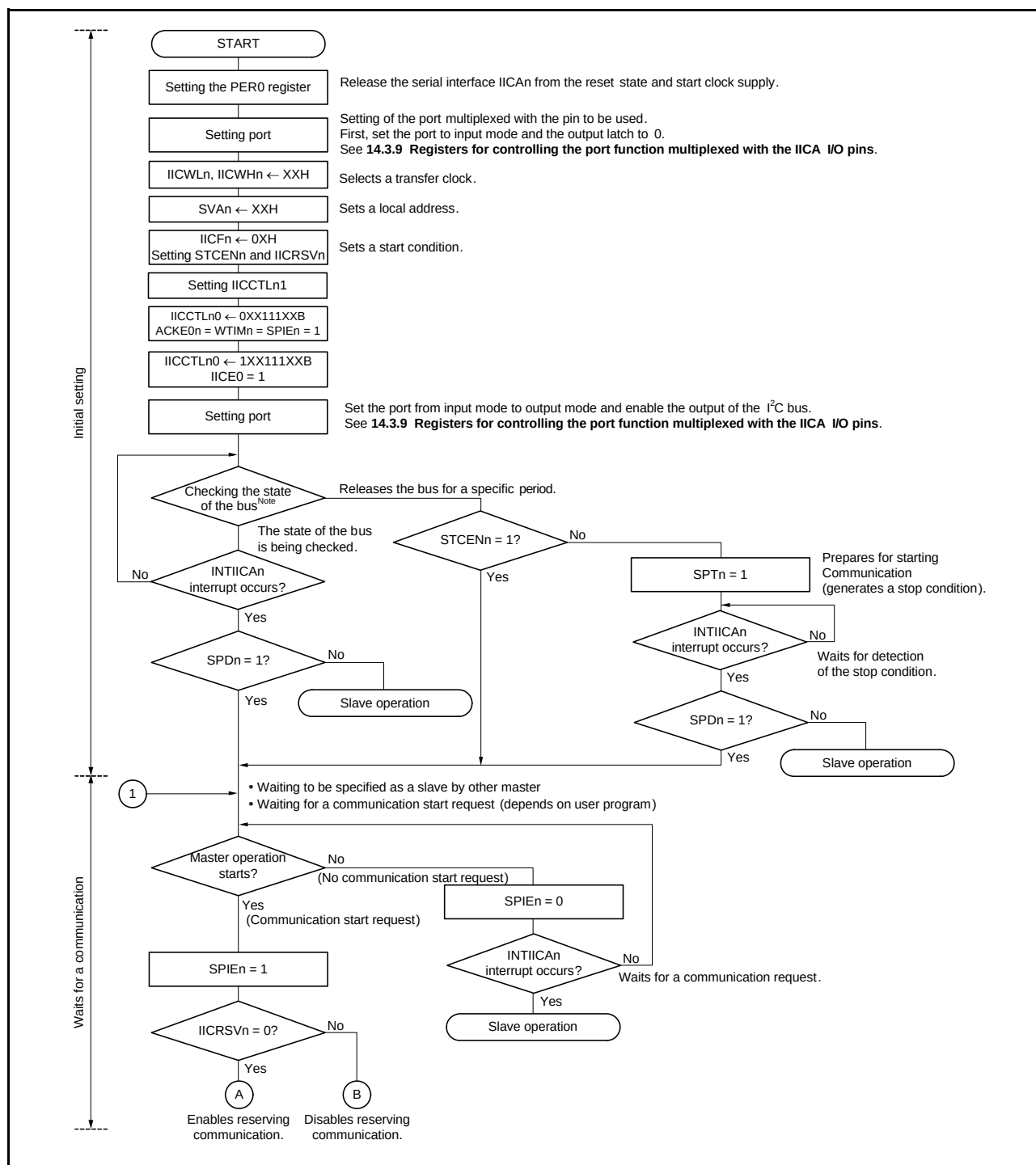
Note Release (SCLAn and SDAAn pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. n = 0

(2) Master operation in multi-master system

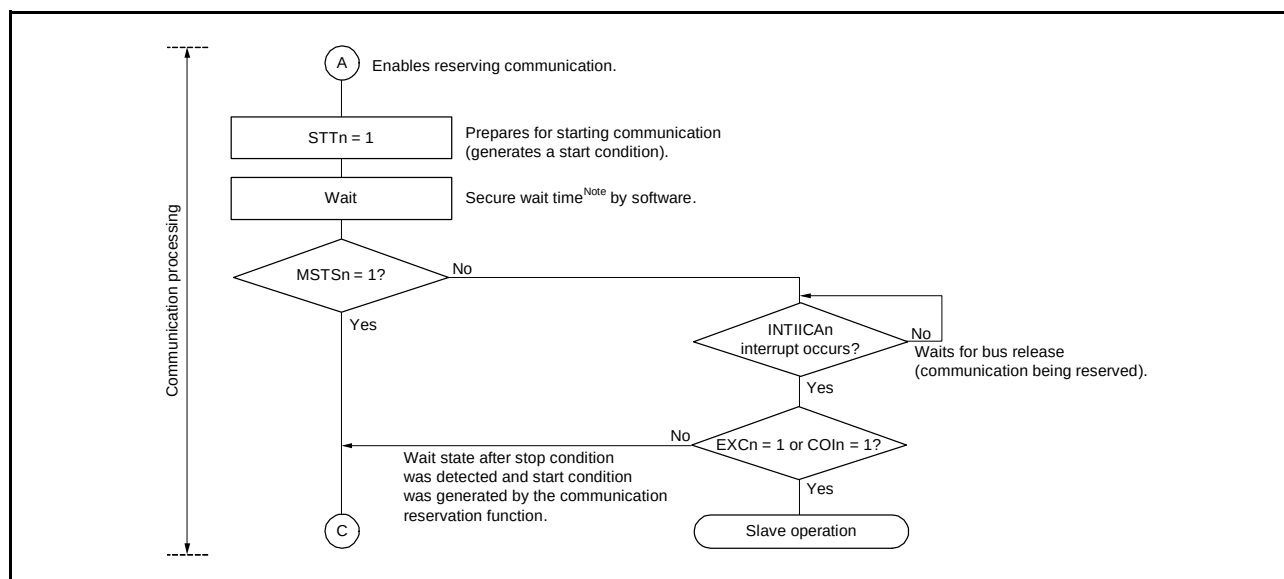
Figure 14 - 29 Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame).
If the SDAAn pin is constantly at low level, decide whether to release the I²C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

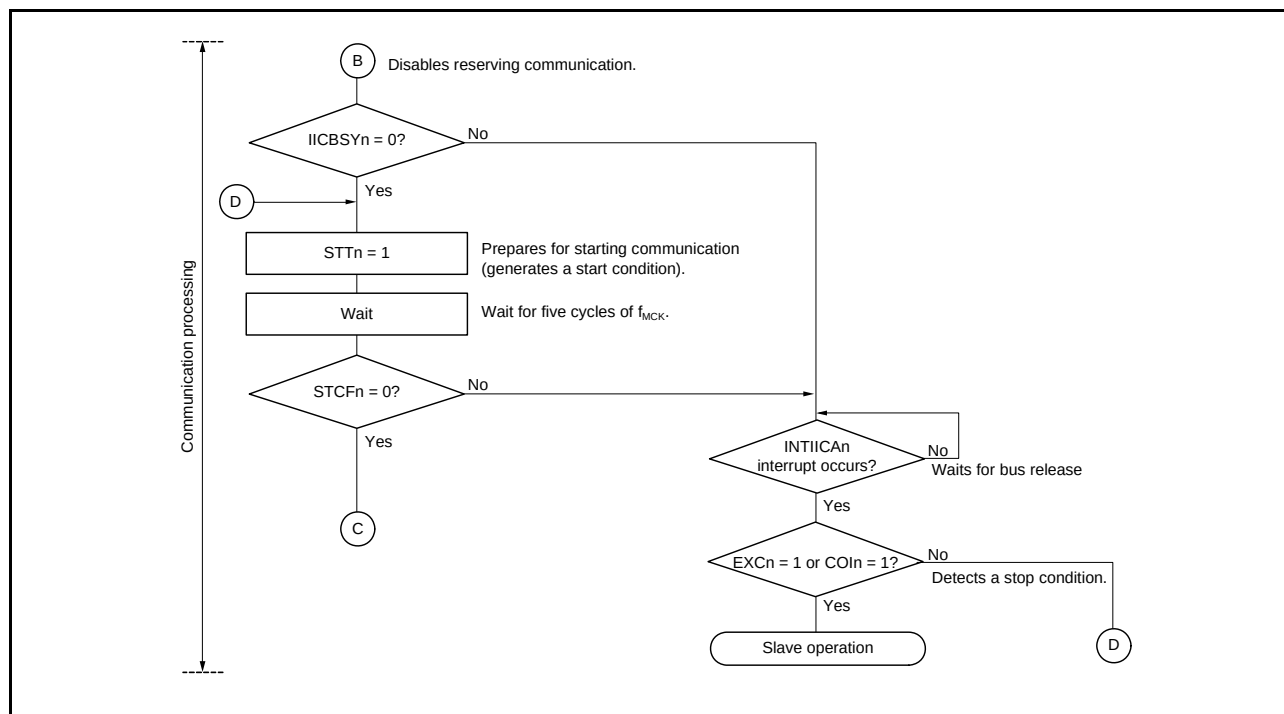
Remark n = 0

Figure 14 - 29 Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

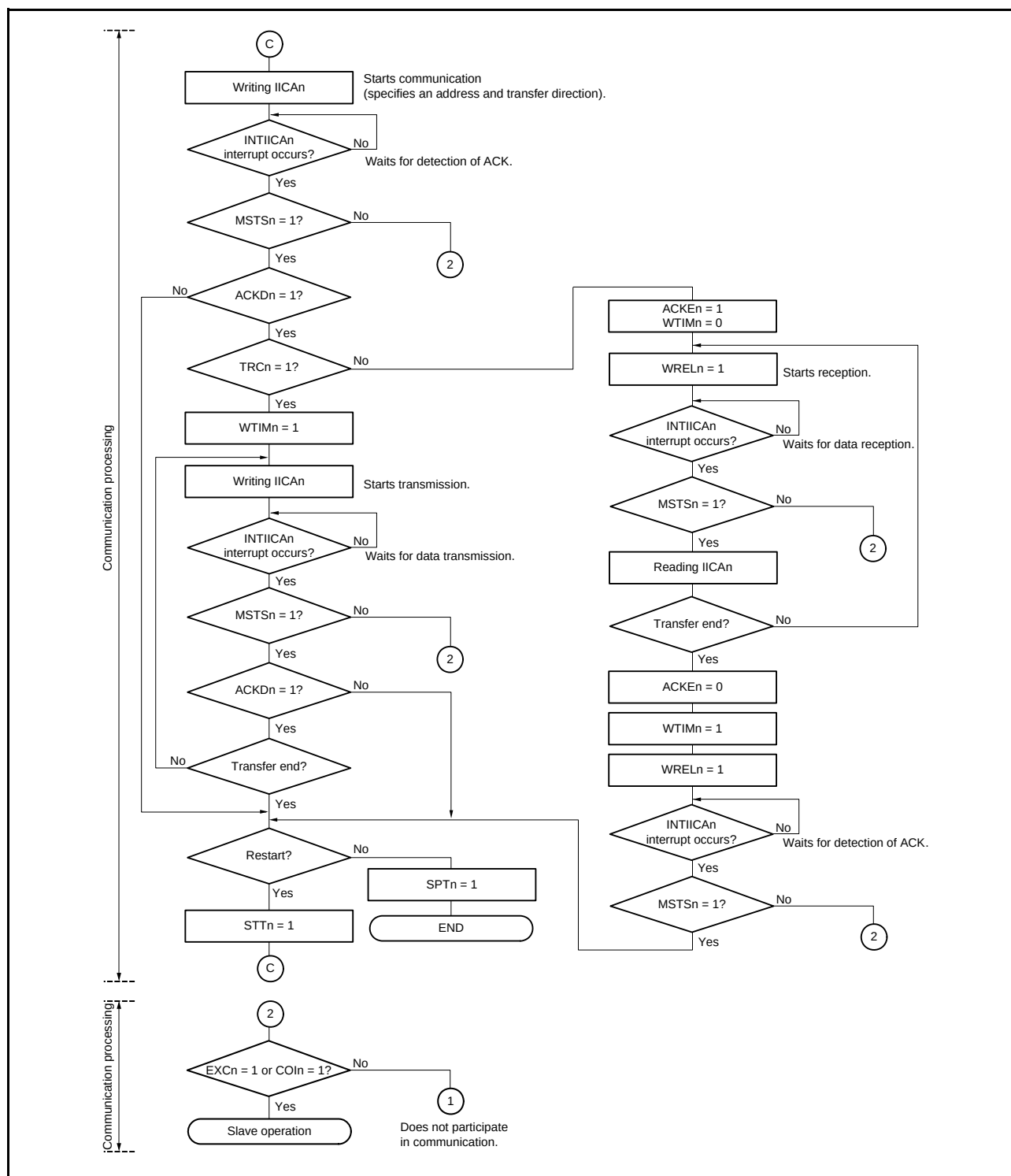
$$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4)/f_{MCK} + t_F \times 2$$



Remark 1. IICWLn: IICA low-level width setting register n
 IICWHn: IICA high-level width setting register n
 tF: SDAAn and SCLAn signal falling times
 fMCK: IICA operation clock frequency

Remark 2. n = 0

Figure 14 - 29 Master Operation in Multi-Master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the state by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

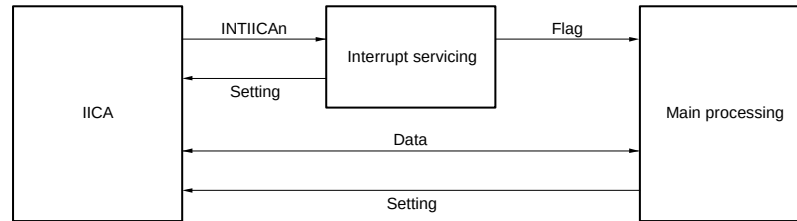
Remark 4. n = 0

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the state of operating such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the all address match function is disabled and the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs state transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communications states.

- Clear mode: State in which data communications are not in progress
- Communication mode: State in which data communications are in progress (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

Remark n = 0

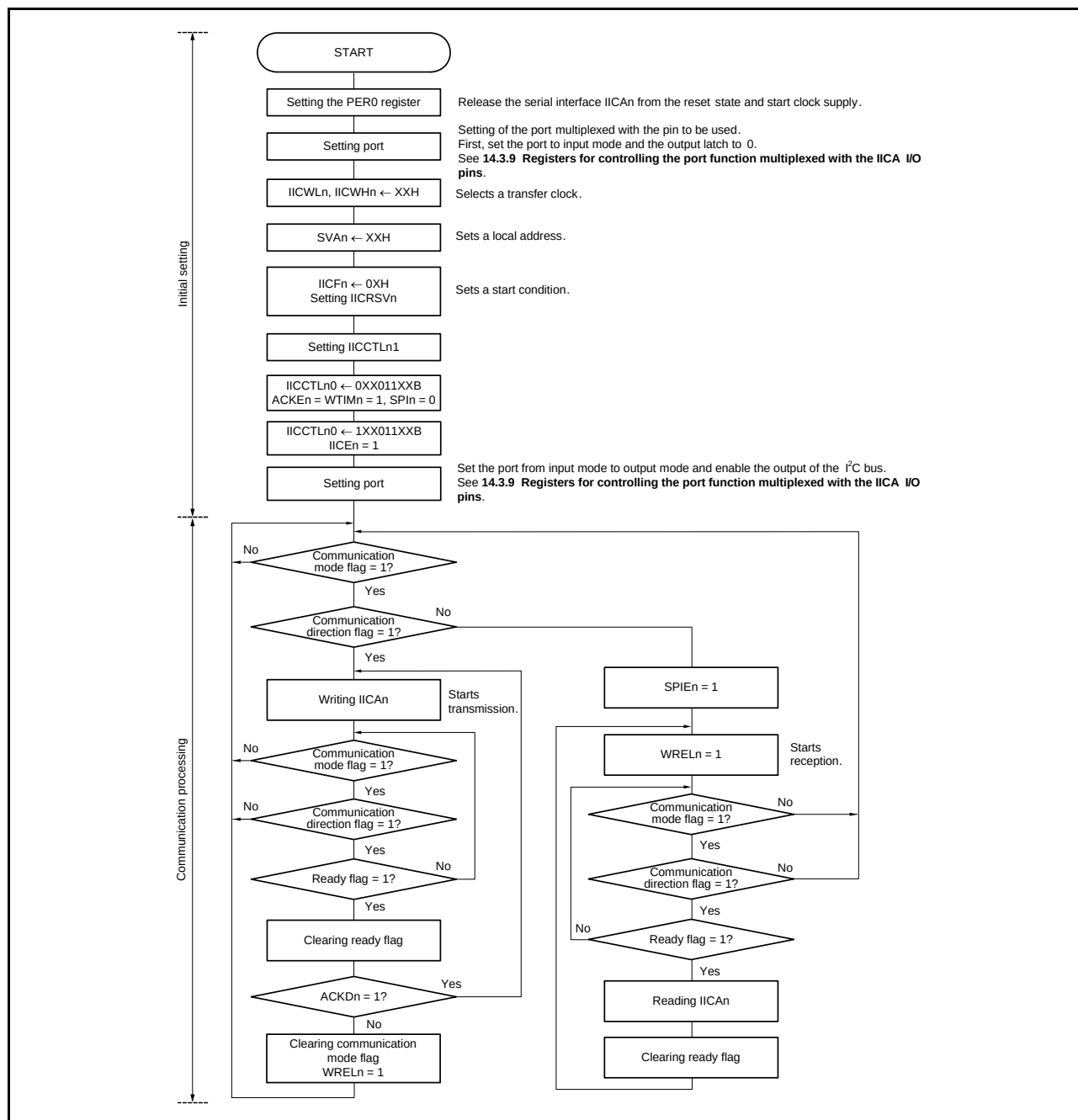
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag. Processing of the stop condition and start condition is performed by an interrupt. Here, check the state by using the flags.

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communications state occurs in this way.

Figure 14 - 30 Slave Operation Flowchart (1)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

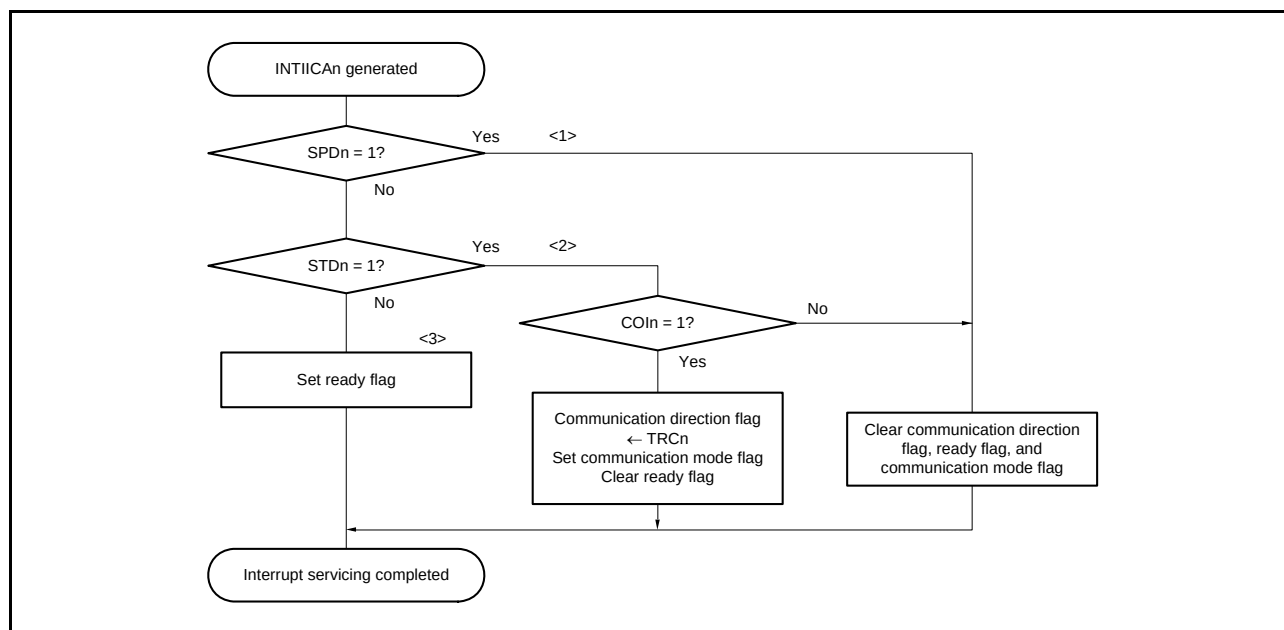
Remark 2. n = 0

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that the all address match function is disabled and no extension code is used). The INTIICAn interrupt checks the state, and the following operations are performed.

- <1> Communication is stopped if the stop condition is issued.
- <2> If the start condition is issued, the address is checked and communication is completed if the address does not match.
If the address matches, the communication mode is set, wait is canceled, and processing returns from the interrupt (the ready flag is cleared).
- <3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I²C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 14 - 31 Slave Operation Flowchart (2)**.

Figure 14 - 31 Slave Operation Flowchart (2)



Remark n = 0

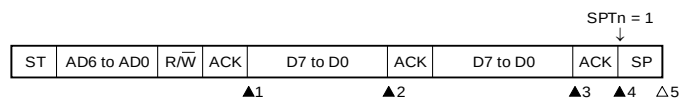
14.5.17 Timing of I²C interrupt request signal (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

Remark 1. ST: Start condition
AD6 to AD0: Address
 $\overline{R/\overline{W}}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0

- (1) Master device operation
- (a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)
 - (i) When $WTIMn = 0$



- ▲1: IICSn = 1000×110B
- ▲2: IICSn = 1000×000B
- ▲3: IICSn = 1000×000B (Sets the $WTIMn$ bit to 1)^{Note}
- ▲4: IICSn = 1000××00B (Sets the $SPTn$ bit to 1)
- Δ5: IICSn = 00000001B

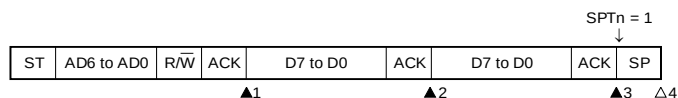
Note To generate a stop condition, set the $WTIMn$ bit to 1 and change the timing for generating the $INTIICAn$ interrupt request signal.

Remark ▲: Always generated

Δ: Generated only when $SPIEn = 1$

×: Don't care

- (ii) When $WTIMn = 1$



- ▲1: IICSn = 1000×110B
- ▲2: IICSn = 1000×100B
- ▲3: IICSn = 1000××00B (Sets the $SPTn$ bit to 1)
- Δ4: IICSn = 00000001B

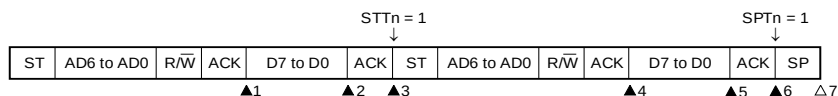
Remark ▲: Always generated

Δ: Generated only when $SPIEn = 1$

×: Don't care

Remark $n = 0$

- (b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)
(i) When WTIMn = 0



- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)^{Note 1}
 ▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0^{Note 2}, sets the STTn bit to 1)
 ▲4: IICSn = 1000×110B
 ▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1)^{Note 3}
 ▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)
 Δ7: IICSn = 00000001B

Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

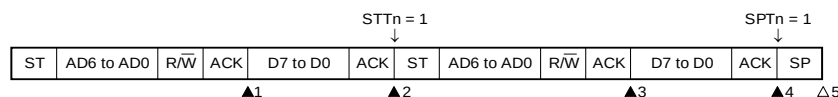
Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

×: Don't care

- (ii) When WTIMn = 1



- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000××00B (Sets the STTn bit to 1)
 ▲3: IICSn = 1000×110B
 ▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)
 Δ5: IICSn = 00000001B

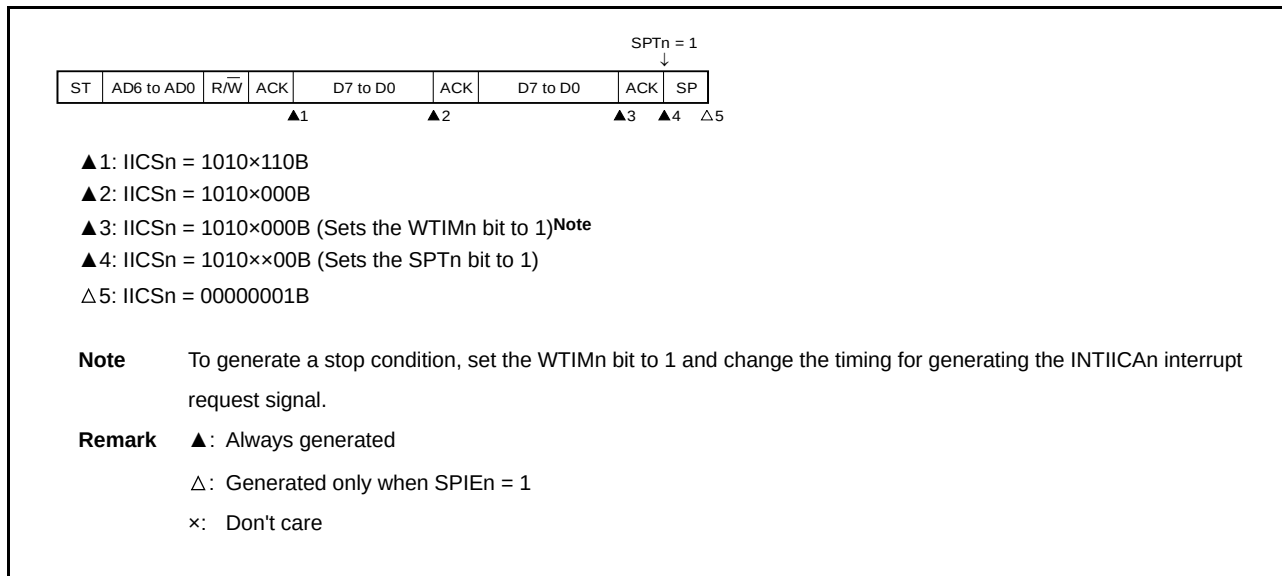
Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

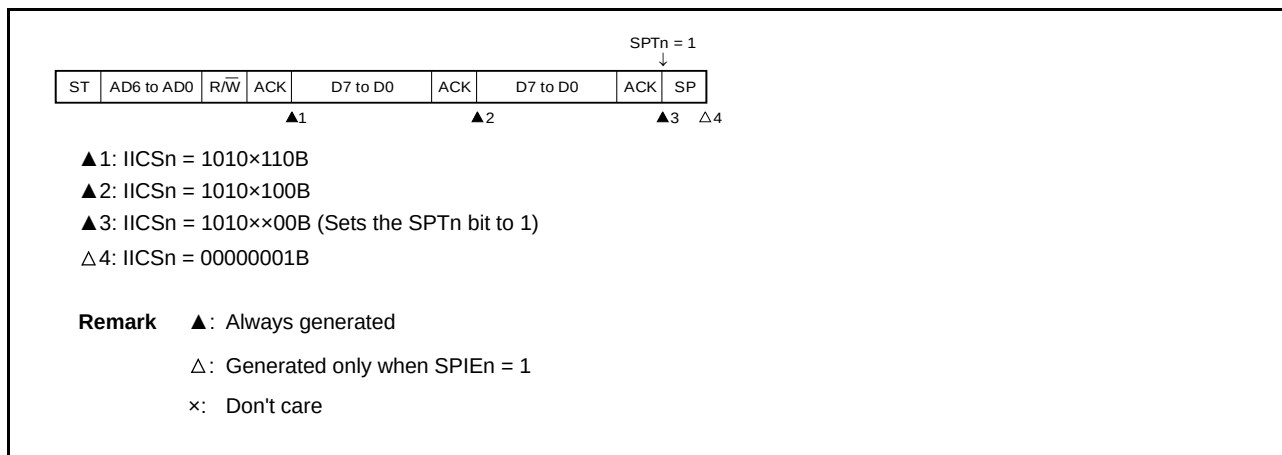
×: Don't care

Remark n = 0

- (c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)
(i) When WTIMn = 0

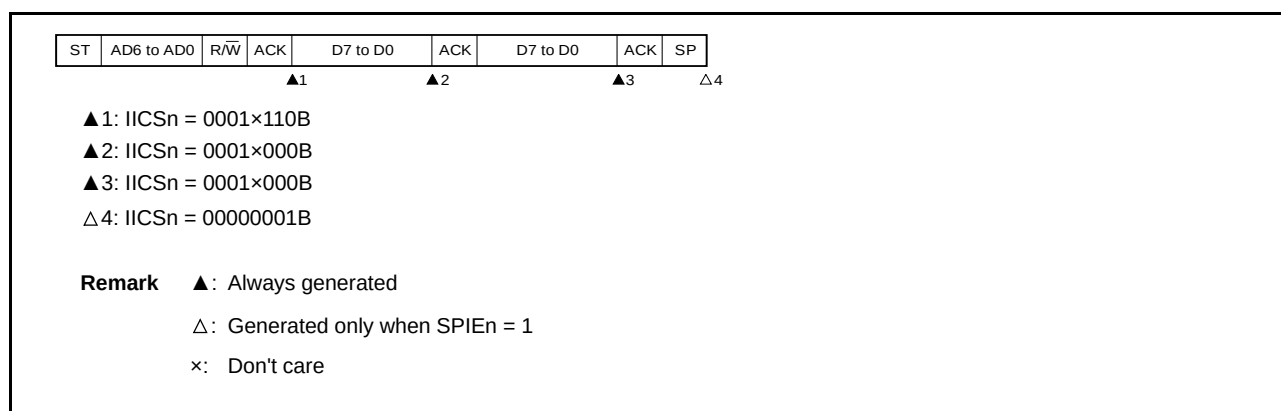


- (ii) When WTIMn = 1

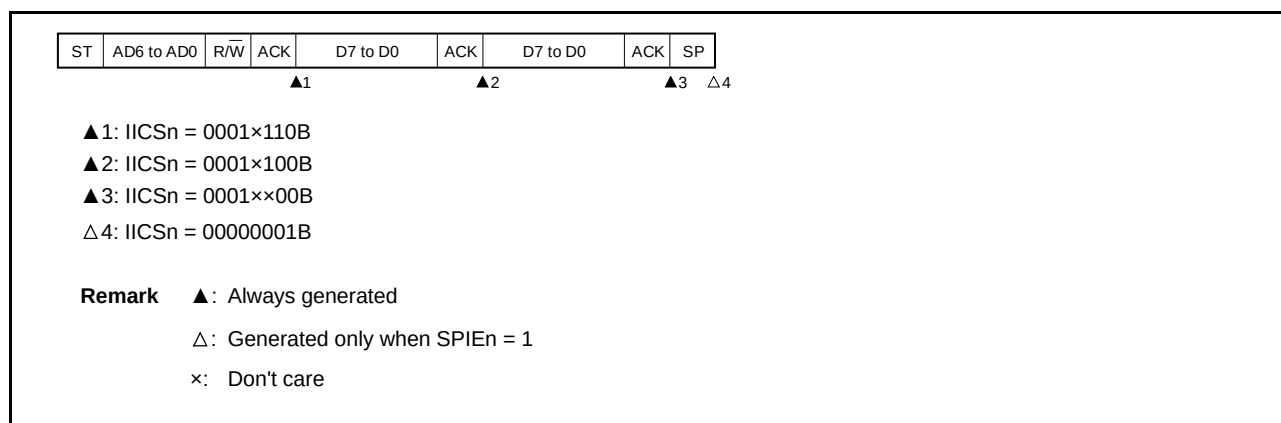


Remark n = 0

- (2) Slave device operation (slave address data reception)
(a) Start ~ Address ~ Data ~ Data ~ Stop
(i) When WTIMn = 0



- (ii) When WTIMn = 1



Remark n = 0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn, the all address match function is disabled)

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches with SVAn, the all address match function is disabled)

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0

(after restart, does not match address (= extension code, the all address match function is disabled))

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1

(after restart, does not match address (= extension code, the all address match function is disabled))

ST	AD6 to AD0	R $\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R $\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2			▲3	▲4		▲5	▲6

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×110B

▲5: IICSn = 0010××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

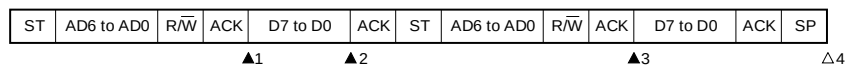
×: Don't care

Remark n = 0

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0

(after restart, does not match address (= not extension code, the all address match function is disabled))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 00000×10B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1

(after restart, does not match address (= not extension code, the all address match function is disabled))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 00000×10B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

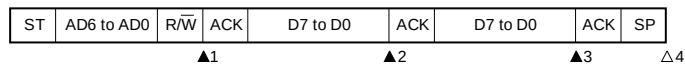
Remark n = 0

- (3) Slave device operation (when receiving extension code and the all address match function is disabled)

The device is always participating in communication when it receives an extension code.

- (a) Start ~ Code ~ Data ~ Data ~ Stop

- (i) When WTIMn = 0



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

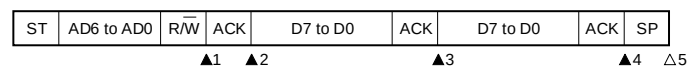
△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

- (ii) When WTIMn = 1



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn, the all address match function is disabled)

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1 (after restart, matches with SVAn, the all address match function is disabled)

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3			▲4		▲5	△6

▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0001×110B

▲5: IICSn = 0001××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

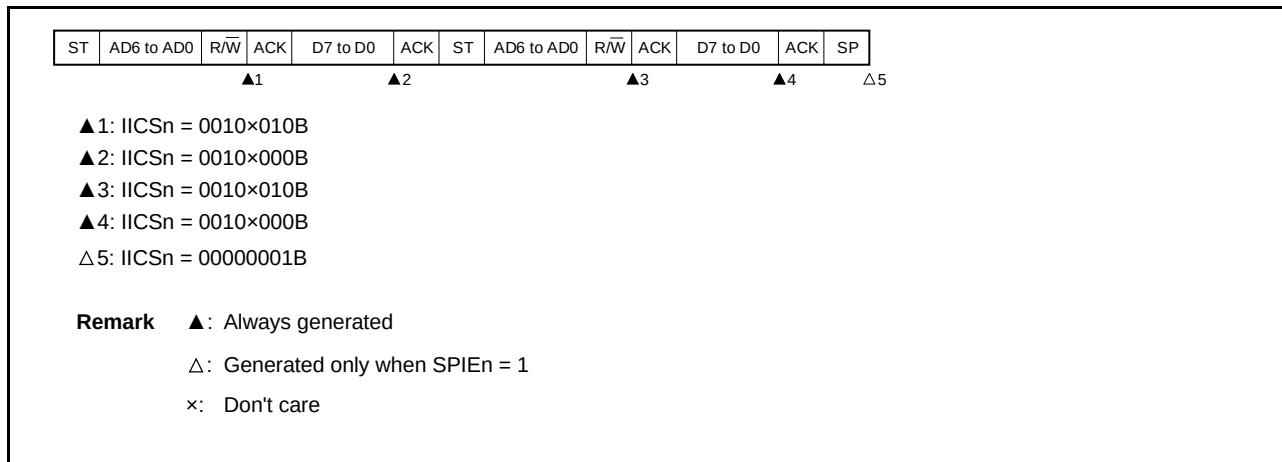
△: Generated only when SPIEn = 1

×: Don't care

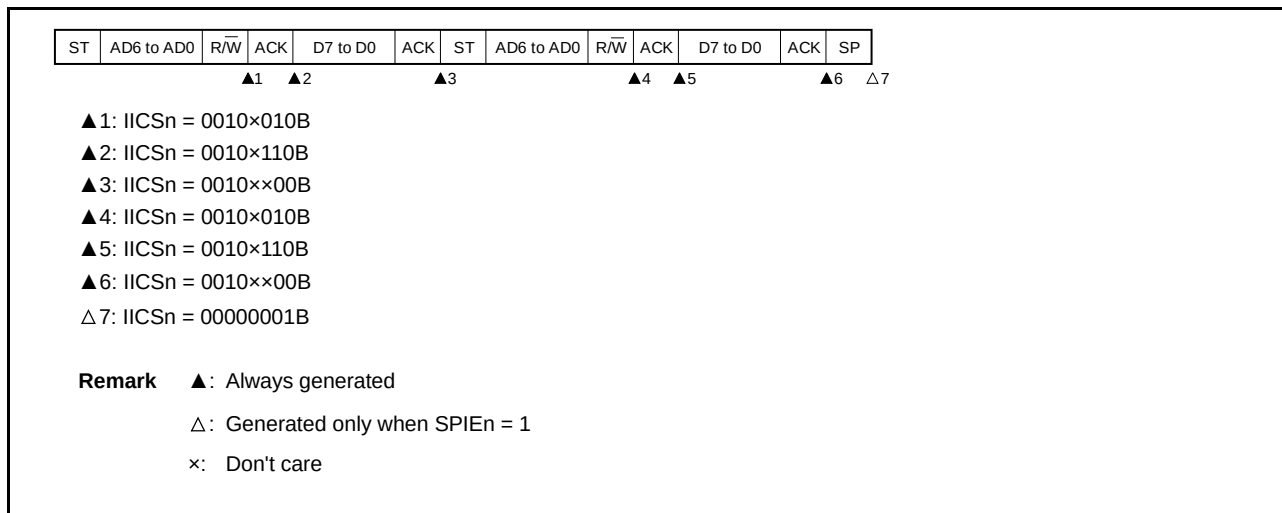
Remark n = 0

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, extension code reception, the all address match function is disabled)



(ii) When WTIMn = 1 (after restart, extension code reception, the all address match function is disabled)

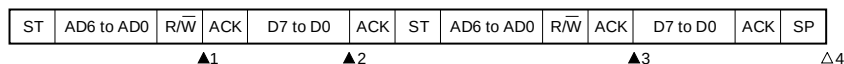


Remark n = 0

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0

(after restart, does not match address (= not extension code, the all address match function is disabled))



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 00000×10B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(ii) When WTIMn = 1

(after restart, does not match address (= not extension code, the all address match function is disabled))



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 00000×10B

△5: IICSn = 00000001B

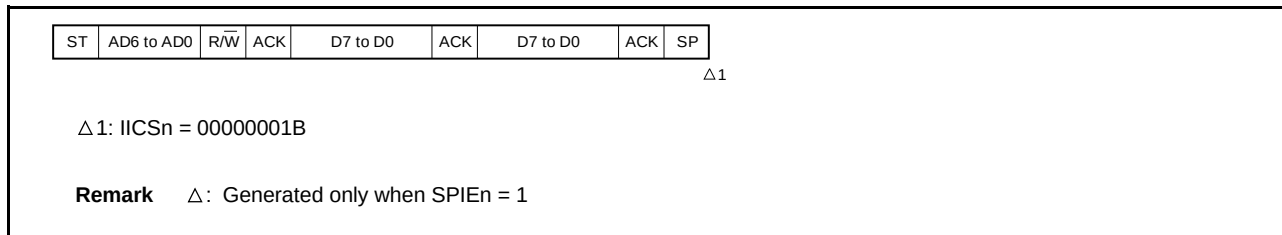
Remark ▲: Always generated

△: Generated only when SPIEn = 1

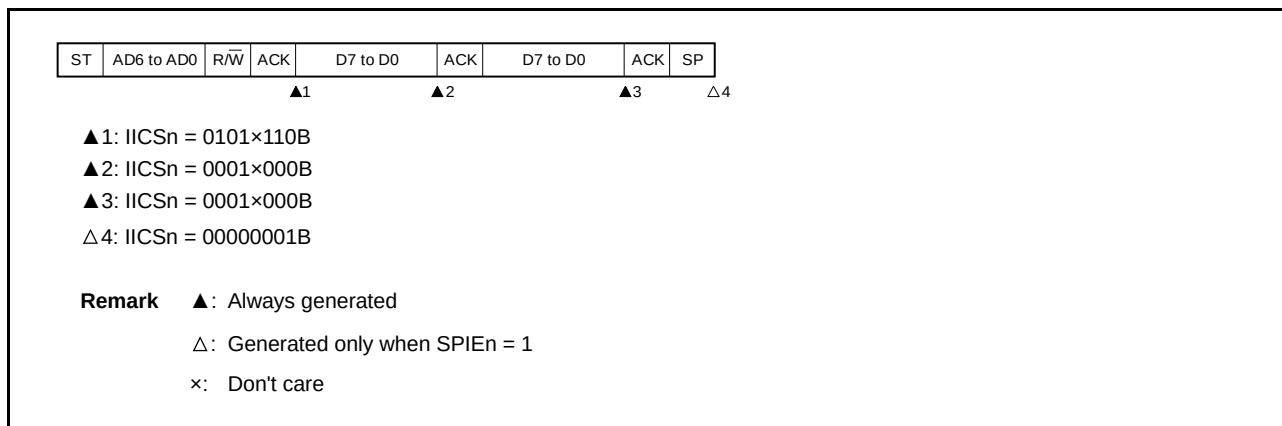
×: Don't care

Remark n = 0

- (4) Operation without communication
(a) Start ~ Code ~ Data ~ Data ~ Stop



- (5) Arbitration loss operation (operation as slave after arbitration loss)
When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.
- (a) When arbitration loss occurs during transmission of slave address data
(i) When WTIMn = 0



Remark n = 0

(ii) When WTIMn = 1

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

(b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)

(i) When WTIMn = 0

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

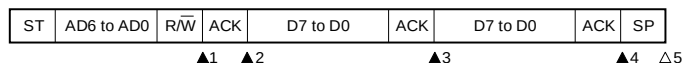
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0

(ii) When WTIMn = 1



▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

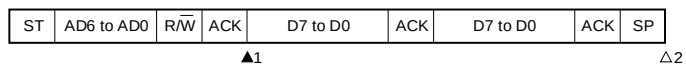
△: Generated only when SPIEn = 1

×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



▲1: IICSn = 01000110B

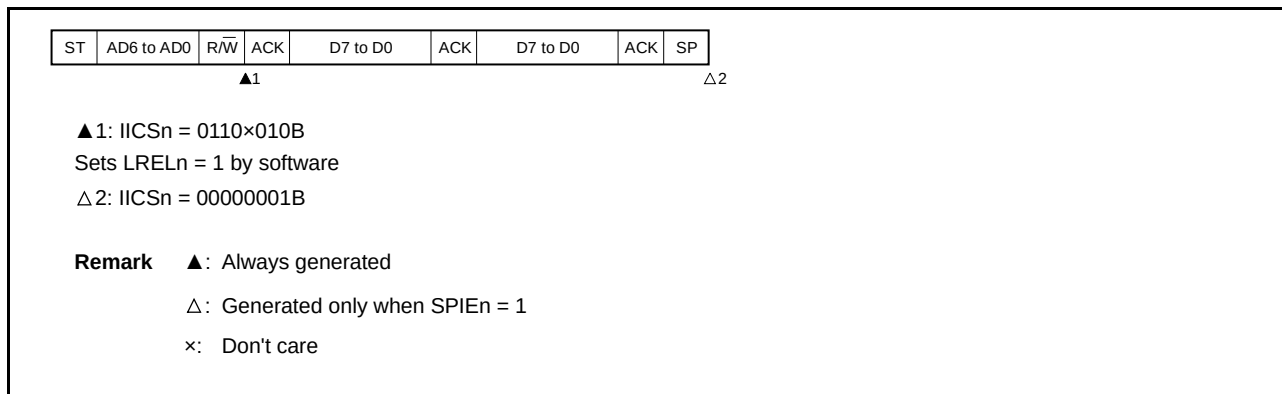
△2: IICSn = 00000001B

Remark ▲: Always generated

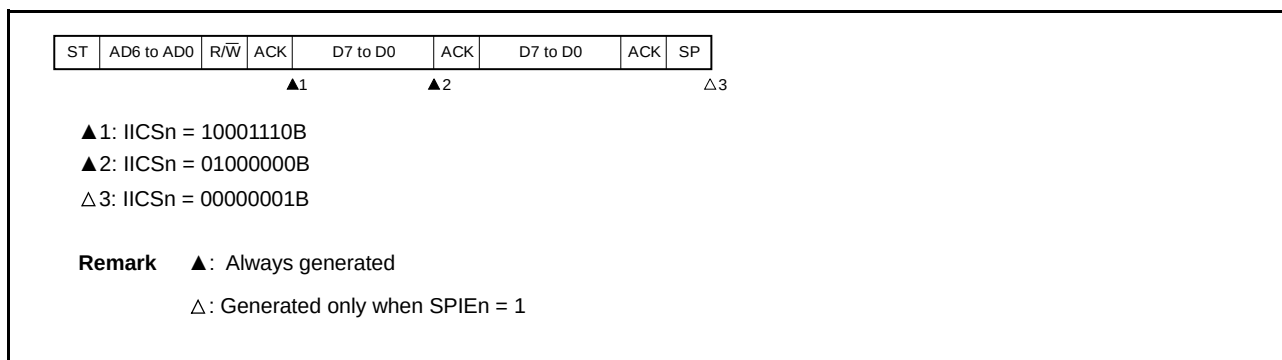
△: Generated only when SPIEn = 1

Remark n = 0

- (b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)

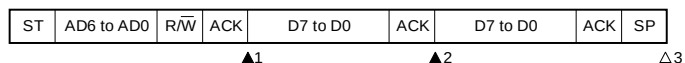


- (c) When arbitration loss occurs during transmission of data
(i) When WTIMn = 0



Remark n = 0

(ii) When WTIMn = 1



▲1: IICSn = 10001110B

▲2: IICSn = 01000100B

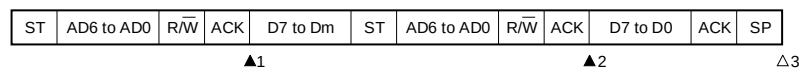
△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatched with SVAn, the all address match function is disabled)



▲1: IICSn = 1000×110B

▲2: IICSn = 01000110B

△3: IICSn = 00000001B

Remark ▲: Always generated

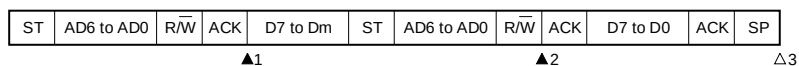
△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

Remark n = 0

(ii) Extension code (the all address match function is disabled)



▲1: IICSn = 1000×110B

▲2: IICSn = 01100010B

Sets LRELn = 1 by software

△3: IICSn = 00000001B

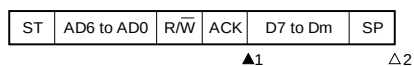
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

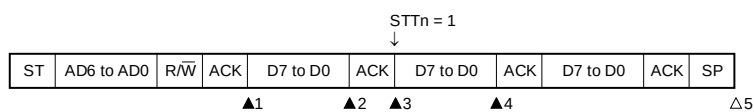
△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

Remark n = 0

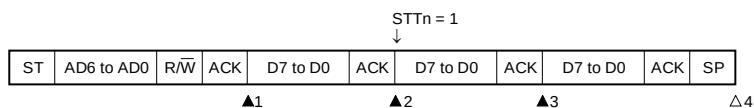
- (f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition
(i) When $WTIMn = 0$



- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000×000B (Sets the $WTIMn$ bit to 1)
 ▲3: IICSn = 1000×100B (Clears the $WTIMn$ bit to 0)
 ▲4: IICSn = 01000000B
 △5: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when $SPIEn = 1$
 ×: Don't care

- (ii) When $WTIMn = 1$



- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000×100B (Sets the $STTn$ bit to 1)
 ▲3: IICSn = 01000100B
 △4: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when $SPIEn = 1$
 ×: Don't care

Remark n = 0

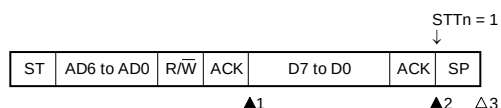
- (g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition
(i) When $WTIMn = 0$



- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000×000B (Sets the $WTIMn$ bit to 1)
 ▲3: IICSn = 1000××00B (Sets the $STTn$ bit to 1)
 △4: IICSn = 01000001B

Remark ▲: Always generated
 △: Generated only when $SPIEn = 1$
 ×: Don't care

- (ii) When $WTIMn = 1$

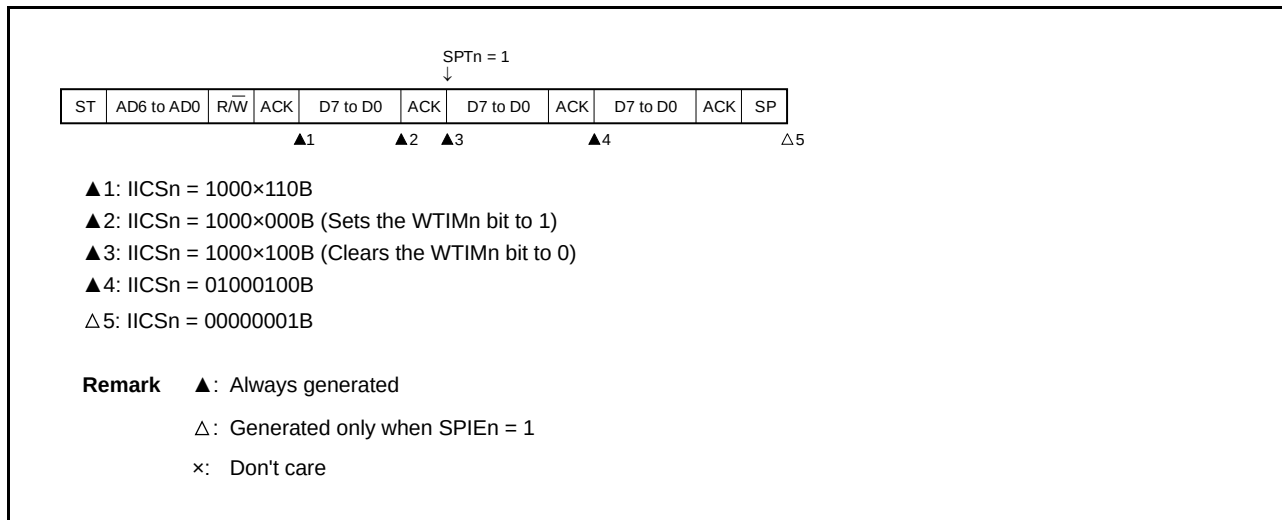


- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000××00B (Sets the $STTn$ bit to 1)
 △3: IICSn = 01000001B

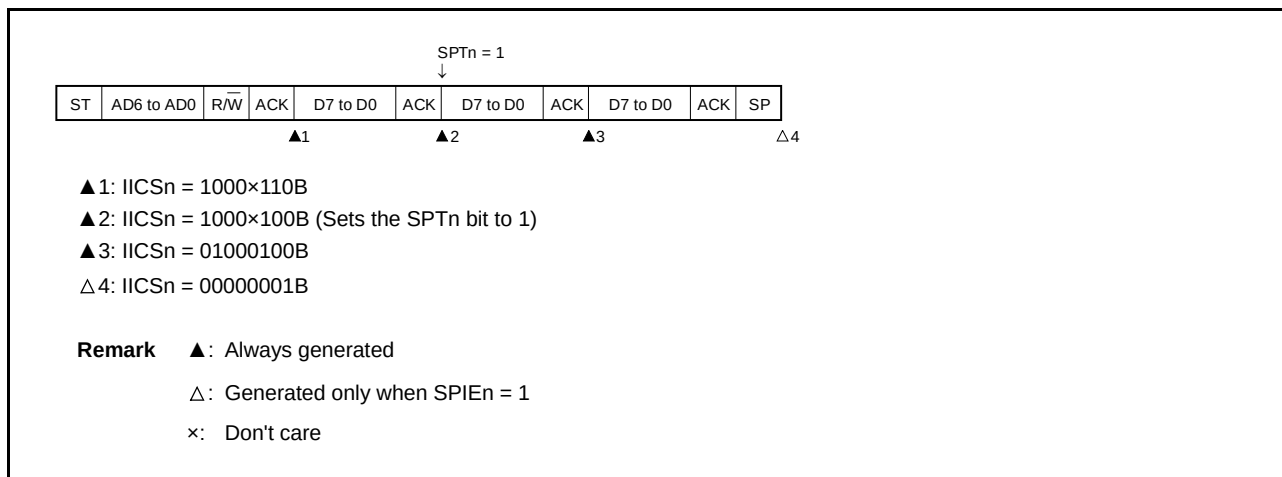
Remark ▲: Always generated
 △: Generated only when $SPIEn = 1$
 ×: Don't care

Remark $n = 0$

- (h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition
(i) When $WTIMn = 0$



- (ii) When $WTIMn = 1$



Remark n = 0

14.6 Timing Charts

When using the I²C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRCn bit (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device.

Figure 14 - 32 and **Figure 14 - 33** show timing charts of the data communication.

The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

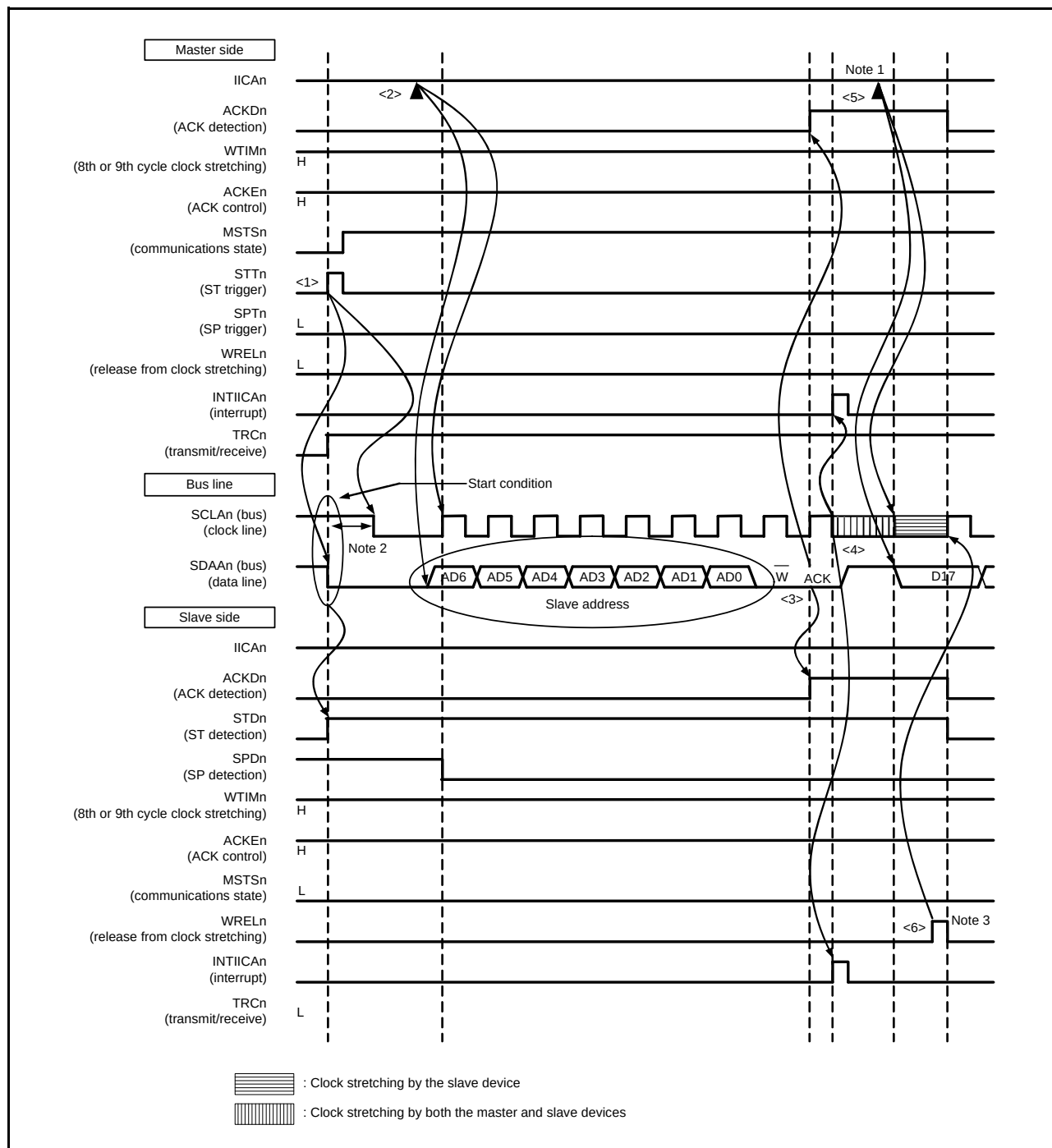
Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

In the timing charts described in this section, it is assumed that the all address match function is disabled.

Remark n = 0

Figure 14 - 32 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (1/4)

(1) Start condition to address to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <1> to <6> in **(1) Start condition to address to data** in **Figure 14 - 32** are explained below.

- <1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn).
- When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.
- <2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.
- <3> In the slave device if the address received matches the address (SVAn value) of a slave device^{Note}, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)^{Note}.
- <5> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.
- <6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state.
The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 14 - 32** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 32 (1) Start condition to address to data shows the processing from <1> to <6>,

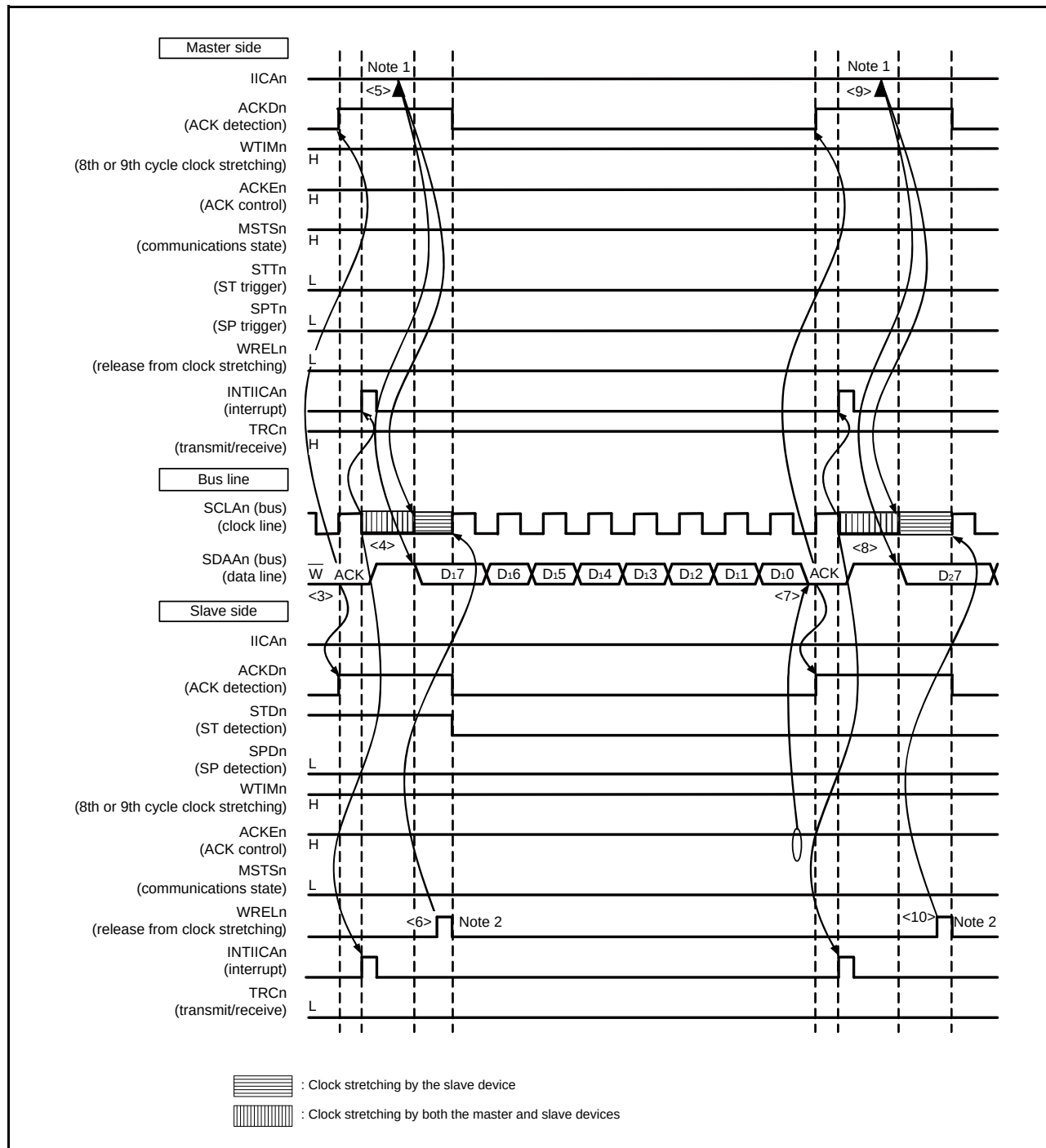
Figure 14 - 32 (2) Address to data to data shows the processing from <3> to <10>, and

Figure 14 - 32 (3) Data to data to stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 14 - 32 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (2/4)

(2) Address to data to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <3> to <10> in **(2) Address to data to data** in **Figure 14 - 32** are explained below.

- <3> In the slave device if the address received matches the address (SVAn value) of a slave device^{Note}, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)^{Note}.
- <5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.
- <6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.
- <7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <9> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.
- <10> The slave device reads the received data and releases the clock stretch state (WRELn = 1). The master device then starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state.
The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 14 - 32** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 32 (1) Start condition to address to data shows the processing from <1> to <6>,

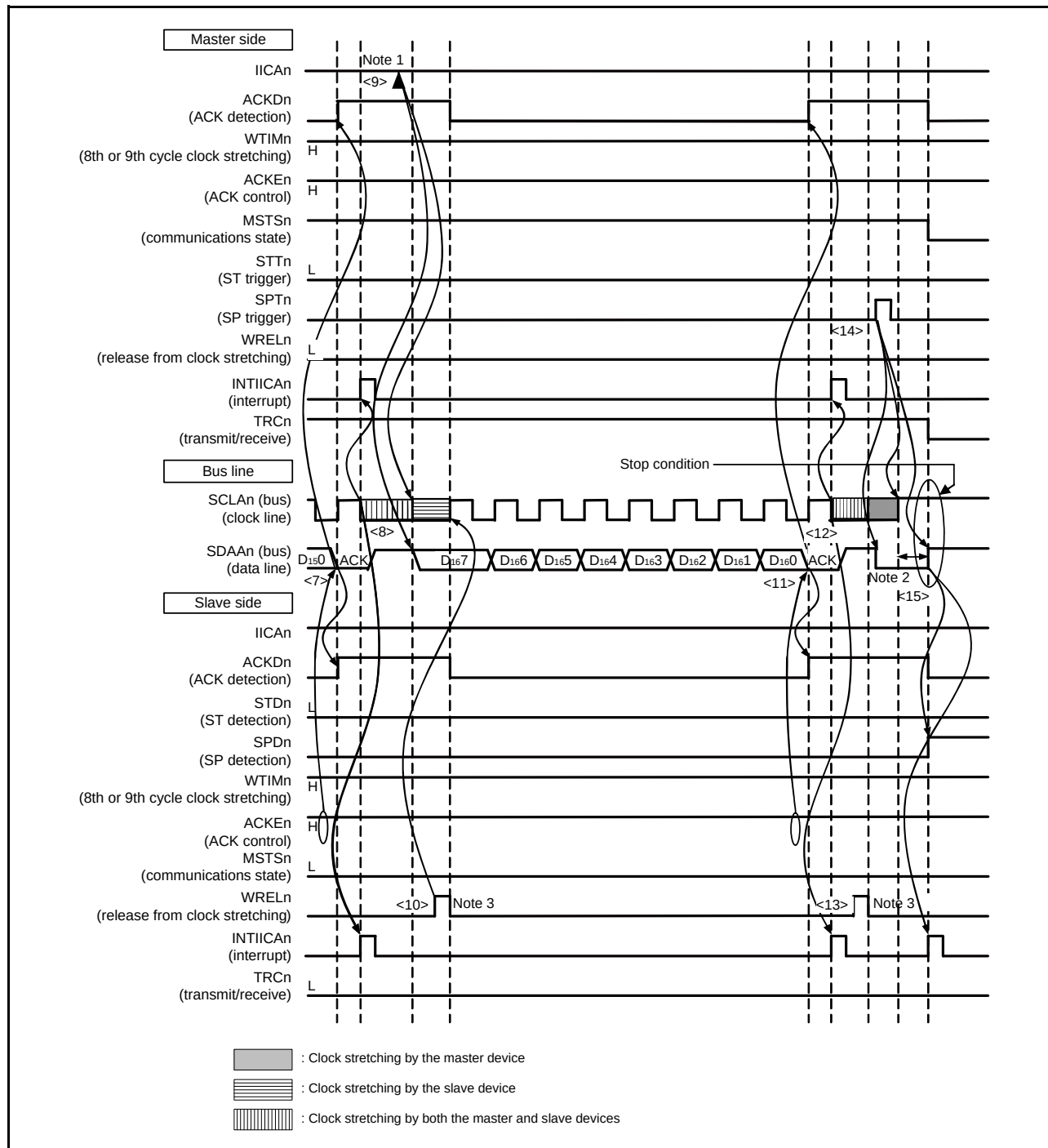
Figure 14 - 32 (2) Address to data to data shows the processing from <3> to <10>, and

Figure 14 - 32 (3) Data to data to stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 14 - 32 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (3/4)

(3) Data to data to stop condition



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0

The meanings of <7> to <15> in **(3) Data to data to stop condition** in **Figure 14 - 32** are explained below.

- <7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.
- <10> The slave device reads the received data and releases the clock stretch state (WRELn = 1). The master device then starts transferring data to the slave device.
- <11> When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <12> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <13> The slave device reads the received data and releases the clock stretch state (WRELn = 1).
- <14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e. SCLAn =1 and SDAAn changes from 0 to 1).
- <15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in **Figure 14 - 32** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 32 (1) Start condition to address to data shows the processing from <1> to <6>,

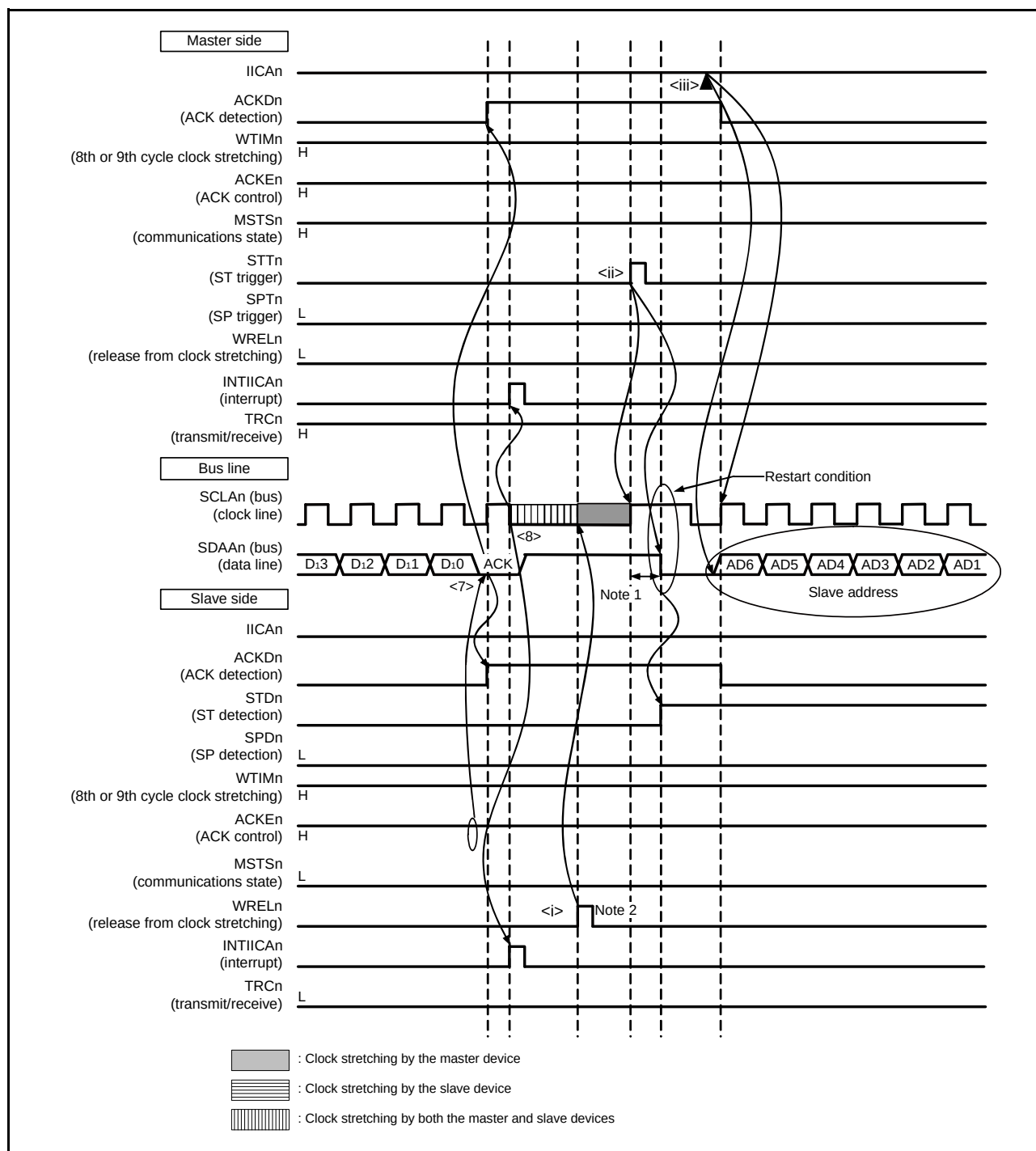
Figure 14 - 32 (2) Address to data to data shows the processing from <3> to <10>, and

Figure 14 - 32 (3) Data to data to stop condition shows the processing from <7> to <15>.

Remark 2. n = 0

Figure 14 - 32 Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) (4/4)

(4) Data to restart condition to address



Note 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μ s when specifying standard mode and at least 0.6 μ s when specifying fast mode.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark $n = 0$

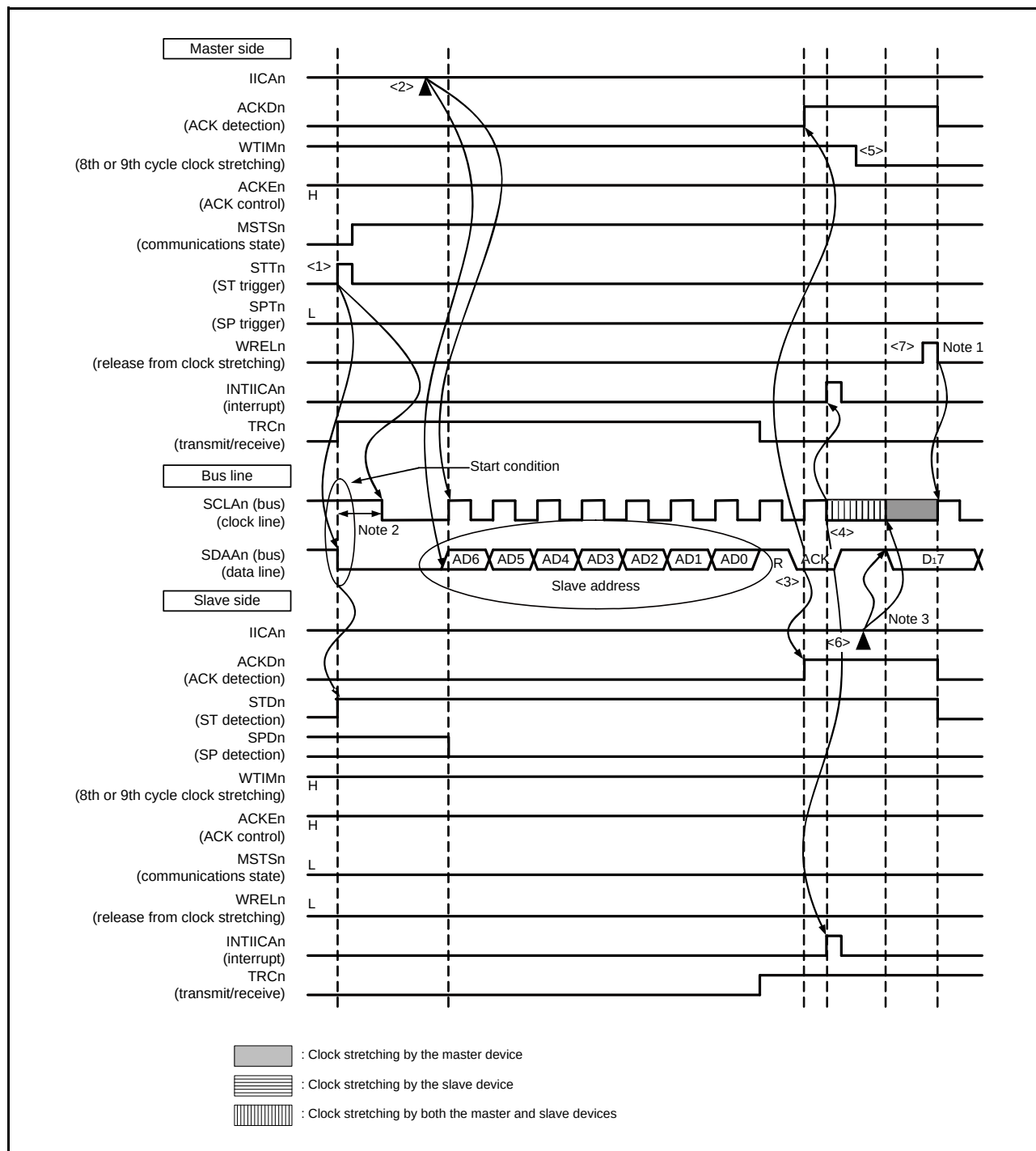
The following describes the operations in **Figure 14 - 32 (4) Data to restart condition to address**. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <3>, the data transmission step.

- <7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <i> The slave device reads the received data and releases the clock stretch state (WRELn = 1).
- <ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.
- <iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0

Figure 14 - 33 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (1/3)

(1) Start condition to address to data



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μ s when specifying standard mode and at least 0.6 μ s when specifying fast mode.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0

The meanings of <1> to <7> in **(1) Start condition to address to data** in **Figure 14 - 33** are explained below.

- <1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.
- <2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.
- <3> In the slave device if the address received matches the address (SVAn value) of a slave device^{Note}, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)^{Note}.
- <5> The timing at which the master device sets the clock stretch state changes to the 8th clock (WTIMn = 0).
- <6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch state set by the slave device.
- <7> The master device releases the clock stretch state (WRELn = 1) and starts transferring data from the slave device to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state.
The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in **Figure 14 - 33** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 33 (1) Start condition to address to data shows the processing from <1> to <7>,

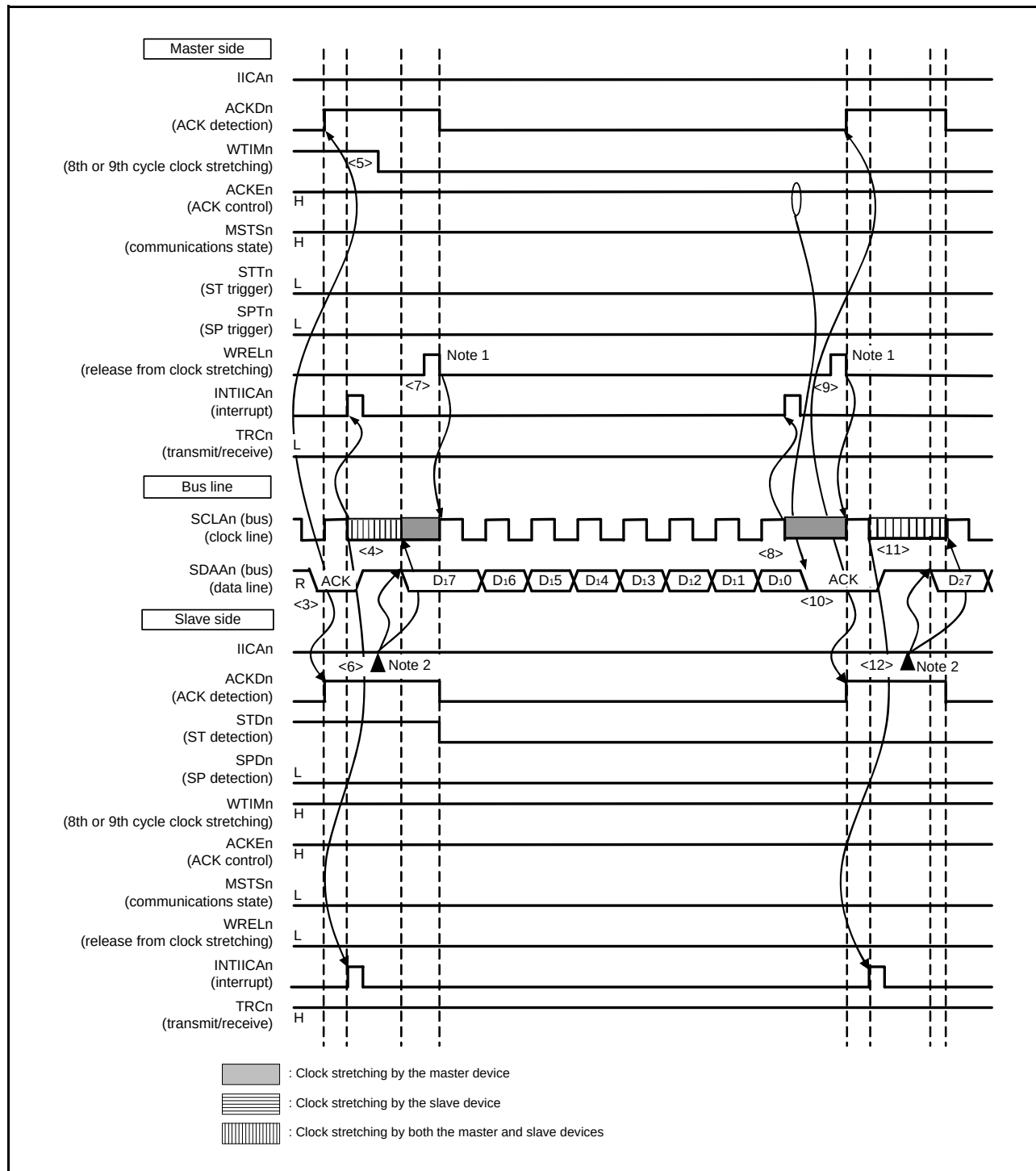
Figure 14 - 33 (2) Address to data to data shows the processing from <3> to <12>, and

Figure 14 - 33 (3) Data to data to stop condition shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 14 - 33 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (2/3)

(2) Address to data to data



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0

The meanings of <3> to <12> in **(2) Address to data to data** in **Figure 14 - 33** are explained below.

- <3> In the slave device if the address received matches the address (SVAn value) of a slave device^{Note}, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.
- <4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)^{Note}.
- <5> The master device changes the timing of clock stretching to the 8th clock (WTIMn = 0).
- <6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the slave device.
- <7> The master device releases the clock stretch state (WRELn = 1) and starts transferring data from the slave device to the master device.
- <8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 1 in the master device, the master device then sends an ACK by hardware to the slave device.
- <9> The master device reads the received data and releases the clock stretch state (WRELn = 1).
- <10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.
- <11> The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).
- <12> By the slave device writing the data to transmit to the IICAn register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state.
The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in **Figure 14 - 33** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 33 (1) Start condition to address to data shows the processing from <1> to <7>.

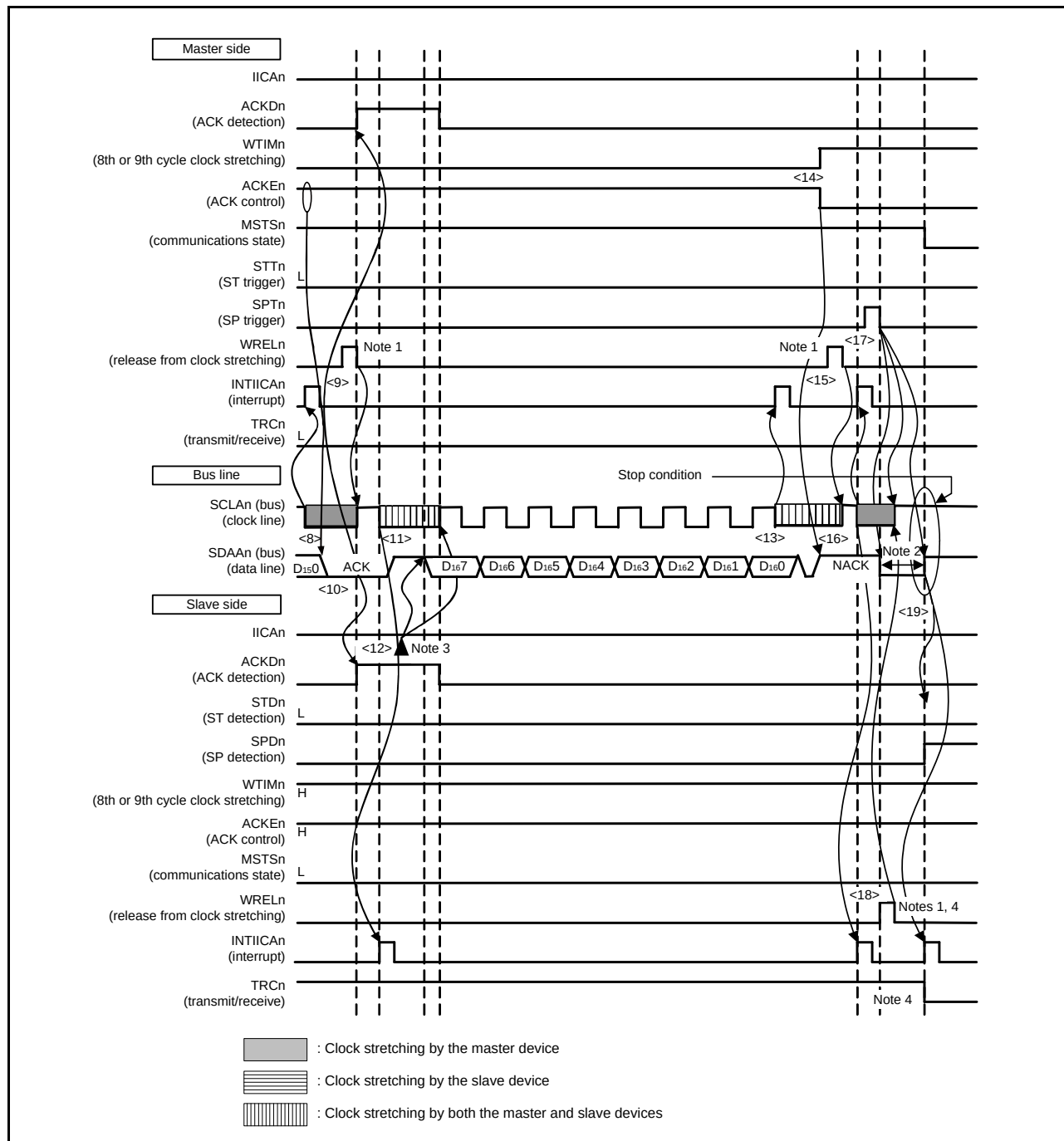
Figure 14 - 33 (2) Address to data to data shows the processing from <3> to <12>, and

Figure 14 - 33 (3) Data to data to stop condition shows the processing from <8> to <19>.

Remark 2. n = 0

Figure 14 - 33 Example of Slave to Master Communications (8th Cycle Clock Stretching Is Changed to 9th Cycle Clock Stretching for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) (3/3)

(3) Data to data to stop condition



- Note 1.** For release from the clock stretch state, write FFH to IICAn or set the WRELn bit.
- Note 2.** Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μ s when specifying standard mode and at least 0.6 μ s when specifying fast mode.
- Note 3.** For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.
- Note 4.** If the clock stretch state during transmission by a slave device is released by setting the WRELn bit, the TRCn bit will be cleared.

Remark n = 0

The meanings of <8> to <19> in **(3) Data to data to stop condition** in **Figure 14 - 33** are explained below.

- <8> The master device sets the clock stretch state ($SCLAn = 0$) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of $ACKEn = 0$ in the master device, the master device then sends an ACK by hardware to the slave device.
- <9> The master device reads the received data and releases the clock stretch state ($WRELn = 1$).
- <10> The ACK is detected by the slave device ($ACKDn = 1$) at the rising edge of the 9th clock.
- <11> The slave device sets the clock stretch state ($SCLAn = 0$) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).
- <12> By the slave device writing the data to transmit to the IICA register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.
- <13> The master device issues an interrupt (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets the clock stretch state ($SCLAn = 0$). Because ACK control ($ACKEn = 1$) is performed, the bus data line is at the low level ($SDAAn = 0$) at this stage.
- <14> The master device sets NACK as the response ($ACKEn = 0$) and changes the timing at which it sets the clock stretch state to the 9th clock ($WTIMn = 1$).
- <15> If the master device releases the clock stretch state ($WRELn = 1$), the slave device detects the NACK ($ACK = 0$) at the rising edge of the 9th clock.
- <16> The master device and slave device set the clock stretch state ($SCLAn = 0$) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).
- <17> When the master device issues a stop condition ($SPTn = 1$), the bus data line is cleared ($SDAAn = 0$) and the master device releases the clock stretch state. The master device then waits until the bus clock line is set ($SCLAn = 1$).
- <18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch state ($WRELn = 1$) to end communication. Once the slave device releases the clock stretch state, the bus clock line is set ($SCLAn = 1$).
- <19> Once the master device recognizes that the bus clock line is set ($SCLAn = 1$) and after the stop condition setup time has elapsed, the master device sets the bus data line ($SDAAn = 1$) and issues a stop condition (i.e. $SCLAn = 1$ and $SDAAn$ changes from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in **Figure 14 - 33** represent the entire procedure for communicating data using the I²C bus.

Figure 14 - 33 (1) Start condition to address to data shows the processing from <1> to <7>,

Figure 14 - 33 (2) Address to data to data shows the processing from <3> to <12>, and

Figure 14 - 33 (3) Data to data to stop condition shows the processing from <8> to <19>.

Remark 2. $n = 0$

Section 15 Serial Interface UARTA (UARTA)

The number of channels of the serial interface UARTA depends on the product.

	16-, 20-, 24-, 25-, 30-, and 32-pin products	36-, 40-, 44-, and 48-pin products
Number of channels	—	1

15.1 Overview

The serial interface UARTAn (n = 0) supports the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed; power consumption can be reduced.

(2) UART mode

This is a UART mode that supports continuous transmission. The list below outlines the features:

UARTAn performs an asynchronous communication. It has the following functions.

- Maximum transfer rate: 153.6 kbps
- Transmission and reception using two pins
TxDAn: Transmit data output pin
RxDAn: Receive data input pin
- Character length of transfer data selectable from 5, 7, and 8 bits
- Baud rate arbitrarily settable with the dedicated internal 8-bit baud rate generator
- Transmission and reception independent of each other (full-duplex communication)
- MSB or LSB first transfer selectable
- Inversion control of communication logic level provided
- Operating clock independent of the CPU/peripheral hardware clock selectable

Remark n: Unit number (n = 0)

Figure 15 - 1 shows a block diagram of UARTAn and Table 15 - 1 shows the pin configuration of UARTAn.

Figure 15 - 1 Block Diagram of UARTAn

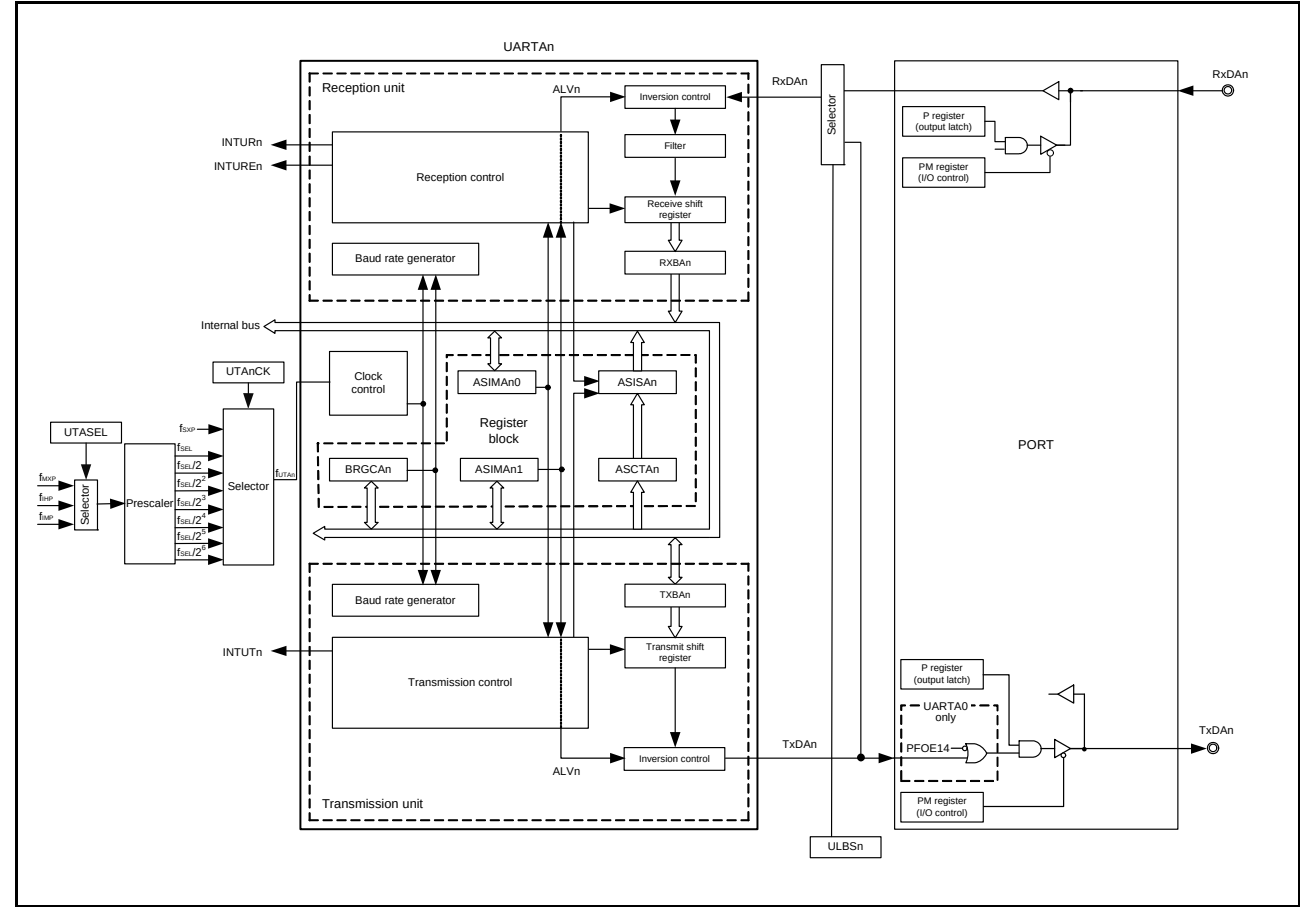


Table 15 - 1 UARTAn Pin Configuration (n = 0)

Name	I/O	Function
RxDA[n]	Input	Serial data input signal
TxDA[n]	Output	Serial data output signal

Remark f_{MXP} : High-speed peripheral clock frequency
 f_{HXP} : High-speed on-chip oscillator peripheral clock frequency
 f_{IMP} : Middle-speed on-chip oscillator peripheral clock frequency
 f_{SXP} : Low-speed peripheral clock frequency
 f_{SEL} : Selected clock to be divided for the UARTA
 f_{UTAN} : UARTAn operation clock

15.2 Registers for Controlling the Serial Interface UARTA

The following registers are used to control the serial interface UARTA.

- Peripheral enable register 1 (PER1)
- Transmit buffer register (TXBAn)
- Receive buffer register (RXBAn)
- Operation mode setting register 0 (ASIMAn0)
- Operation mode setting register 1 (ASIMAn1)
- Baud rate generator control register (BRGCAn)
- Status register (ASISAn)
- Status clear trigger register (ASCTAn)
- UARTA clock select register 0 (UTA0CK)
- UART loopback select register (ULBS)

Remark UARTAn uses the following registers, in addition to those listed above.

- Port mode register 7 (PM7) (See **Section 4 Port Functions.**)
- Port register 7 (P7) (See **Section 4 Port Functions.**)
- Pull-up resistor option register 7 (PU7) (See **Section 4 Port Functions.**)
- Port input mode register 7 (PIM7) (See **Section 4 Port Functions.**)
- Port output mode register 7 (POM7) (See **Section 4 Port Functions.**)
- Port mode control T register 7 (PMCT7) (See **Section 4 Port Functions.**)
- Port function output enable register 1 (PFOE1) (See **Section 4 Port Functions.**)

15.2.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

To use UARTA0, be sure to set bit 2 (UTAEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the PER1 register following a reset is 00H.

Figure 15 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>
PER1	0	SMSSEN	0	TML32EN	DTCEN	UTAEN	0	CTSUEN

UTAEN	Control of supply of an input clock to the serial interface UARTAn
0	Stops supply of an input clock. • The SFRs used by the serial interface UARTAn cannot be written. • When an SFR used by the serial interface UARTAn is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the serial interface UARTAn can be read and written.

Caution 1. Make sure that the setting of the UTAEN bit is 1 before using UARTAn. If UTAEN = 0, writing to the registers which control UARTAn is ignored.

Caution 2. The functions that are mounted depend on the product. For details on the PER1 register, see the description in Section 6 Clock Generator.

15.2.2 Transmit buffer register (TXBAn)

TXBAn is a buffer register for setting transmit data.

Transmission starts by writing data for transmission to the TXBAn register.

The TXBAn register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is FFH.

Figure 15 - 3 Format of Transmit Buffer Register (TXBAn)

Address: F0300H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TXBAn								

Bits 7 to 0	Function
—	When a character length of 8 bits is specified: - Data in bits 7 to 0 of TXBAn are transferred. When a character length of 7 bits is specified: - Data in bits 6 to 0 of TXBAn are transferred in either MSB- or LSB-first mode; bit 7 is invalid. When a character length of 5 bits is specified: - Data in bits 4 to 0 of TXBAn are transferred in either MSB- or LSB-first mode; bits 7 to 5 are invalid.

Caution 1. When the TXBFAn bit of the ASISAn register is 1, do not write data for transmission to the TXBAn register.

Caution 2. After setting the TXEAn bit of the ASIMAn0 register to 1, wait for the period of at least one cycle of the UARTAn operation clock (fUTAn) before setting the first data for transmission in the TXBAn register. If data for transmission is set within one cycle of the UARTAn operation clock after the TXEAn bit is set to 1, the start of transmission is delayed by one cycle of the UARTAn operation clock.

Remark Data is transferred from the TXBAn register to this register, and is then transmitted as serial data through the TxDAn pin. In the first transmission, data is transferred from the TXBAn register to this register immediately after data is written to the TXBAn register. In continuous transmission, data is transferred after transmission of one frame and just before generation of the transfer completion interrupt.
The transmit shift register cannot be manipulated directly by a program.

15.2.3 Receive buffer register (RXBAn)

The RXBAn register stores the parallel data converted by the receive shift register. Every time one byte of data is received, the next receive data is transferred from the receive shift register^{Note} to this register.

The RXBAn register can be read by an 8-bit memory manipulation instruction.

The value of this register following a reset is FFH.

Figure 15 - 4 Format of Receive Buffer Register (RXBAn)

Address: F0301H
After reset: FFH
R/W: R

Symbol	7	6	5	4	3	2	1	0
RXBAn								

Bits 7 to 0	Function
—	When a character length of 8 bits is specified: - Receive data is transferred to bits 7 to 0 of this register. When a character length of 7 bits is specified: - Receive data is transferred to bits 6 to 0 of this register in either MSB- or LSB-first mode; bit 7 is always 0. When a character length of 5 bits is specified: - Receive data is transferred to bits 4 to 0 of this register in either MSB- or LSB-first mode; bits 7 to 5 are always 0.

Note The receive shift register converts the serial data that is input through the RxDAn pin to parallel data. The receive shift register cannot be manipulated directly by a program.

Caution If an overrun error (OVEAn) occurs, the data received at that time are not stored in the RXBAn register.

15.2.4 Operation mode setting register 0 (ASIMAn0)

The ASIMAn0 register is an 8-bit register that controls serial communication of the serial interface UARTAn.

The ASIMAn0 register can be set by 1-bit or 8-bit memory manipulation instructions.

The value of this register following a reset is 01H.

Figure 15 - 5 Format of Operation Mode Setting Register 0 (ASIMAn0)

Address: F0302H
After reset: 01H
R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
ASIMAn0	UARTAENn	TXEAn	RXEAn	0	0	0	ISSMAAn	ISRMAAn
UARTAENn								
Note 1								
UART operation enable								
0								
Disables the UART operation clock. Resets the internal circuits. Note 2								
1								
Enables the UART operation clock.								
TXEAn								
Transmission enable								
0								
Disables transmission. (Resets the transmission circuit.)								
1								
Enables transmission.								
RXEAn								
Reception enable								
0								
Disables reception. (Resets the reception circuit.)								
1								
Enables reception.								
ISSMAAn								
Transmit interrupt mode select								
0								
The INTUTn interrupt is generated on completion of transmission.								
1								
The INTUTn interrupt is generated when the transmit buffer becomes empty. (for continuous transmission)								
ISRMAAn								
Receive interrupt mode select								
0								
The INTUREn interrupt is generated when a reception error occurs. (INTURn is not generated.)								
1								
The INTURn interrupt is generated when a reception error occurs. (INTUREn is not generated.)								

Note 1. When UARTAENn = 0, the level being output from the TxDAn pin and the level being input from the RxDAAn pin are determined according to the setting of the ALVn bit as described below.

- When ALVn = 0, output from the TxDAn pin is high.
- When ALVn = 1, output from the TxDAn pin is low.

Note 2. The ASISAn and RXBAAn registers are reset by clearing the UARTAENn bit to 0.

Caution 1. Be sure to clear bits 4, 3, and 2 to 0.

Caution 2. To start transmission, set the UARTAENn bit to 1 and then set the TXEAn bit to 1.

To stop transmission, clear the TXEAn bit to 0 and then clear the UARTAENn bit to 0.

Caution 3. To start reception, set the UARTAENn bit to 1 and then set the RXEAn bit to 1.

To stop reception, clear the RXEAn bit to 0 and then clear the UARTAENn bit to 0.

(Cautions continue on the next page.)

- Caution 4.** Follow the procedure below when setting the UARTAENn bit to 1 and then setting the RXEAn bit to 1.
- When ALVn = 0, the setting must be made while the level being input to the RxDAn pin is high. Otherwise, reception will start at that point and a framing error may occur.
 - When ALVn = 1, the setting must be made while the level being input to the RxDAn pin is low. Otherwise, reception will start at that point and a framing error may occur.
- Caution 5.** The TXEAn and RXEAn bits are synchronized with the UARTA_n operation clock (fUTAn). To enable transmission or reception again, set the TXEAn or RXEAn bit to 1 at least two cycles of the UARTA_n operation clock after clearing the TXEAn or RXEAn bit to 0. If the bit is set to 1 within two cycles of the UARTA_n operation clock after the clearing, the transmission or reception circuit may not be able to be initialized.
- Caution 6.** After setting TXEAn bit to 1, wait for at least one cycle of the UARTA_n operation clock (fUTAn) before setting the transmit data in the TXBAn register.
- Caution 7.** Clear the RXEAn bit to 0 before modifying the ISRMAN bit.

15.2.5 Operation mode setting register 1 (ASIMAn1)

The ASIMAn1 register is an 8-bit register that controls serial communication of the serial interface UARTAn.

The ASIMAn1 register can be set by 1-bit or 8-bit memory manipulation instructions.

The ASIMAn1 register must be modified while TXEAn = 0 and RXEAn = 0.

The value of this register following a reset is 1AH.

Figure 15 - 6 Format of Operation Mode Setting Register 1 (ASIMAn1)

Address: F0303H
After reset: 1AH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ASIMAn1	0	PSn1	PSn0	CLn1	CLn0	SLn	DIRn	ALVn

PSn1	PSn0	Transmission/reception parity bit setting 1, 0	
		Transmission	Reception
0	0	No parity bit is output.	Data is received without parity.
0	1	0 parity is output.	Data is received with 0 parity. Note
1	0	Odd parity is output.	Check is made for odd parity.
1	1	Even parity is output.	Check is made for even parity.

CLn1	CLn0	Transmission/reception character length setting 1, 0
0	0	Character length of data = 5 bits
0	1	Setting prohibited
1	0	Character length of data = 7 bits
1	1	Character length of data = 8 bits

SLn	Transmission stop bit length setting
0	Stop bit length = 1 bit
1	Stop bit length = 2 bits

DIRn	Transmission/reception order setting
0	MSB first
1	LSB first

ALVn	Transmission/reception level setting
0	Positive logic (wait state = high level, start bit = low level, stop bit = high level)
1	Negative logic (wait state = low level, start bit = high level, stop bit = low level)

Note When "Data is received with 0 parity" is set, parity check is not performed. Accordingly the PEAn bit of the ASISAn register is not set: no reception error interrupts are generated.

Caution 1. Clear both the TXEAn and RXEAn bits to 0 before modifying the ASIMAn1 register.

Caution 2. Reception is always handled as including a stop bit. The setting of the SLn bit does not affect reception.

15.2.6 Baud rate generator control register (BRGCAn)

The BRGCAn register sets the frequency divisor for the 8-bit counter in the serial interface UARTAn.

The BRGCAn register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is FFH.

Figure 15 - 7 Format of Baud Rate Generator Control Register (BRGCAn)

Address: F0304H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
BRGCAn								
Bits 7 to 0	Function							
—	Controls the UART baud rate (serial transfer speed). For details on the settings, see Table 15 - 2 .							

Caution Modify the BRGCAn bits while the TXEAn and RXEAn bits are 0 (in the transmission/reception stopped state).

Table 15 - 2 BRGCAn Settings

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	k	Selection of 8-bit counter output clock
0	0	0	0	0	0	0	X	X	Setting prohibited
0	0	0	0	0	0	1	0	2	$f_{UTAn}/2$
0	0	0	0	0	0	1	1	3	$f_{UTAn}/3$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UTAn}/252$
1	1	1	1	1	1	0	1	253	$f_{UTAn}/253$
1	1	1	1	1	1	1	0	254	$f_{UTAn}/254$
1	1	1	1	1	1	1	1	255	$f_{UTAn}/255$

Caution The baud rate is one half the frequency of the output clock signal from the 8-bit counter.

Remark 1. k: The value set with the BRGCAn register (k = 2, 3, 4, 5, 6, ..., 255)

Remark 2. X: Don't care.

For an example of the baud rate setting, see **15.3.4 (3) (c) Baud rate setting example**.

15.2.7 Status register (ASISAn)

The ASISAn register indicates the states in terms of errors and transmission on completion of reception by the serial interface UARTAn. It consists of three error flag bits (PEAn, FEAn, and OVEAn) and two transmission status flag bits (TXBFAn and TXSFAn).

The ASISAn register is read-only and can be read by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H. The PEAn, FEAn, and OVEAn bits are initialized by clearing the UARTAENn or RXEAn bit to 0. These bits are also cleared by writing to the corresponding bit of the ASCTAn register.

The TXBFAn and TXSFAn bits are initialized by clearing the UARTAENn or TXEAn bit to 0.

Figure 15 - 8 Format of Status Register (ASISAn) (1/2)

Address: F0305H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ASISAn	0	0	TXBFAn	TXSFAn	0	PEAn	FEAn	OVEAn
TXBFAn								
Transmit buffer data flag								
—								
[Setting condition]								
• Data is written to the TXBAn register. (Data exists in the TXBAn register.)								
[Clearing conditions]								
• The UARTAENn or TXEAn bit is cleared to 0.								
• Data is transferred to the transmit shift register.								
TXSFAn								
Transmit shift register data flag								
—								
[Setting condition]								
• Data is transferred from the TXBAn register. (Data is being transmitted.)								
[Clearing conditions]								
• The UARTAENn or TXEAn bit is cleared to 0.								
• Data is transferred from the transmit shift register and then no subsequent data is transferred from the TXBAn register.								
PEAn								
Parity error flag								
—								
[Setting condition]								
• The parity of the received data does not match the parity bit.								
[Clearing conditions]								
• The UARTAENn or RXEAn bit is cleared to 0.								
• 1 is written to the PECTAn bit.								
FEAn								
Framing error flag								
—								
[Setting condition]								
• A stop bit is not detected when receiving data.								
[Clearing conditions]								
• The UARTAENn or RXEAn bit is cleared to 0.								
• 1 is written to the FECTAn bit.								

Figure 15 - 8 Format of Status Register (ASISAn) (2/2)

OVEAn	Overflow error flag
—	[Setting condition] - The next reception is completed before the receive data in the RXBAn register is read. [Clearing conditions] - The UARTEAn or RXEAn bit is cleared to 0. 1 is written to the OVECTAn bit.

Caution 1. Be sure to clear bits 7, 6, and 3 to 0.

Caution 2. For continuous transmission, be sure to check that the TXBFAn flag is 0 after writing the first transmit data (the first byte) to the TXBAn register and then write the next transmit data (the second byte) to the TXBAn register. Otherwise, the transmit data become undefined.

However, the TXBFAn flag need not be checked when continuous transmission is performed by using the buffer empty interrupt (ISSMAAn bit = 1).

Caution 3. When initializing the transmission unit (TXEAn = 0) after completion of continuous transmission, be sure to check that the TXSFAn flag is 0 after the transfer completion interrupt is generated, and then initialize the unit. Otherwise, the transmit data become undefined.

Caution 4. The operation of the PEAn bit depends on the setting of the PSn1 and PSn0 bits of the ASIMAn1 register.

Caution 5. For the receive data, only the first 1 bit of the stop bits is checked regardless of the stop bit length.

Caution 6. When an overrun error occurs, the next receive data is not written to the RXBAn register and discarded.

15.2.8 Status clear trigger register (ASCTAn)

The ASCTAn register sets the trigger to clear the states in terms of errors on completion of reception of the serial interface UARTAn. It contains 3 bits of the error clear trigger flags (PECTAn, FECTAn, and OVECTAn).

The ASCTAn register can be written by an 8-bit or 1-bit memory manipulation instruction.

When the ASCTAn register is read, 00H is always read.

The value of this register following a reset is 00H. Writing 1 to the PECTAn, FECTAn, and OVECTAn bits clears the PEAAn, FEAn, and OVEAn bits of the ASISAn register, respectively. When writing 0, the corresponding error flags are not cleared.

Figure 15 - 9 Format of Status Clear Trigger Register (ASCTAn)

Address: F0306H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
ASCTAn	0	0	0	0	0	PECTAn	FECTAn	OVECTAn
PECTAn Note	Parity error flag clear trigger							
0	Does not clear the PEAAn flag. (The flag is retained.)							
1	Clears the PEAAn flag.							
FECTAn Note	Framing error flag clear trigger							
0	Does not clear the FEAn flag. (The flag is retained.)							
1	Clears the FEAn flag.							
OVECTAn Note	Overrun error flag clear trigger							
0	Does not clear the OVEAn flag. (The flag is retained.)							
1	Clears the OVEAn flag.							

Note When reading the ASCTAn register, 0 is returned.

Caution After writing 1 to the trigger bit, the corresponding error flag is cleared on the next rising edge of the operating clock (fUTAn). Accordingly, if reading the ASISAn register immediately after writing 1 to the trigger bit, the corresponding error flag may not have been cleared yet.

15.2.9 UARTA clock select register 0 (UTA0CK)

The UTA0CK register selects the operating clock of the UARTAn. The UTASEL1 and UTASEL0 bits select the clock source, fSEL, for UARTAn from fMXP, fIHP, and fIMP. The bits from UTA0CK3 to UTA0CK0 select the operating clock for UARTA0 from fSEL/1 to fSEL/64 and fSXP.

Figure 15 - 10 Format of UARTA0 Clock Select Register (UTA0CK)

Address: F0310H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
UTA0CK	0	0	UTASEL1	UTASEL0	UTA0CK3	UTA0CK2	UTA0CK1	UTA0CK0

UTASEL1	UTASEL0	fSEL clock select
0	0	Stop
0	1	fMXP
1	0	fIHP
1	1	fIMP

UTA0CK3	UTA0CK2	UTA0CK1	UTA0CK0	UARTA0 operation clock select (fUTA0)
0	0	0	0	fSEL
0	0	0	1	fSEL/2
0	0	1	0	fSEL/4
0	0	1	1	fSEL/8
0	1	0	0	fSEL/16
0	1	0	1	fSEL/32
0	1	1	0	fSEL/64
1	0	0	0	fSXP
Other than above				Setting prohibited

Caution This register should be read or written when the TXEAn and RXEAn bits are 0 (in the transmission/reception stopped state).

Remark fMXP: High-speed peripheral clock frequency
fIHP: High-speed on-chip oscillator peripheral clock frequency
fIMP: Middle-speed on-chip oscillator peripheral clock frequency
fSXP: Low-speed peripheral clock frequency
fSEL: Selected clock to be divided for the UARTA

15.2.10 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback function. This register has a bit to control loopback for each UART. Setting a bit to 1 enables the loopback function for the given UART, with the output from the transmit shift register being looped back to the receive buffer register.

The ULBS register can be set by 1-bit or 8-bit memory manipulation instructions. The value of the ULBS register following a reset is 00H.

Figure 15 - 11 Format of the UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	3	<2>	<1>	<0>
ULBS	0	0	0	ULBS4	0	ULBS2	ULBS1	ULBS0

ULBS4	Selection of the UART loopback function
0	The state of the RxDA0 pin of the serial interface UARTA0 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

Caution 1. Be sure to clear bits 7 to 5 and 3 to 0.

Caution 2. When using the loopback function in UARTA0, set the PFOE14 bit of the port function output enable register 1 (PFOE1) to 1.

Remark The RxDA0 pin can be used only in the products with 36 to 48 pins.

15.3 Operation

UARTAn operates in the following two modes.

- Operation stop mode
- UART mode

15.3.1 Operation stop mode

In the operation stop mode, serial communication is not performed, and thus the power consumption can be reduced. In addition, in this mode, the pins can be used as ordinary port pins. To set the operation stop mode, clear bits 7, 6, and 5 (UARTAENn, TXEAn, RXEAn) of the ASIMAn0 register to 0.

The bus clock is not stopped by the above setting. To completely stop operation, clear the UTAEN bit of the PER1 register to 0 after the above setting.

15.3.2 UART mode

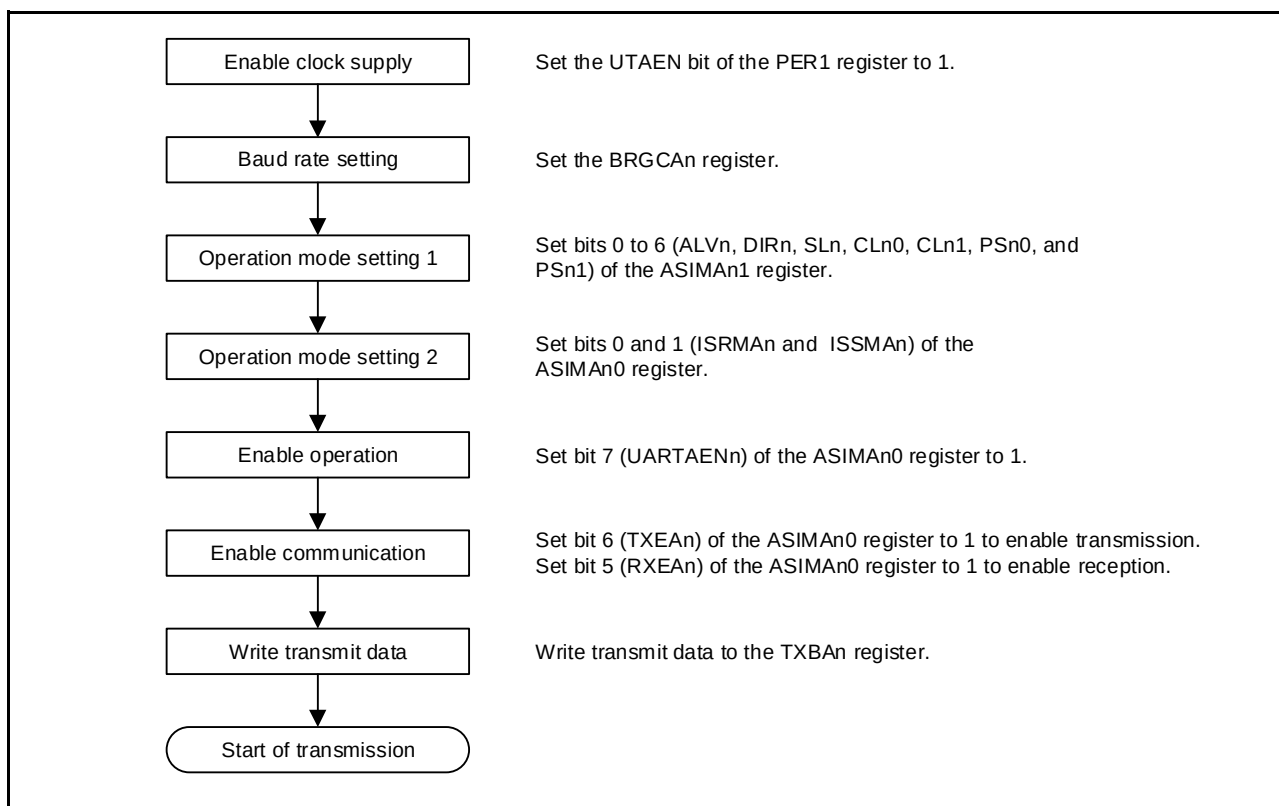
In this mode, one byte of data is transmitted and one byte is received following the start bit. That is, operation is full duplex.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Communication procedure

Figure 15 - 12 shows the flowchart of communication procedure.

Figure 15 - 12 Flowchart of Communication Procedure



Caution When using the receiving function, set the port pin allocated for reception to input mode by using the port mode registers. When using the transmitting function, set the port pin allocated for transmission to output mode by using the port mode registers, and set the respective bits in the port registers to 1.

Using a port pin with a multiplexed serial data output function (P72/TxDA0^{Note}) for serial data output requires setting the corresponding bit in the port mode register (PMxx) to 0 and the corresponding bit in the port register (Pxx) to 1. When using the port pin in N-ch open drain output (V_{DD} tolerance^{Note}) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Example: When P72/TxDA0 is to be used for serial data output^{Note}

Set the PM72 bit of port mode register 7 to 0.

Set the P72 bit of port register 7 to 1.

Using a port pin with a multiplexed serial data or serial clock input function (e.g. P71/RxDA0^{Note}) for serial data or serial clock input requires setting the corresponding bit in the port mode register (PMxx) to 1. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Example: When P71/RxDA0 is to be used for serial data input^{Note}

Set the PM71 bit of port mode register 7 to 1.

Set the P71 bit of port register 7 to 0 or 1.

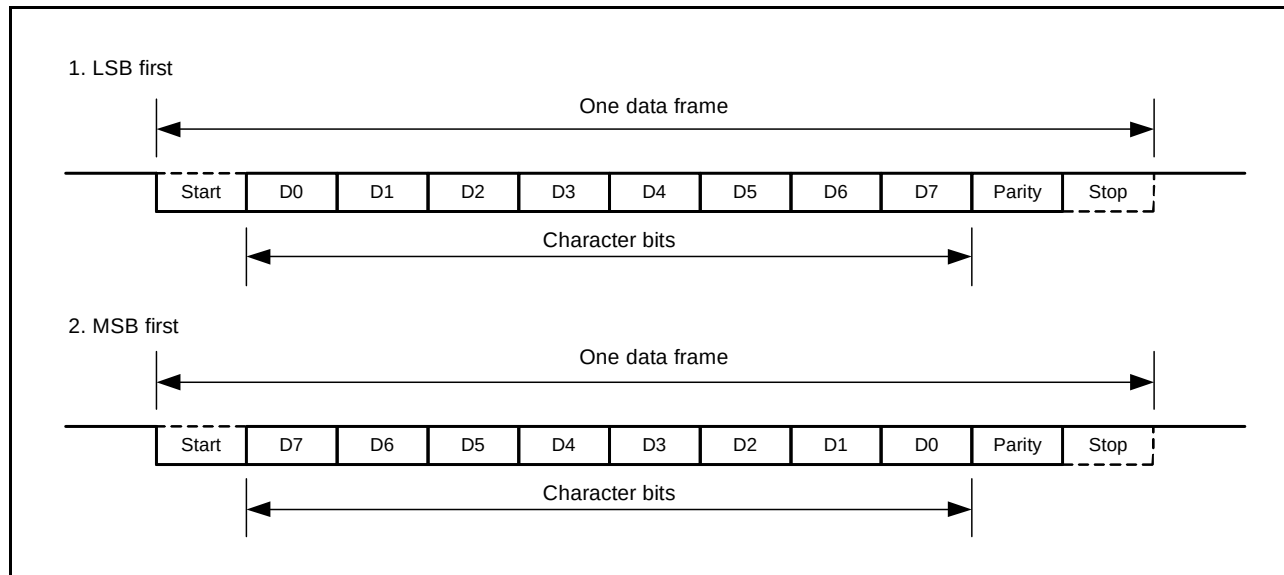
Note For products with 36 to 48 pins

(2) Format and waveform example of transmit/receive data

The following describes the communication data format of UARTAn.

Figure 15 - 13 shows the data format.

Figure 15 - 13 Transmit/Receive Data Format



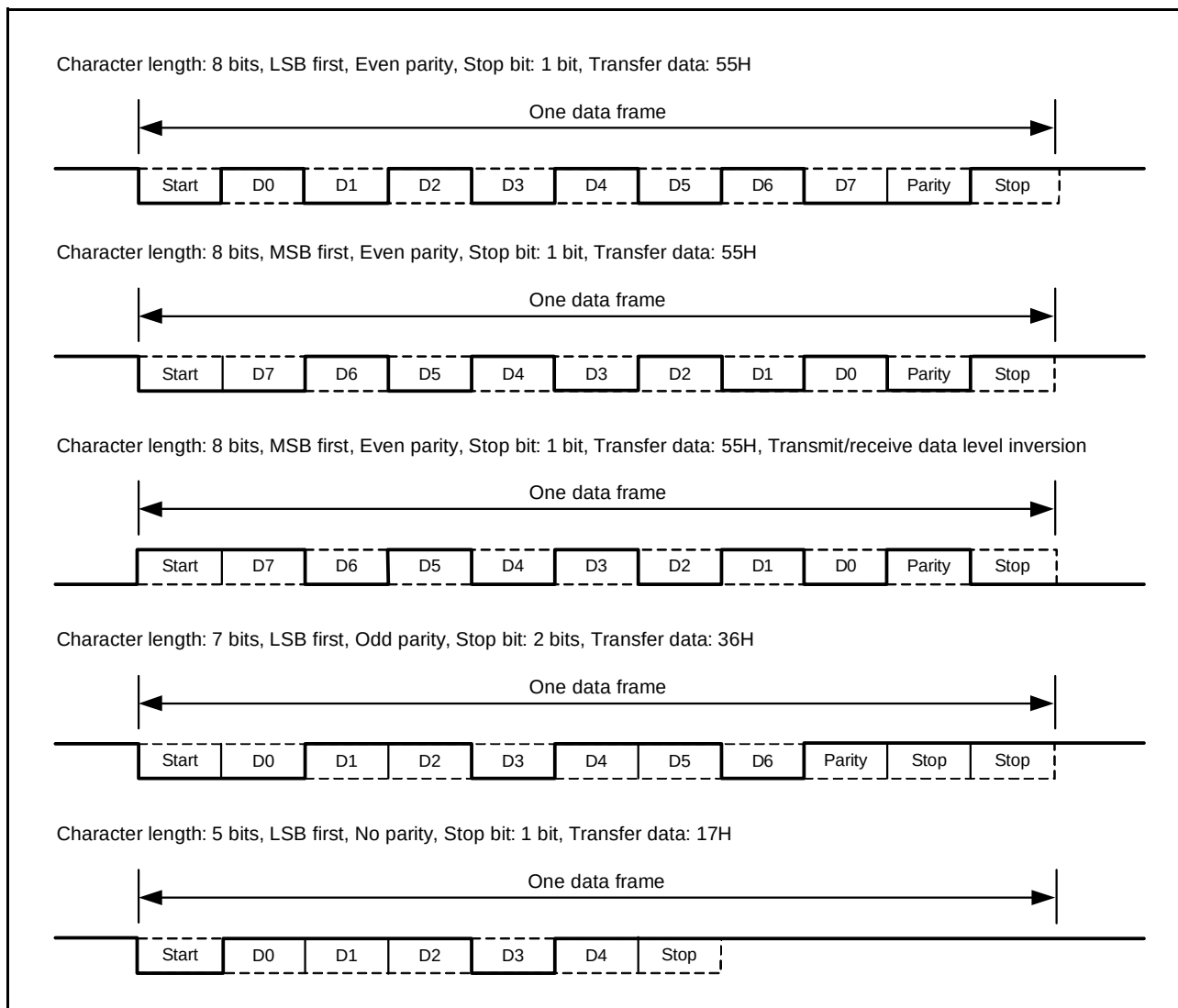
One data frame consists of the following bits.

- Start bit: 1 bit
- Character bits: 5, 7 or 8 bits
- Parity bit: Even parity, odd parity, 0 parity, or no parity
- Stop bit: 1 or 2 bits

The character bit length, the parity, the stop bit length, the transfer direction (LSB/MSB first), and the TxDAn pin output (direct/inverted) in one data frame are specified by the ASIMAn1 register.

Figure 15 - 14 shows the examples of transmit/receive data waveforms.

Figure 15 - 14 Example of Transmit/Receive Data Waveform



(3) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmitting and reception sides. With even/odd parity, a 1-bit (odd number) error can be detected. With zero/no parity, an error cannot be detected.

(a) Even parity

- In transmission

Data for transmission, including the parity bit, are controlled so that an even number of bits have the value 1. The value of the parity bit is set as follows.

If the data for transmission have an odd number of bits with the value 1: 1

If the data for transmission have an even number of bits with the value 1: 0

- In reception

In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is odd, a parity error occurs.

(b) Odd parity

- In transmission

Unlike even parity, data for transmission, including the parity bit, are controlled so that an odd number of bits have the value 1.

If the data for transmission have an odd number of bits with the value 1: 0

If the data for transmission have an even number of bits with the value 1: 1

- In reception

In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is even, a parity error occurs.

(c) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.

The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is appended to the transmit data.

Reception is performed assuming that there is no parity bit. A parity error does not occur, because there is no parity bit.

(4) Normal transmission

Transmission is enabled by setting bit 7 (UARTAENn) of the operation mode setting register 0 (ASIMAn0) to 1 and then setting bit 6 (TXEAn) of ASIMAn0 to 1. Transmission can be started by writing the data for transmission to the transmission buffer register (TXBAn). The start bit, parity bit, and stop bit are automatically appended to the data. When transmission is started, the data in the TXBAn register are transferred to the transmit shift register. After that, the transmit data are sequentially output from the transmit shift register to the TxDAn pin in the specified transfer direction. When transmission is completed, the parity and stop bits which are set by the ASIMAn0 register are appended and a transfer completion interrupt request signal (INTUTn) is generated.

Transmission is suspended until the next transmit data is written to the TXBAn register.

Figure 15 - 15 shows the timing of the transfer completion interrupt request signal (INTUTn). INTUTn is issued at the following timing.

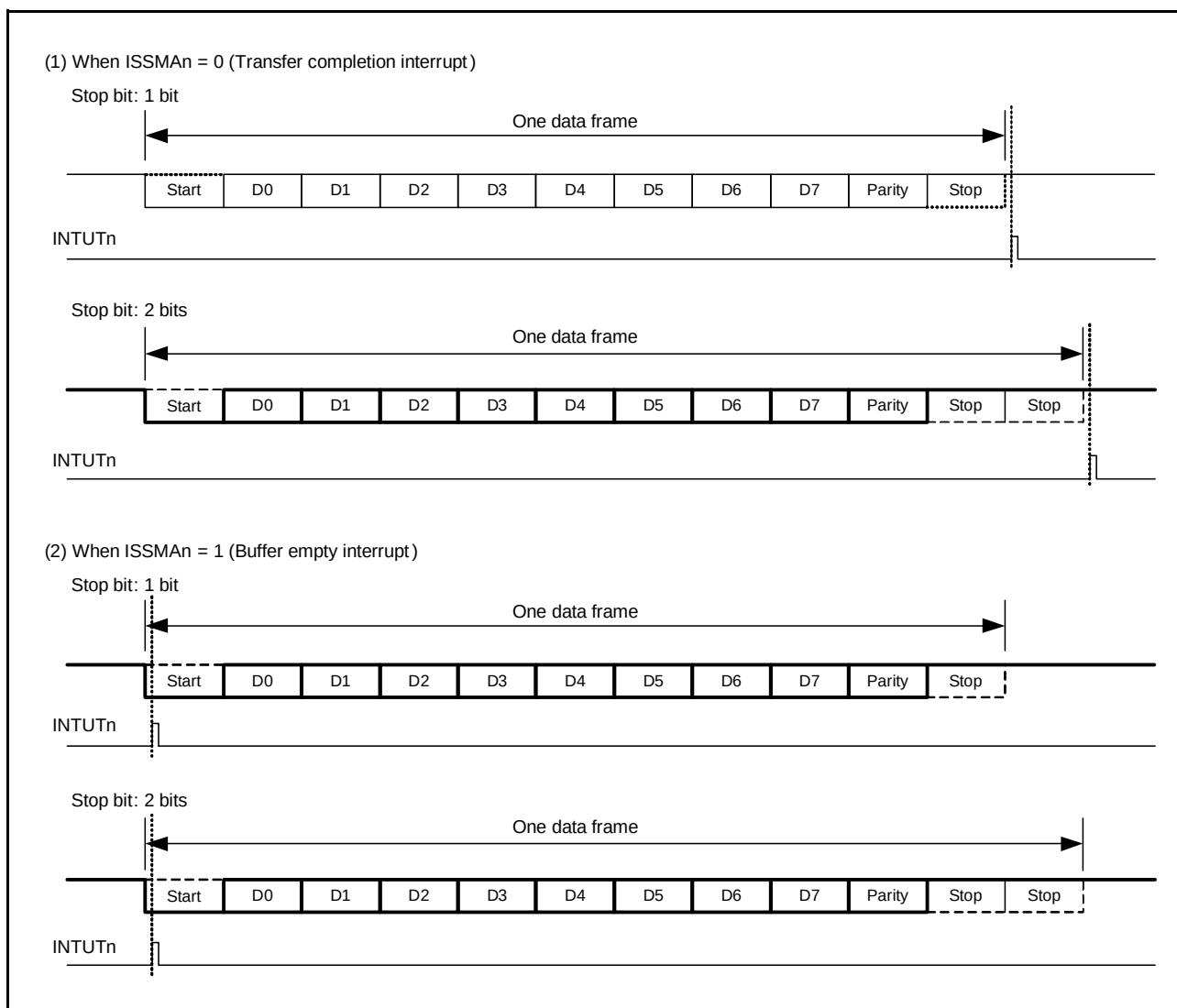
(a) When ISSMAAn = 0 (INTUTn functions as a transfer completion interrupt.)

INTUTn is issued after the output of the last stop bit.

(b) ISSMAAn = 1 (INTUTn functions as a buffer empty interrupt.)

INTUTn is issued when the start bit is output.

Figure 15 - 15 Interrupt Output Timing



(5) Continuous transmission

UARTAn has two separate registers for continuous transmission: the transmit buffer register (TXBAn) and the transmit shift register.

At the moment the transmit shift register starts a shift operation, the next transmit data can be written to the transmit buffer register (TXBAn). This operation enables continuous transmission, thereby improving communication rate.

Note that continuous transmission is not achieved when writing to the TXBAn register is not completed within the maximum number of clock cycles defined below from generation of the buffer empty interrupt.

Maximum number of clock cycles = Data transfer length $\times$ 2k - (2k + 3)

k: the value set with the BRGCAn bits (k = 2, 3, 4, 5, 6, ..., 255)

An example of calculating the maximum number of clock cycles is described below.

When the BRGCAn register = 02H (k = 2),

start bit = 1 bit, character length = 8 bits, parity used, and stop bit = 1 bit:

The maximum number of clock cycles = Transfer length $\times$ 2k - (2k + 3) = 11 $\times$ 2 $\times$ 2 - (2 $\times$ 2 + 3) = 37

(Writing must be completed within 37 cycles of the UARTAn operating clock (fUTAn).)

Continuous transmission is achieved by the following two methods.

(a) Continuous transmission by polling

Continuous transmission is achieved by polling the transmit buffer data flag (bit 5: TXBFAn) and the transmit shift register data flag (bit 4: TXSFAn) of the status register (ASISAn).

When using this method, clear bit 1 (ISSMAn) of the operation mode setting register 0 (ASIMAn0) to 0.

- At the start of and during continuous transmission

At the start of continuous transmission, write the first byte of data to the TXBAn register, check that the transmit buffer data flag (TXBFAn) is 0, and then write the second byte of data. In a similar way, check that the TXBFAn flag is 0 and then write the subsequent data to the TXBAn register.

TXBFAn	Determination flag indicating that writing to TXBAn is enabled or disabled at the start of continuous transmission
0	Writing is enabled.
1	Writing is disabled.

Caution To determine if continuous transmission is enabled or disabled, only check the TXBFAn flag. The TXSFAn flag must not be used for the determination in combination with this flag.

- Completion of continuous transmission

In continuous transmission, when data in the transmit shift register and the TXBAn register are transmitted after the required number of transmit data are written to the TXBAn register, the continuous transmission is completed. To confirm the completion, check the setting of the transmit shift register data flag (TXSFAn).

TXSFAn	Confirmation flag indicating whether transmission is in progress or not
0	Transmission is completed.
1	Transmission is in progress.

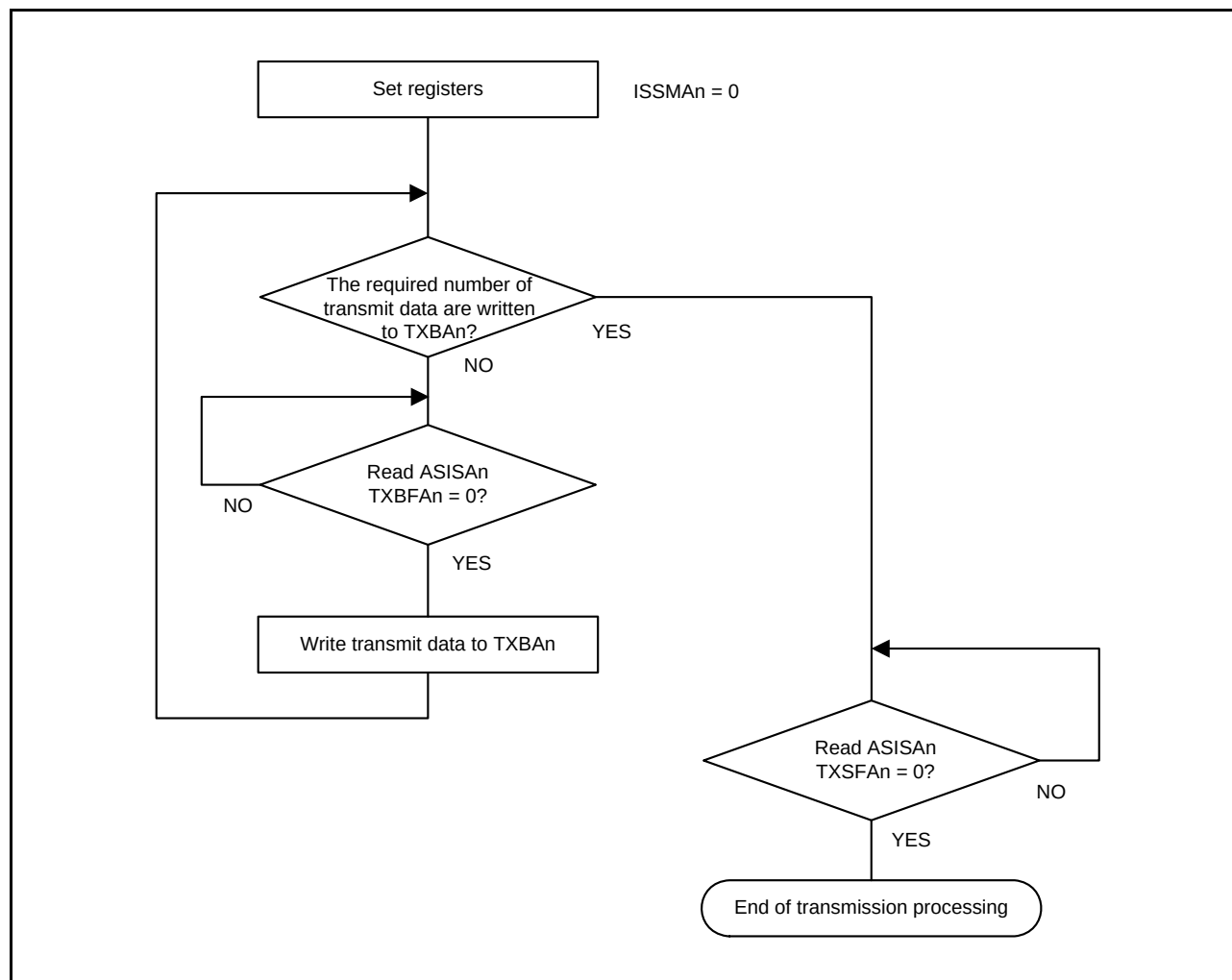
Caution 1. When initializing the transmission unit after completion of continuous transmission, check that the TXSFAn flag is 0 after the transfer completion interrupt is generated, and then initialize the unit.

Caution 2. During continuous transmission, after transmission of one data frame, the subsequent transmission may be completed before execution of the INTUTn interrupt processing.

This can be detected by incorporating the program that counts the number of transmit data and by referencing the TXSFAn flag.

Figure 15 - 16 shows a flow example of continuous transmission processing by polling.

Figure 15 - 16 Flow Example of Continuous Transmission Processing by Polling



(b) Continuous transfer by using an interrupt

Continuous transmission is achieved by using the interrupt (INTUTn).

An interrupt can be generated when data in the transmit buffer register (TXBAn) are transferred to the transmit shift register by setting bit 1 (ISSMA_n) to 1 in the operation mode setting register 0 (ASIMAn0).

With this setting, continuous transmission is enabled by writing data to the TXBAn register on occurrence of the buffer empty interrupt.

In addition, the transfer completion interrupt can be generated on completion of continuous transmission by clearing the ISSMA_n bit to 0 after writing the last transmit data to the TXBAn register.

Figure 15 - 17 shows a flow example of continuous transmission using interrupt.

Figure 15 - 17 Flow Example of Continuous Transmission Using Interrupt

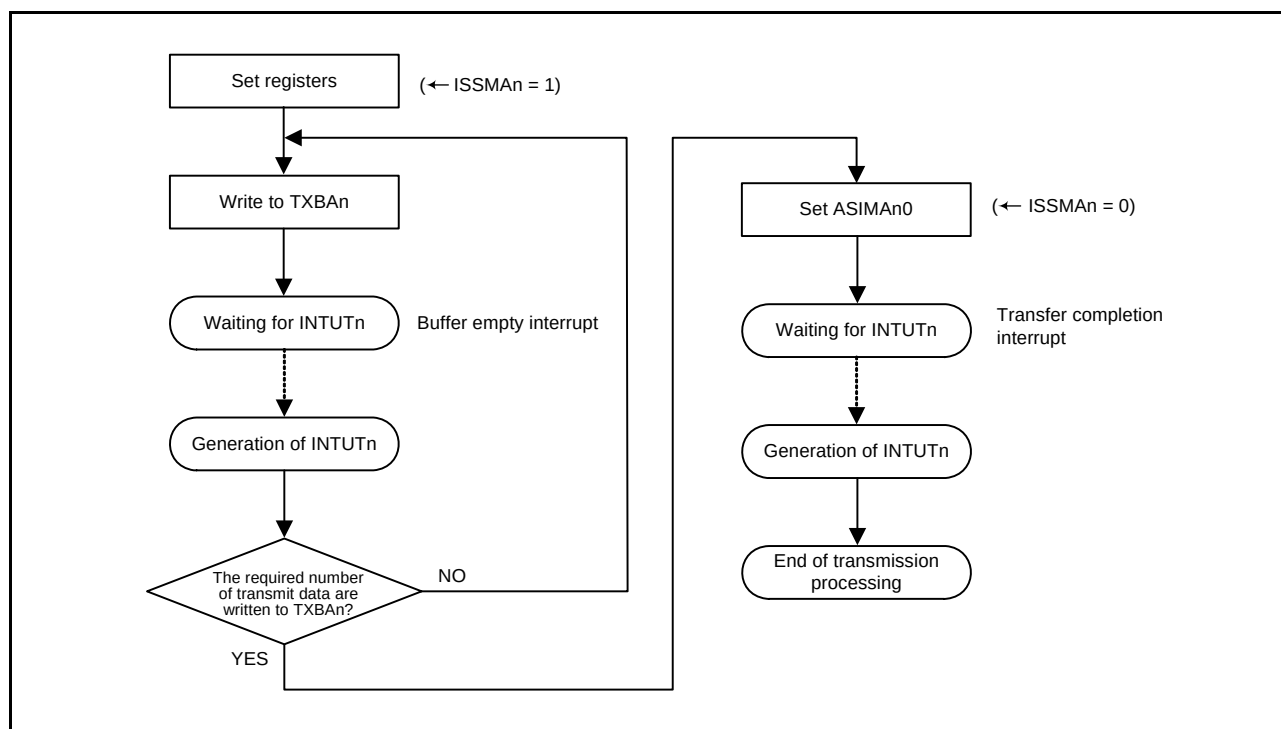
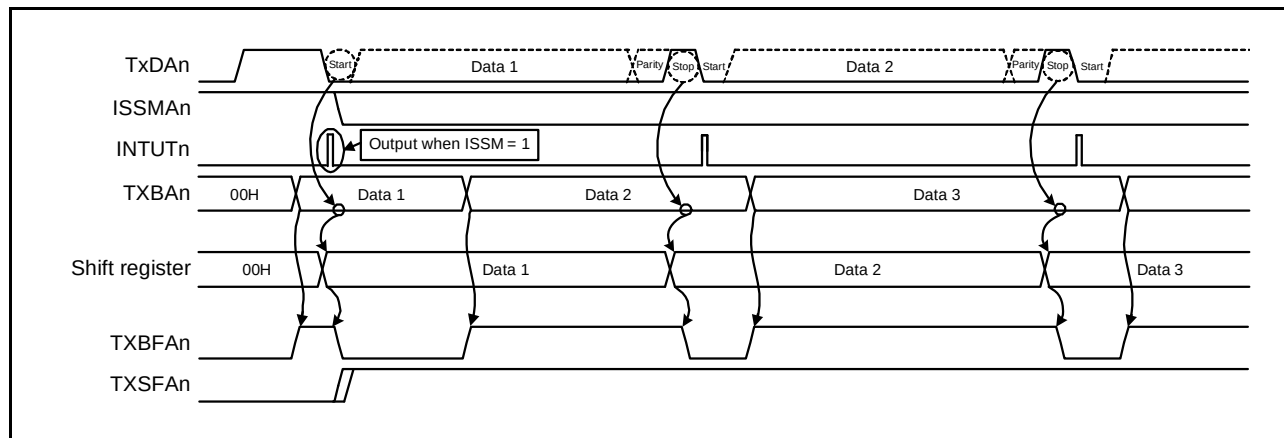


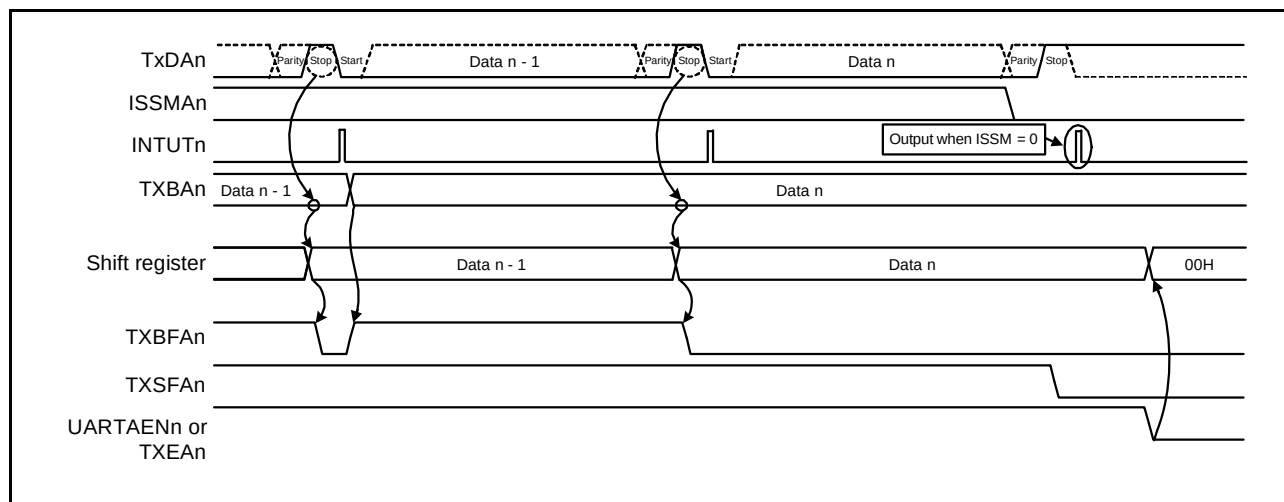
Figure 15 - 18 and **Figure 15 - 19** show the timing charts when the continuous transmission is started and completed, respectively.

Figure 15 - 18 Timing Chart When Continuous Transmission Is Started



Caution When the ASISAn register is read, both the TXBFAn and TXSFAn bits are read as 1 within this period.
Accordingly, use only the TXBFAn flag to determine if writing is enabled or disabled.

Figure 15 - 19 Timing Chart When Continuous Transmission Is Completed



(6) Normal reception

When setting bit 7 (UARTAENn) of the operation mode setting register 0 (ASIMAn0) to 1 and then setting bit 5 (RXEAn) of the ASIMAn0 register to 1, reception is enabled, and sampling of the input to the RxDAn pin is performed.

When the ALVn bit is 0, the 8-bit counter of the baud rate generator starts counting on detection of the falling edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 15 - 20**). If the RxDAn pin is low, it is regarded as a start bit.

When the ALVn bit is 1, the 8-bit counter of the baud rate generator starts counting on detection of the rising edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 15 - 20**). If the RxDAn pin is high, it is regarded as a start bit.

Figure 15 - 20 shows the timing chart of receive operation.

On detection of a start bit, receive operation is started: serial data is sequentially stored in the receive shift register at a specified baud rate. On reception of a stop bit, the transfer completion interrupt (INTURn) is generated, and at the same time, the data in the receive shift register is written to the receive buffer register (RXBAn).

Note that when an overrun error (OVEAn) occurs, the data received on occurrence of the error is not written to the RXBAn register.

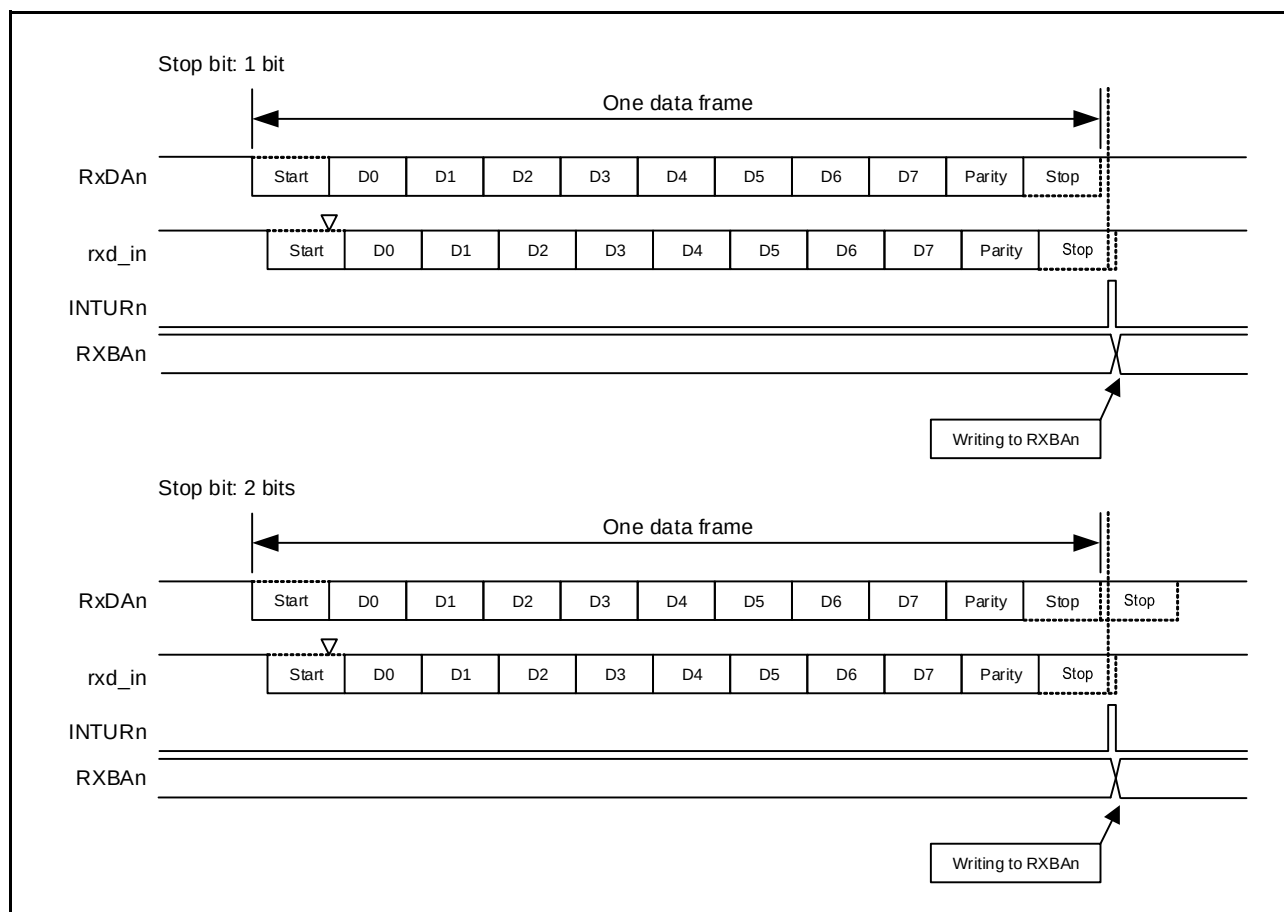
When a parity error (PEAn) or a framing error (FEAn) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (INTURn/INTUREn) set in the ISRMAn bit is generated.

When a reception error occurs, read the status register (ASISAn) and then read the receive buffer register (RXBAn) to clear the error flag.

If the receive buffer register (RXBAn) is not read, an overrun error will occur when the next data is received: the reception error state will continue.

Reception is always handled as including a stop bit. Accordingly, the second stop bit is ignored.

Figure 15 - 20 Timing of UART Receive Operation



Remark 1. rxd_in: The internal signal generated by latching RxDA[n] with a noise filter
(rxd_in is delayed relative to RxDA[n] by maximum of 3 cycles of the UART operation clock.)

Remark 2. The INTUR output timing in the figure is just an example.
The timing relative to RxDA[n] varies according to the setting of the BRGCA[n] register.

(7) Reception error

Three types of errors may occur during reception; parity error, framing error, and overrun error.

When these errors occur, the corresponding error flag in the status register (ASISAn) is set, and the reception error interrupt request signal (INTURn or INTUREn) is generated.

The type of the reception error can be identified by the reception error interrupt processing routine, which reads and checks the contents of the status register (ASISAn).

The contents of the ASISAn register is cleared to 0 by setting the corresponding bit of the status clear trigger register (ASCTAn) to 1.

Table 15 - 3 shows the causes of the reception errors.

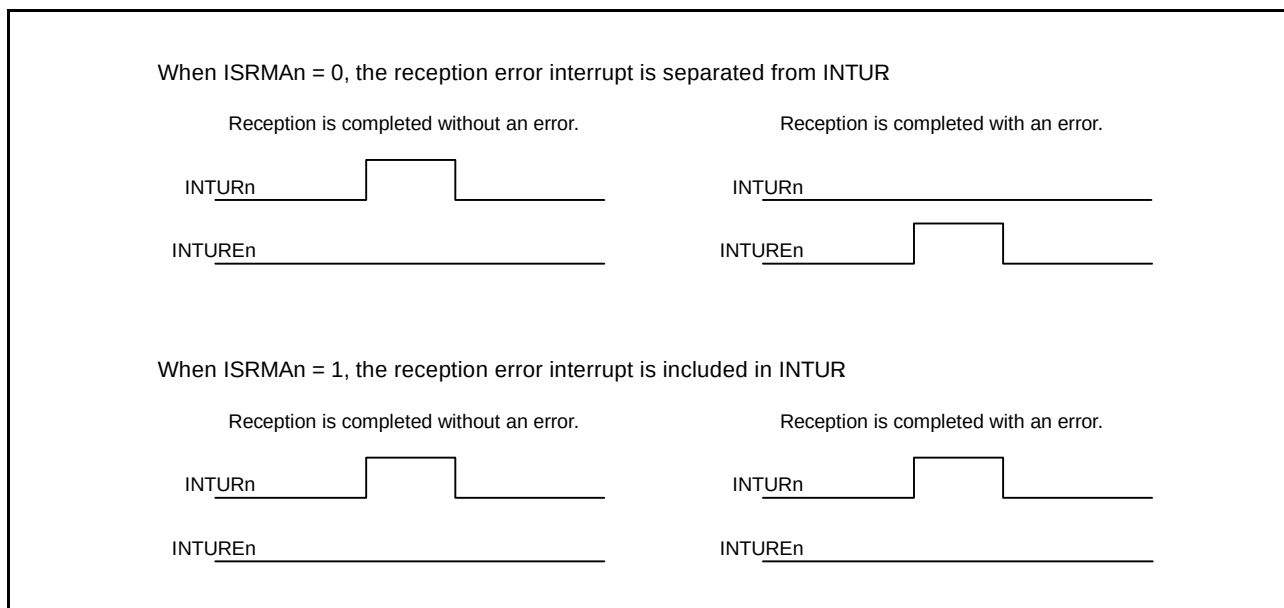
Table 15 - 3 Causes of Reception Errors

Error flag	Reception error	Cause
PEAn	Parity error	The parity specified for reception does not match the parity of receive data.
FEAn	Framing error	No stop bit is detected.
OVEAn	Overrun error	Before the receive data is read from the receive buffer, the next data reception is completed.

Setting bit 0 (ISRMAN) of the operation mode setting register 0 (ASIMAn0) to 0 allows the reception error interrupt to be separated from INTURn and allows it to be generated as INTUREn.

Figure 15 - 21 shows the interrupt output waveform which varies depending on the setting of the ISRMAN bit.

Figure 15 - 21 Various Interrupt Output Waveforms Depending on ISRMAN Setting

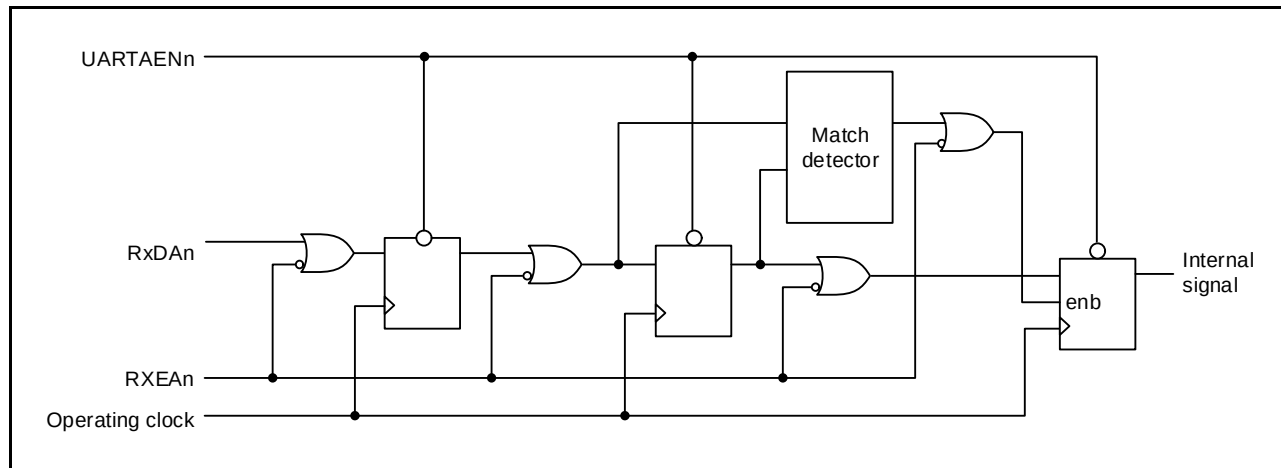


15.3.3 Receive data noise filter

This filter samples the receive data (RxDAn), and determines the level when the same level is sampled twice. The receive data is delayed by maximum of three cycles of the operating clock because of the circuit configuration.

Figure 15 - 22 shows the noise filter circuit.

Figure 15 - 22 Noise Filter



Caution 1. When ALVn = 0 (wait state = high level; start bit = low level), the initial value of the receive data (RxDAn) needs to be “high”.

Caution 2. When ALVn = 1 (wait state = low level; start bit = high level), the initial value of the receive data (RxDAn) needs to be “low”.

Remark n: Unit number (n = 0)

15.3.4 Baud rate generator

The baud rate generator consists of 8-bit programmable counters, and generates a serial clock for transmission/reception of UARTAn.

An 8-bit counter is provided each for transmission and reception.

(1) Configuration of baud rate generator

(a) UARTAn operation clock

When bit 7 (UARTAENn) = 1 in the operation mode setting register 0 (ASIMAn0), the UARTAn operation clock (f_{UTAn}) is supplied to each module. When UARTAENn = 0, the UARTAn operation clock is fixed to low level.

(b) Transmission counter

This counter is cleared to 0 and stops when bit 7 (UARTAENn) = 0 or bit 6 (TXEAn) = 0 in the operation mode setting register 0 (ASIMAn0). It starts counting when UARTAENn = 1 and TXEAn = 1.

The counter is cleared to 0 when the first transmit data is written to the transmit buffer register (TXBAn).

When continuous transmission is performed, the counter is cleared to 0 again when transmission of one frame of data has been completed. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until the UARTAENn or TXEAn bit is cleared to 0. When UARTAENn = 0 or TXEAn = 0 in the ASIMAn0 register, the counter stops at 00H.

(c) Reception counter

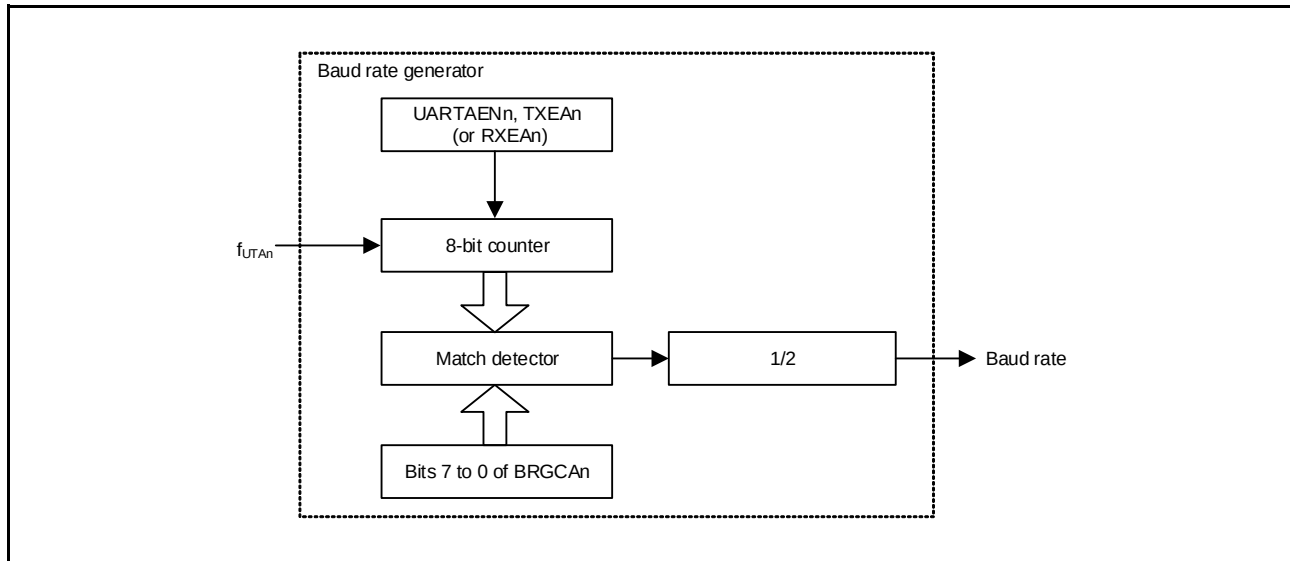
This counter is cleared to 0 and stops when bit 7 (UARTAENn) = 0 or bit 5 (RXEAn) = 0 in the operation mode setting register 0 (ASIMAn0). It starts counting when the start bit is detected.

The counter stops operation after one frame has been received, until the next start bit is detected. When UARTAENn = 0 or RXEAn = 0 in the ASIMAn0 register, the counter stops at 00H.

Remark n: Unit number (n = 0)

Figure 15 - 23 shows the configuration of the baud rate generator.

Figure 15 - 23 Configuration of Baud Rate Generator



(2) Generation of serial clock

A serial clock to be generated can be specified by using the baud rate generator control register (BRGCAn).

The baud rate generator divides the frequency of the input clock signal to the 8-bit counter (f_{UTAn}) by the divisor set by the BRGCAn register. The result of this division is further divided by 2 to produce the serial clock.

(3) Baud rate calculation

(a) Baud rate calculation expression

The baud rate can be calculated by the following expression.

$$\text{Baud rate} = f_{UTAn} \div (2 \times k) \text{ [bps]}$$

f_{UTAn} : Frequency of operating clock

k: Value set by bits 7 to 0 of the BRGCAn register ($k = 2, 3, 4, \dots, 255$)

(b) Baud rate error

The baud rate error can be calculated by the following expression.

$$\text{Error} = \left[\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right] \times 100 \text{ [\%]}$$

Caution 1. Keep the baud rate error during transmission to within the permissible error range on the reception side.

Caution 2. Make sure that the baud rate error during reception satisfies the permissible baud rate error range during reception. Permissible baud rate error during reception is described in 15.3.4 (2) (d) Permissible baud rate range during reception.

(c) Baud rate setting example

Table 15 - 4 Set Data of Baud Rate Generator (1/4)

Desired baud rate	In operation with f _{HFP} = 32 MHz (UTASEL1 and UTASEL0 = 10B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
4800 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
9600 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
19200 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
38400 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	
76800 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
115200 bps	139	-0.08%	69	0.64%	35	-0.79%	17	2.12%	Disabled		Disabled		Disabled	
153600 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCAn)
(k = 2, 3, 4, ..., 255)
n: Unit number (n = 0)

Table 15 - 4 Set Data of Baud Rate Generator (2/4)

Desired baud rate	In operation with f _{HFP} = 4 MHz (UTASEL1 and UTASEL0 = 11B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		156	0.16%
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
600 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
1200 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
2400 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
4800 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	
9600 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
19200 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	
38400 bps	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled	
76800 bps	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled	
115200 bps	17	2.12%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
153600 bps	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCAn)
(k = 2, 3, 4, ..., 255)
n: Unit number (n = 0)

Table 15 - 4 Set Data of Baud Rate Generator (3/4)

Desired baud rate	In operation with f _{MXP} = 20 MHz (UTASEL1 and UTASEL0 = 01B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%
4800 bps	Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%
9600 bps	Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%
19200 bps	Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%
38400 bps	255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%
76800 bps	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled	
115200 bps	87	-0.22%	43	0.94%	22	-1.36%	11	-1.36%	Disabled		Disabled		Disabled	
153600 bps	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCAn)
(k = 2, 3, 4, ..., 255)
n: Unit number (n = 0)

Table 15 - 4 Set Data of Baud Rate Generator (4/4)

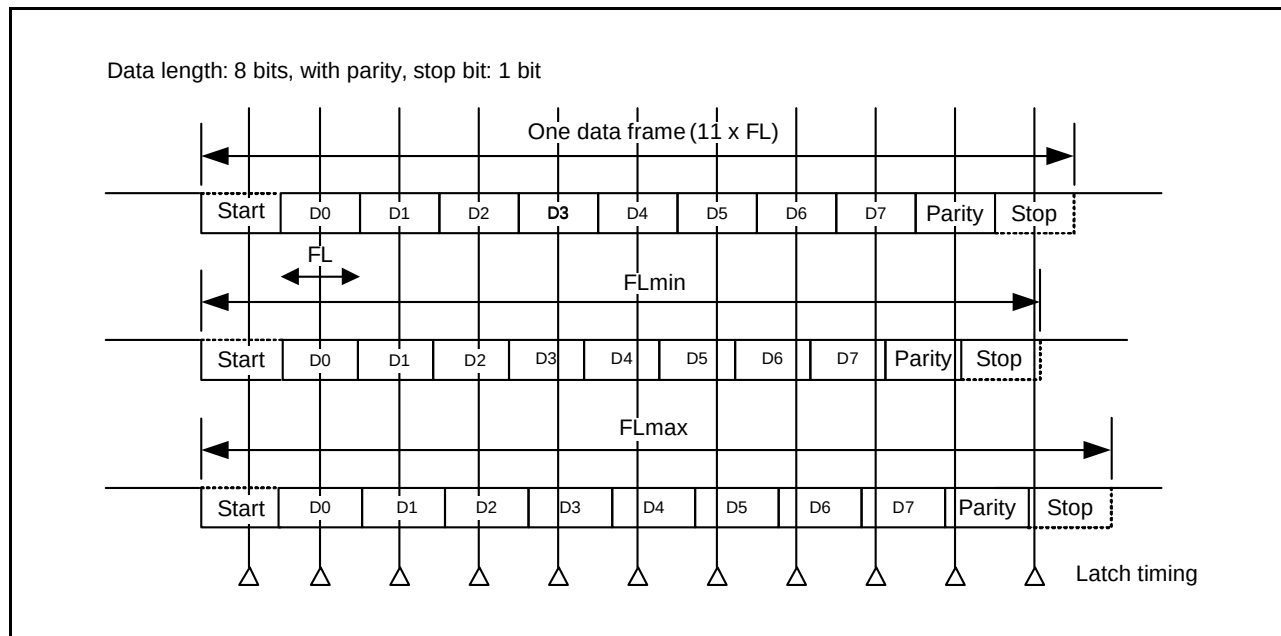
Desired baud rate	In operation with $f_{SXP} = 32.768 \text{ kHz}$ (UTAnCK3 to UTAnCK0 = 1000B)	
	k	Error from the desired baud rate
200 bps	82	-0.10%
300 bps	55	-0.70%
600 bps	27	-1.14%
1200 bps	14	-2.48%
2400 bps	7	-2.48%
4800 bps	Disabled	
9600 bps	Disabled	
19200 bps	Disabled	
38400 bps	Disabled	
76800 bps	Disabled	
115200 bps	Disabled	
153600 bps	Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCAn)
(k = 2, 3, 4, ..., 255)
n: Unit number (n = 0)

- (d) Permissible baud rate range during reception

Figure 15 - 24 shows the permissible error from the baud rate on the transmitting side during reception.

Figure 15 - 24 Permissible Baud Rate Range during Reception



Caution Be sure to make settings so that the baud rate error during reception is within the permissible error range. Use the calculation expression below to check if the error is within the permissible range.

After the start bit is detected, the latch timing of receive data is determined by the counter specified with the baud rate generator control register (BRGCAn). If the whole frame including the stop bit has been received before this latching, reception can proceed correctly.

Assuming that 11 bits of data are received, the theoretical values can be calculated as follows.

- The relation between 1-bit data length and baud rate

$$FL = (Brate) - 1$$

Brate: Baud rate of UART

k: Set value of BRGCAn

FL: 1-bit data length

Margin of latch timing: 1 clock

- Minimum permissible data frame length (FLmin)

$$FLmin = 11 \times FL - \frac{k-1}{2k} \times FL = \frac{21k+1}{2k} FL$$

- Maximum permissible baud rate for reception on the transmitting side (BRmax)

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+1} Brate$$

- Maximum permissible data frame length (FLmax)

$$FLmax = \frac{21k+1}{20k} FL \times 11$$

- Minimum permissible baud rate for reception on the transmitting side (BRmin)

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-1} Brate$$

Table 15 - 5 shows the permissible baud rate error between UART and the transmitting side can be calculated from the above minimum and maximum baud rate expressions.

Table 15 - 5 Maximum/Minimum Permissible Baud Rate Error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.32%	-2.43%
4	+3.52%	-3.61%
8	+4.14%	-4.19%
20	+4.51%	-4.53%
50	+4.66%	-4.67%
100	+4.71%	-4.71%
255	+4.74%	-4.74%

Remark 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k).
The higher the input clock frequency and the division ratio (k), the higher the permissible error.

Remark 2. k: Set value of BRGCAn
n: Unit number (n = 0)

15.4 Points for Caution when the Serial Interface UARTA is to be Used

15.4.1 Port setting for RxDAn pin

When $ALV_n = 0$ (wait state = high level, start bit = low level), the initial value of receive data (RxDAn) must be high.

When $ALV_n = 1$ (wait state = low level, start bit = high level), the initial value of receive data (RxDAn) must be low.

Accordingly, port setting is required for the RxDAn pin before setting $UARTAEN_n = 1$.

15.4.2 Point for caution when selecting the UARTAn operation clock (fUTAn)

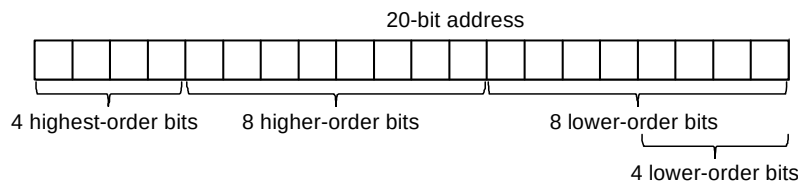
When the middle-speed on-chip oscillator peripheral clock (fMPP) is selected for fUTAn, communication may not be executed correctly due to the oscillation frequency accuracy of the middle-speed on-chip oscillator. Adjust the accuracy, therefore, by using the middle-speed on-chip oscillator trimming register (MIOTRM).

When the low-speed peripheral clock (fsXP) is selected for fUTAn and the low-speed on-chip oscillator peripheral clock (fIL) is selected for fsXP, communication may not be executed correctly due to the oscillation frequency accuracy of the low-speed on-chip oscillator. Adjust the accuracy, therefore, by using the low-speed on-chip oscillator trimming register (LIOTRM).

Remark n: Unit number (n = 0)

Section 16 Data Transfer Controller (DTC)

The term “8 higher-order bits of the address” in this section indicates bits 15 to 8 of 20-bit address as shown below.



Unless otherwise specified, the 4 highest-order address bits all become 1 (values are of the form FxxxxH).

16.1 Functions of DTC

The data transfer controller (DTC) transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt and transfers data. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 16 - 1 lists the DTC specifications.

Table 16 - 1 DTC Specifications (1/2)

Item		Specification
Activation sources		14 sources (16-pin products)/18 sources (20-pin products)/21 sources (24- and 25-pin products)/28 sources (30- and 32-pin products)/30 sources (36-pin products)/31 sources (40- and 44-pin products)/32 sources (48-pin products)
Allocatable control data		24 sets
Address space available for use with DTC transfer	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Source	Special function registers (SFRs), RAM area (excluding general-purpose registers), mirror area ^{Note} , data flash memory area ^{Note} , extended special function registers (2nd SFRs)
	Destination	Special function registers (SFRs), RAM area (excluding general-purpose registers), extended special function registers (2nd SFRs)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		Refer to Table 16 - 3 DTC Activation Sources and Vector Addresses .

Table 16 - 1 DTC Specifications (2/2)

Item		Specification
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

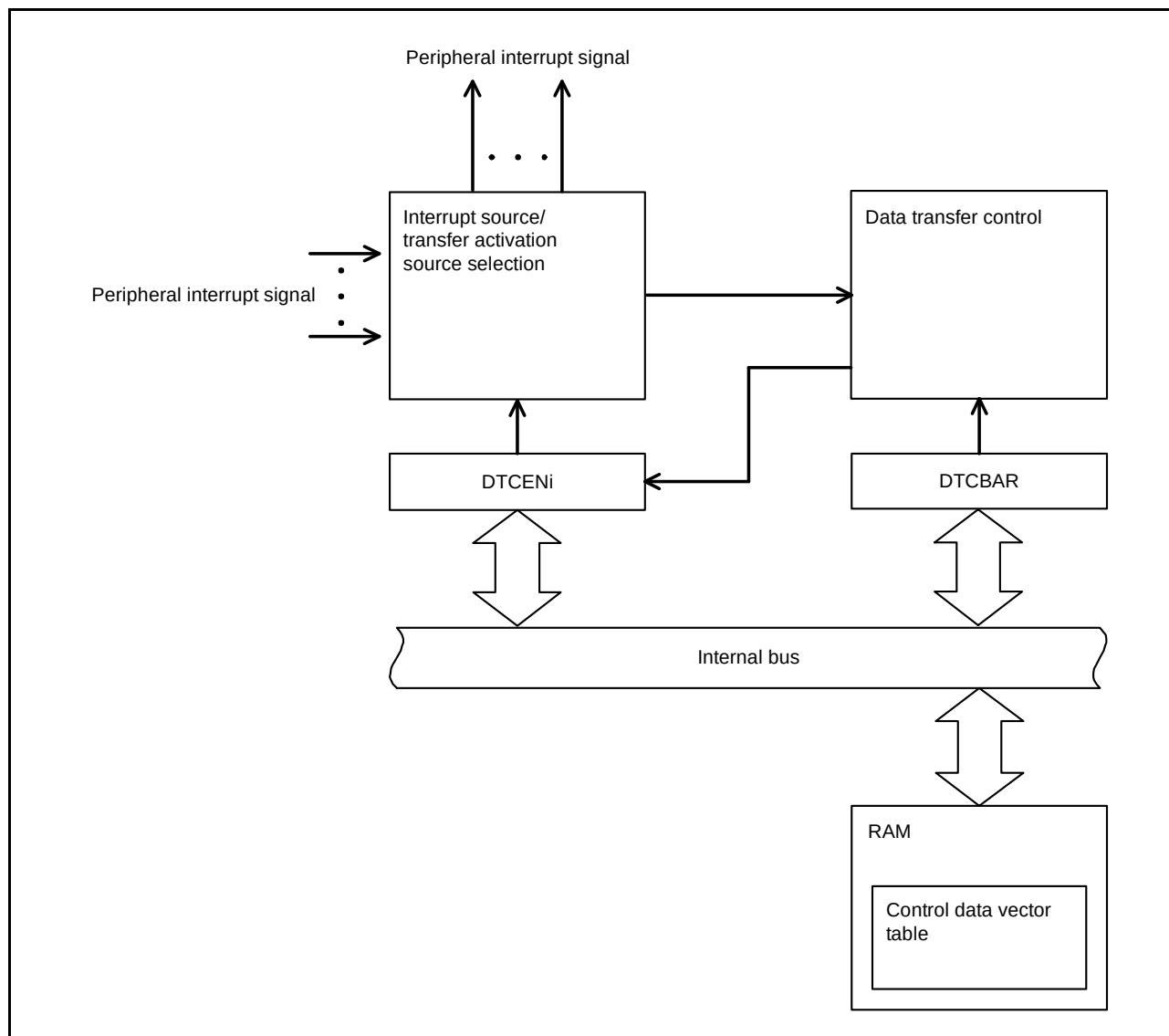
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 4, j = 0 to 23

16.2 Configuration of DTC

Figure 16 - 1 shows the DTC block diagram.

Figure 16 - 1 DTC Block Diagram



Remark i = 0 to 4

16.3 Registers for Controlling the DTC

The following registers are used to control the DTC.

- Peripheral enable register 1 (PER1)
- DTC activation enable register i (DTCENi) (i = 0 to 4)
- DTC base address register (DTCBAR)

The DTC control data are listed below.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

- DTC control register j (DTCCRj) (j = 0 to 23)
- DTC block size register j (DTBLSj) (j = 0 to 23)
- DTC transfer count register j (DTCCTj) (j = 0 to 23)
- DTC transfer count reload register j (DTRLdj) (j = 0 to 23)
- DTC source address register j (DTSARj) (j = 0 to 23)
- DTC destination address register j (DTDARj) (j = 0 to 23)

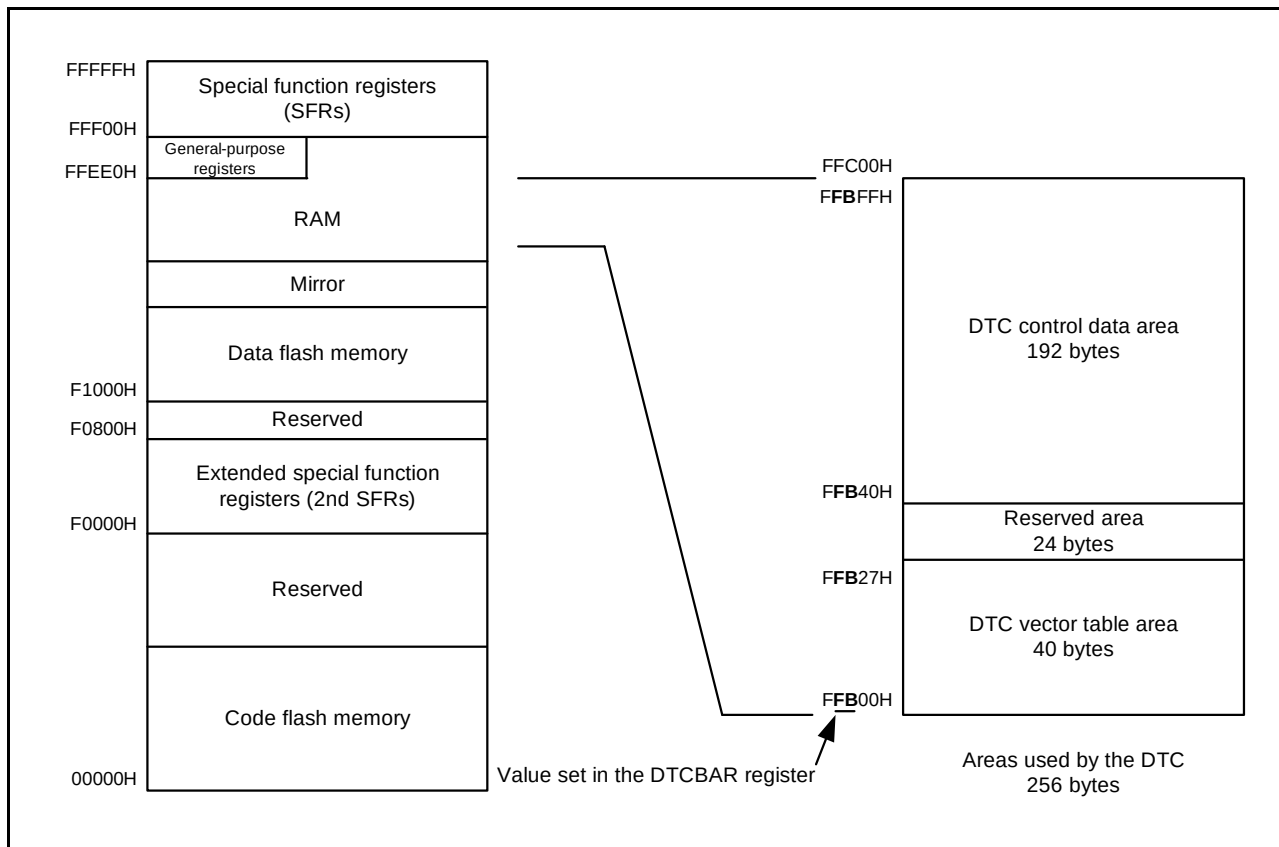
16.3.1 Allocation of DTC control data area and DTC vector table area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table are allocated within the RAM area.

Figure 16 - 2 shows a memory map example when the DTCBAR register is set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 16 - 2 Memory Map Example when the DTCBAR Register is Set to FBH



Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

16.3.2 Control data allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).

The 8 higher-order bits for start addresses 0 to 23 are set by the DTCBAR register, and the 8 lower-order bits are separately set according to the vector table assigned to each activation source.

Figure 16 - 3 shows control data allocation.

Caution 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (activation disabled).

Caution 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 16 - 3 Control Data Allocation

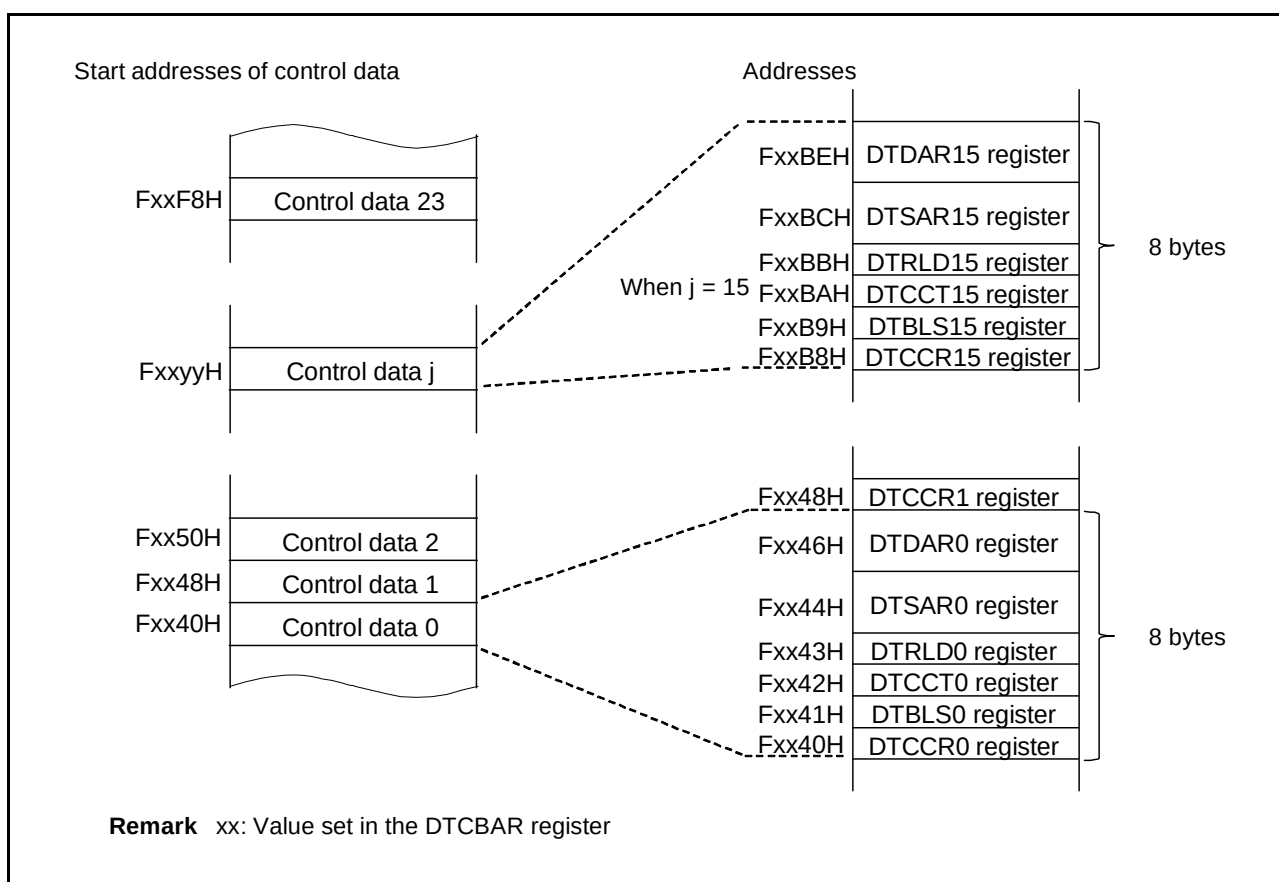


Table 16 - 2 Start Addresses of Control Data

j	Address
11	Fxx98H
10	Fxx90H
9	Fxx88H
8	Fxx80H
7	Fxx78H
6	Fxx70H
5	Fxx68H
4	Fxx60H
3	Fxx58H
2	Fxx50H
1	Fxx48H
0	Fxx40H

j	Address
23	FxxF8H
22	FxxF0H
21	FxxE8H
20	FxxE0H
19	FxxD8H
18	FxxD0H
17	FxxC8H
16	FxxC0H
15	FxxB8H
14	FxxB0H
13	FxxA8H
12	FxxA0H

Remark xx: Value set in the DTCBAR register

16.3.3 Vector table

When the DTC is activated, one control data is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data is read from the DTC control data area.

Table 16 - 3 lists the DTC activation sources and vector addresses. A one byte of the vector table is assigned to each activation source, and data from 40H to F8H is stored in each area to select one of the 24 control data sets. The 8 higher-order bits for the vector address are set by the DTCBAR register, and 00H to 27H are allocated to the 8 lower-order bits corresponding to the activation source.

Caution Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register is set to 0 (activation disabled).

Figure 16 - 4 Start Addresses of Control Data and Vector Table

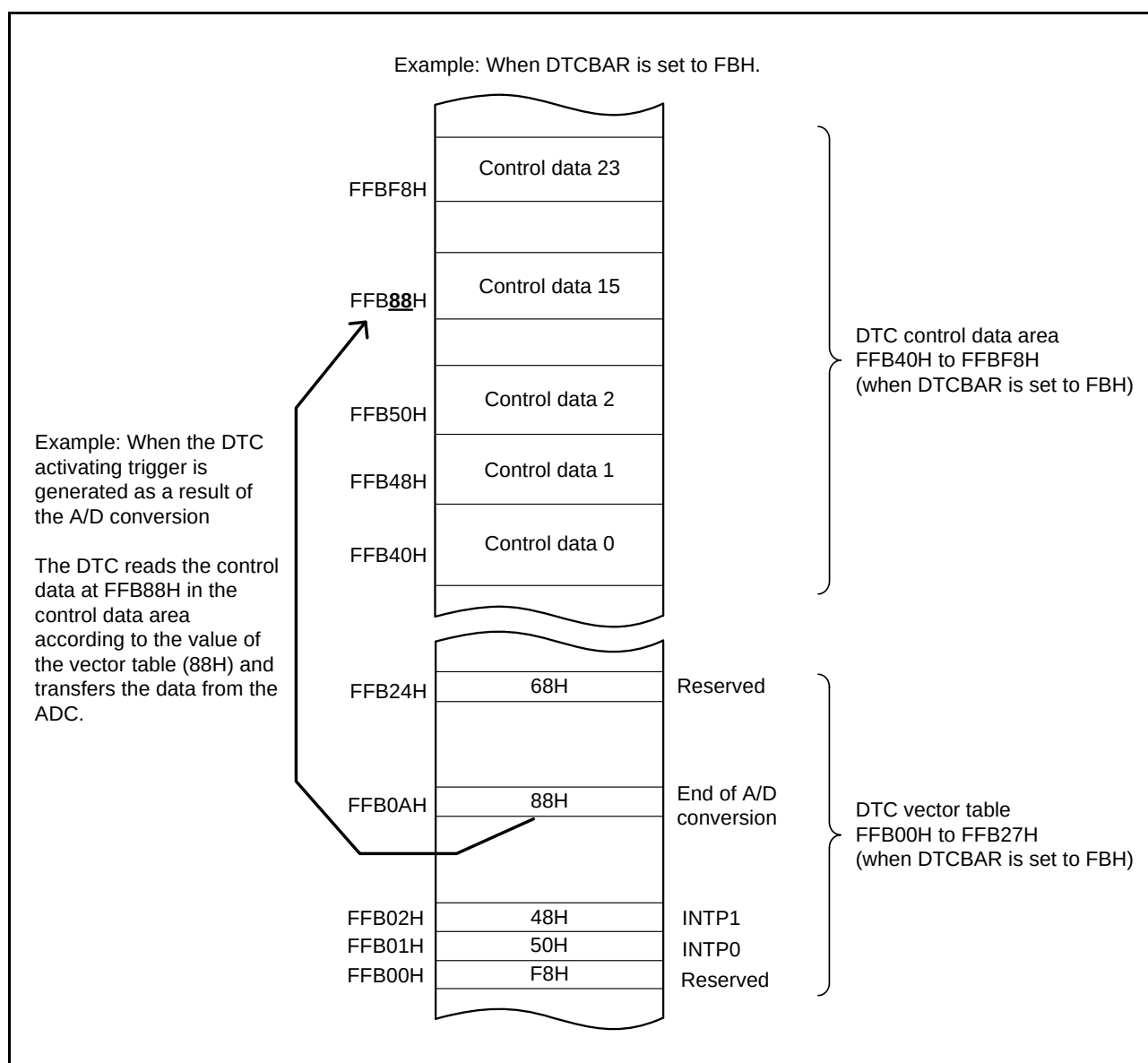


Table 16 - 3 DTC Activation Sources and Vector Addresses

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in the DTCBAR register + 00H	 Highest Lowest
INTP0	1	Address set in the DTCBAR register + 01H	
INTP1 ^{Note 3}	2	Address set in the DTCBAR register + 02H	
INTP2 ^{Note 4}	3	Address set in the DTCBAR register + 03H	
INTP3	4	Address set in the DTCBAR register + 04H	
INTP4 ^{Note 3}	5	Address set in the DTCBAR register + 05H	
INTP5 ^{Note 2}	6	Address set in the DTCBAR register + 06H	
INTP6 ^{Note 7}	7	Address set in the DTCBAR register + 07H	
Reserved	8	Address set in the DTCBAR register + 08H	
Key input ^{Note 6}	9	Address set in the DTCBAR register + 09H	
A/D conversion end	10	Address set in the DTCBAR register + 0AH	
UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	11	Address set in the DTCBAR register + 0BH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	12	Address set in the DTCBAR register + 0CH	
UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	13	Address set in the DTCBAR register + 0DH	
UART1 transmission transfer end ^{Note 2}	14	Address set in the DTCBAR register + 0EH	
UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end ^{Note 4}	15	Address set in the DTCBAR register + 0FH	
UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end ^{Note 4}	16	Address set in the DTCBAR register + 10H	
Reserved	17	Address set in the DTCBAR register + 11H	
Reserved	18	Address set in the DTCBAR register + 12H	
UARTA0 reception transfer end ^{Note 5}	19	Address set in the DTCBAR register + 13H	
UARTA0 transmission transfer end/buffer empty ^{Note 5}	20	Address set in the DTCBAR register + 14H	
Reserved	21	Address set in the DTCBAR register + 15H	
Reserved	22	Address set in the DTCBAR register + 16H	
End of counting or capturing by channel 0 of timer array unit 0 ^{Note 2}	23	Address set in the DTCBAR register + 17H	
End of counting or capturing by channel 1 of timer array unit 0 ^{Note 2}	24	Address set in the DTCBAR register + 18H	
End of counting or capturing by channel 2 of timer array unit 0	25	Address set in the DTCBAR register + 19H	
End of counting or capturing by channel 3 of timer array unit 0 ^{Note 3}	26	Address set in the DTCBAR register + 1AH	
End of counting or capturing by channel 4 of timer array unit 0 ^{Note 4}	27	Address set in the DTCBAR register + 1BH	
End of counting or capturing by channel 5 of timer array unit 0 ^{Note 4}	28	Address set in the DTCBAR register + 1CH	
End of counting or capturing by channel 6 of timer array unit 0 ^{Note 4}	29	Address set in the DTCBAR register + 1DH	
End of counting or capturing by channel 7 of timer array unit 0 ^{Note 4}	30	Address set in the DTCBAR register + 1EH	
Fixed-cycle signal of realtime clock/alarm match detection	31	Address set in the DTCBAR register + 1FH	
Interval signal detection of 32-bit interval timer	32	Address set in the DTCBAR register + 20H	
Request to write to a configuration register of an individual channel of the capacitive sensing unit (INTCTSUSR)	33	Address set in the DTCBAR register + 21H	
Request to transfer data measured by the capacitive sensing unit (INTCTSURD)	34	Address set in the DTCBAR register + 22H	
Reserved	35	Address set in the DTCBAR register + 23H	
Reserved	36	Address set in the DTCBAR register + 24H	
Event output from the event link controller	37	Address set in the DTCBAR register + 25H	
Event output from the SNOOZE mode sequencer	38	Address set in the DTCBAR register + 26H	
Voltage detection ^{Note 1}	39	Address set in the DTCBAR register + 27H	

Note 1. When bit 6 (LVD0SEL) of the option byte (000C1H) is set to 0 or when bit 7 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Note 2. This is applicable to the 20- to 48-pin products.

Note 3. This is applicable to the 24- to 48-pin products.

Note 4. This is applicable to the 30- to 48-pin products.

Note 5. This is applicable to the 36- to 48-pin products.

Note 6. This is applicable to the 40- to 48-pin products.

Note 7. This is applicable to the 48-pin products.

16.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the DTC is to be used, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 16 - 5 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH

After reset: 00H

R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>
PER1	0	SMSSEN	0	TML32EN	DTCEN	UTAEN	0	CTSUN
DTCEN		Control of supply of an input clock to the DTC						
0		Stops supply of an input clock. • The SFRs used by the DTC cannot be written.						
1		Enables supply of an input clock. • The DTC can operate.						

Caution The functions mounted depend on the product. For details on the PER1 register, see the description in Section 6 Clock Generator.

16.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 16 - 6 Format of DTC Control Register j (DTCCRj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE
SZ		Transfer data size selection						
0		8 bits						
1		16 bits						
RPTINT		Enabling/disabling repeat mode interrupts						
0		Interrupt generation disabled						
1		Interrupt generation enabled						
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).								
CHNE		Enabling/disabling chain transfers						
0		Chain transfers disabled						
1		Chain transfers enabled						
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).								
DAMOD		Transfer destination address control						
0		Fixed						
1		Incremented						
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).								
SAMOD		Transfer source address control						
0		Fixed						
1		Incremented						
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).								
RPTSEL		Repeat area selection						
0		Transfer destination is the repeat area.						
1		Transfer source is the repeat area.						
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).								
MODE		Transfer mode selection						
0		Normal mode						
1		Repeat mode						

Caution Do not access the DTCCRj register using a DTC transfer.

16.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

The DTBLSj register is used to set the block size of the data to be transferred by one activation.

Figure 16 - 7 Format of DTC Block Size Register j (DTBLSj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer Block Size	
	8-Bit Transfer	16-Bit Transfer
00H	256 bytes	512 bytes
01H	1 byte	2 bytes
02H	2 bytes	4 bytes
03H	3 bytes	6 bytes
•	•	•
•	•	•
•	•	•
FDH	253 bytes	506 bytes
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Caution Do not access the DTBLSj register using a DTC transfer.

16.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

The DTCCTj register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 16 - 8 Format of DTC Transfer Count Register j (DTCCTj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0
DTCCTj	Number of Transfers							
00H	256 times							
01H	Once							
02H	2 times							
03H	3 times							
•	•							
•	•							
•	•							
FDH	253 times							
FEH	254 times							
FFH	255 times							

Caution Do not access the DTCCTj register using a DTC transfer.

16.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

The DTRLDj register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

Figure 16 - 9 Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register using a DTC transfer.

16.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

The DTSARj register is used to specify the transfer source address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest-order bit is ignored and the address is handled as an even address.

Figure 16 - 10 Format of DTC Source Address Register j (DTSARj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTSARj	DTSARj15	DTSARj14	DTSARj13	DTSARj12	DTSARj11	DTSARj10	DTSARj9	DTSARj8
	7	6	5	4	3	2	1	0
	DTSARj7	DTSARj6	DTSARj5	DTSARj4	DTSARj3	DTSARj2	DTSARj1	DTSARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.

Caution 2. Do not access the DTSARj register using a DTC transfer.

16.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

The DTDARj register is used to specify the transfer destination address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest-order bit is ignored and the address is handled as an even address.

Figure 16 - 11 Format of DTC Destination Address Register j (DTDARj)

Address: Refer to **16.3.2 Control data allocation**.

After reset: Undefined

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTDARj	DTDARj15	DTDARj14	DTDARj13	DTDARj12	DTDARj11	DTDARj10	DTDARj9	DTDARj8
	7	6	5	4	3	2	1	0
	DTDARj7	DTDARj6	DTDARj5	DTDARj4	DTDARj3	DTDARj2	DTDARj1	DTDARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer destination address.

Caution 2. Do not access the DTDARj register using a DTC transfer.

16.3.11 DTC activation enable register i (DTCENi) (i = 0 to 4)

The DTCENi register is an 8-bit register which enables or disables DTC activation by interrupt sources.

Table 16 - 4 lists the correspondences between interrupt sources and bits DTCENi0 to DTCENi7.

The DTCENi register can be set by a 1-bit or 8-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.

Caution 2. Do not access the DTCENi register using a DTC transfer.

Caution 3. The assigned functions differ depending on the product. For the bits to which no function is assigned, be sure to set their values to 0.

Figure 16 - 12 Format of DTC Activation Enable Register i (DTCENi) (i = 0 to 4) (1/2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
1	Activation enabled
The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi6	DTC activation enable i6
0	Activation disabled
1	Activation enabled
The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi5	DTC activation enable i5
0	Activation disabled
1	Activation enabled
The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi4	DTC activation enable i4
0	Activation disabled
1	Activation enabled
The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi3	DTC activation enable i3
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Figure 16 - 12 Format of DTC Activation Enable Register i (DTCENi) (i = 0 to 4) (2/2)

DTCENi2	DTC activation enable i2
0	Activation disabled
1	Activation enabled
The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi1	DTC activation enable i1
0	Activation disabled
1	Activation enabled
The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi0	DTC activation enable i0
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Table 16 - 4 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1 ^{Note 3}	INTP2 ^{Note 4}	INTP3	INTP4 ^{Note 3}	INTP5 ^{Note 2}	INTP6 ^{Note 7}
DTCEN1	Reserved	Key input ^{Note 6}	A/D conversion end	UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	UART1 transmission transfer end ^{Note 2}	UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end ^{Note 4}
DTCEN2	UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end ^{Note 4}	Reserved	Reserved	UARTA0 reception transfer end ^{Note 5}	UARTA0 transmission transfer end/buffer empty ^{Note 5}	Reserved	Reserved	End of counting or capturing by channel 0 of timer array unit ^{Note 2}
DTCEN3	End of counting or capturing by channel 1 of timer array unit ^{Note 2}	End of counting or capturing by channel 2 of timer array unit ⁰	End of counting or capturing by channel 3 of timer array unit ^{Note 3}	End of counting or capturing by channel 4 of timer array unit ^{Note 4}	End of counting or capturing by channel 5 of timer array unit ^{Note 4}	End of counting or capturing by channel 6 of timer array unit ^{Note 4}	End of counting or capturing by channel 7 of timer array unit ^{Note 4}	Fixed-cycle signal of real-time clock/alarm match detection
DTCEN4	Interval signal detection of 32-bit interval timer	Request to write to a configuration register of an individual channel of the capacitive sensing unit	Request to transfer data measured by the capacitive sensing unit	Reserved	Reserved	Event output from the event link controller	Event output from the SNOOZE mode sequencer	Voltage detection ^{Note 1}

Note 1. When bit 6 (LVD0SEL) of the option byte (000C1H) is set to 0 or when bit 7 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Note 2. This is applicable to the 20- to 48-pin products.

Note 3. This is applicable to the 24- to 48-pin products.

Note 4. This is applicable to the 30- to 48-pin products.

Note 5. This is applicable to the 36- to 48-pin products.

Note 6. This is applicable to the 40- to 48-pin products.

Note 7. This is applicable to the 48-pin products.

Caution For the bits to which no function is assigned, be sure to set their values to 0.

Remark i = 0 to 4

16.3.12 DTC base address register (DTCBAR)

The DTCBAR register is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the 8 higher-order bits to generate a 16-bit address.

The DTCBAR register can be set by an 8-bit memory manipulation instruction.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.

Caution 2. Do not rewrite the DTCBAR register more than once.

Caution 3. Do not access the DTCBAR register using a DTC transfer.

Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the cautions in 16.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 16 - 13 Format of the DTC Base Address Register (DTCBAR)

Address: F02E0H
After reset: FDH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

16.4 DTC Operation

When the DTC is activated, the DTC reads control data from the DTC control data area, proceeds with data transfer according to the control data, and writes back the control data after data transfer to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.

There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).

A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj. The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

16.4.1 Activation sources

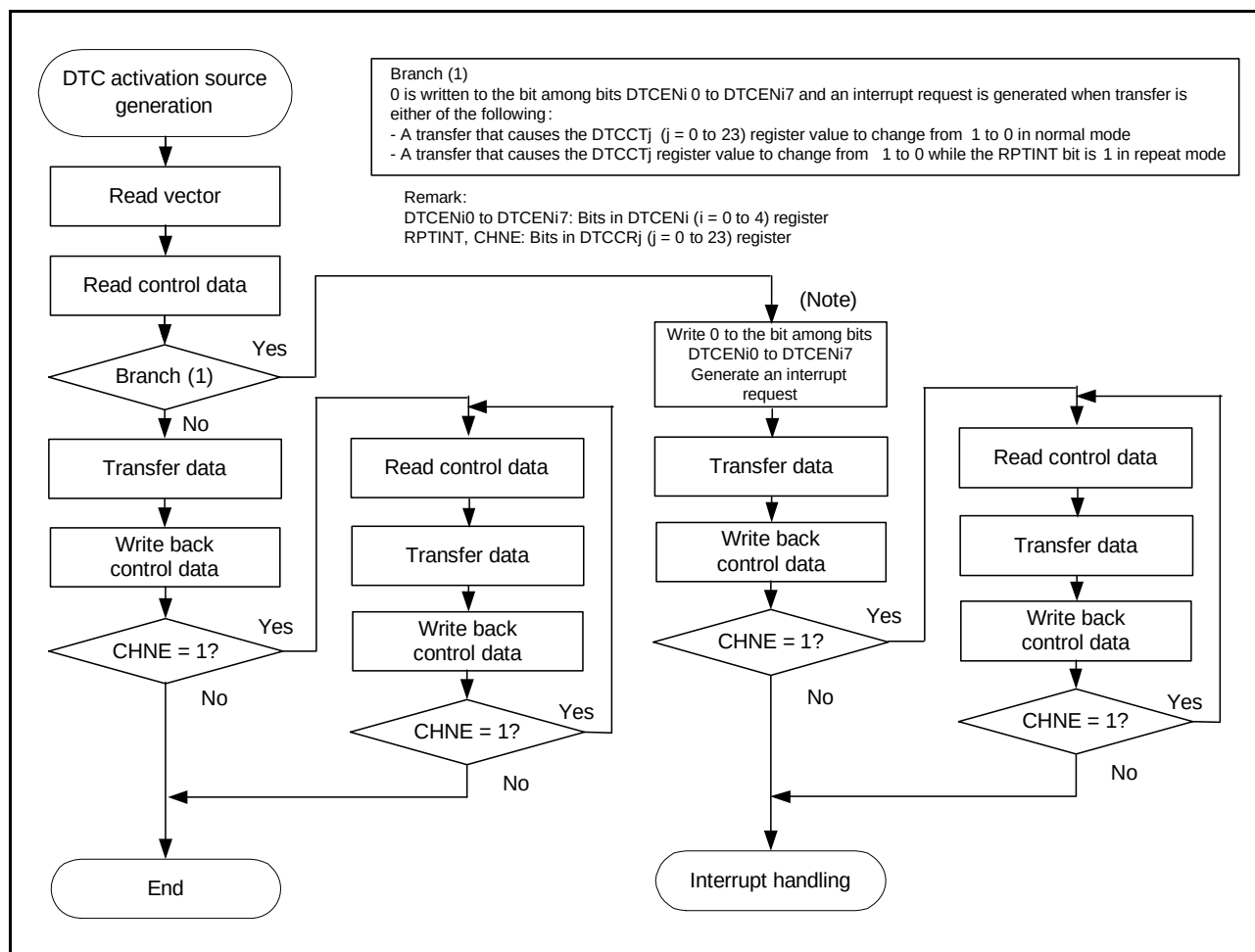
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 4) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 16 - 14 shows DTC internal operation flowchart.

Figure 16 - 14 DTC Internal Operation Flowchart



16.4.2 Normal mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register to 0 (activation disabled).

Table 16 - 5 shows register functions in normal mode. **Figure 16 - 15** shows data transfers in normal mode.

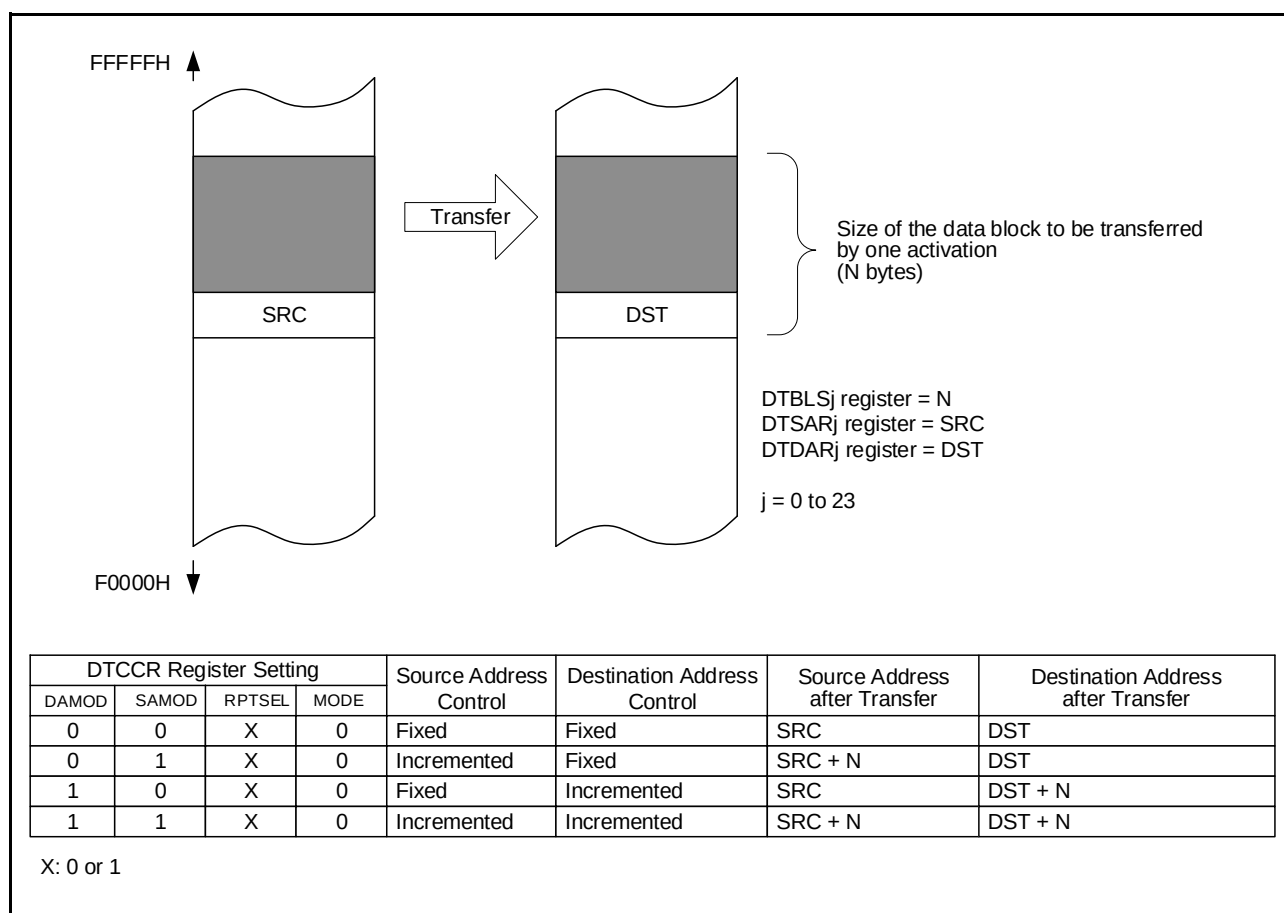
Table 16 - 5 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used ^{Note}
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 16 - 15 Data Transfers in Normal Mode

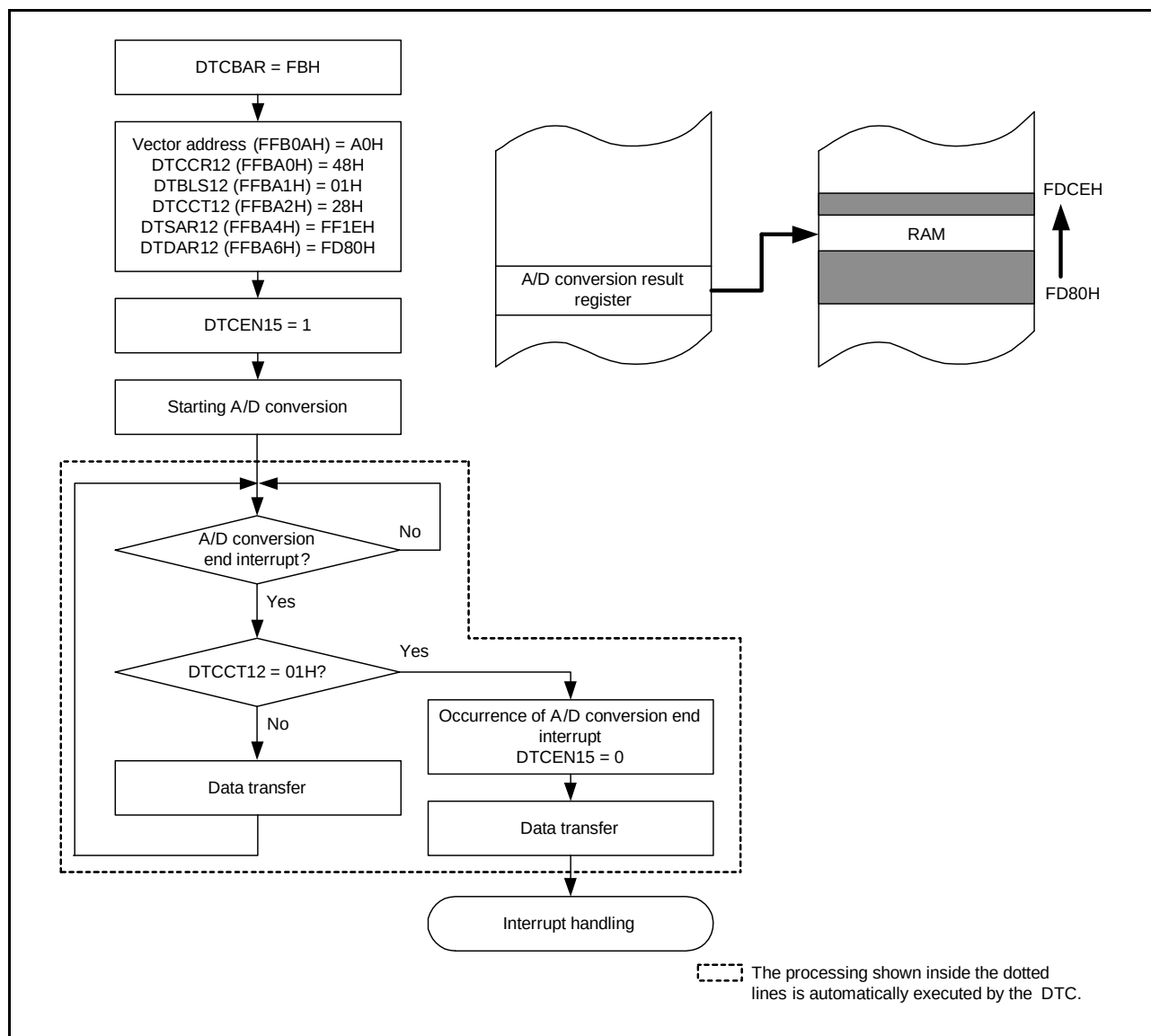


(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFBA0H and control data is allocated at FFBA0H to FFBA7H.
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM 40 times.

Figure 16 - 16 Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results



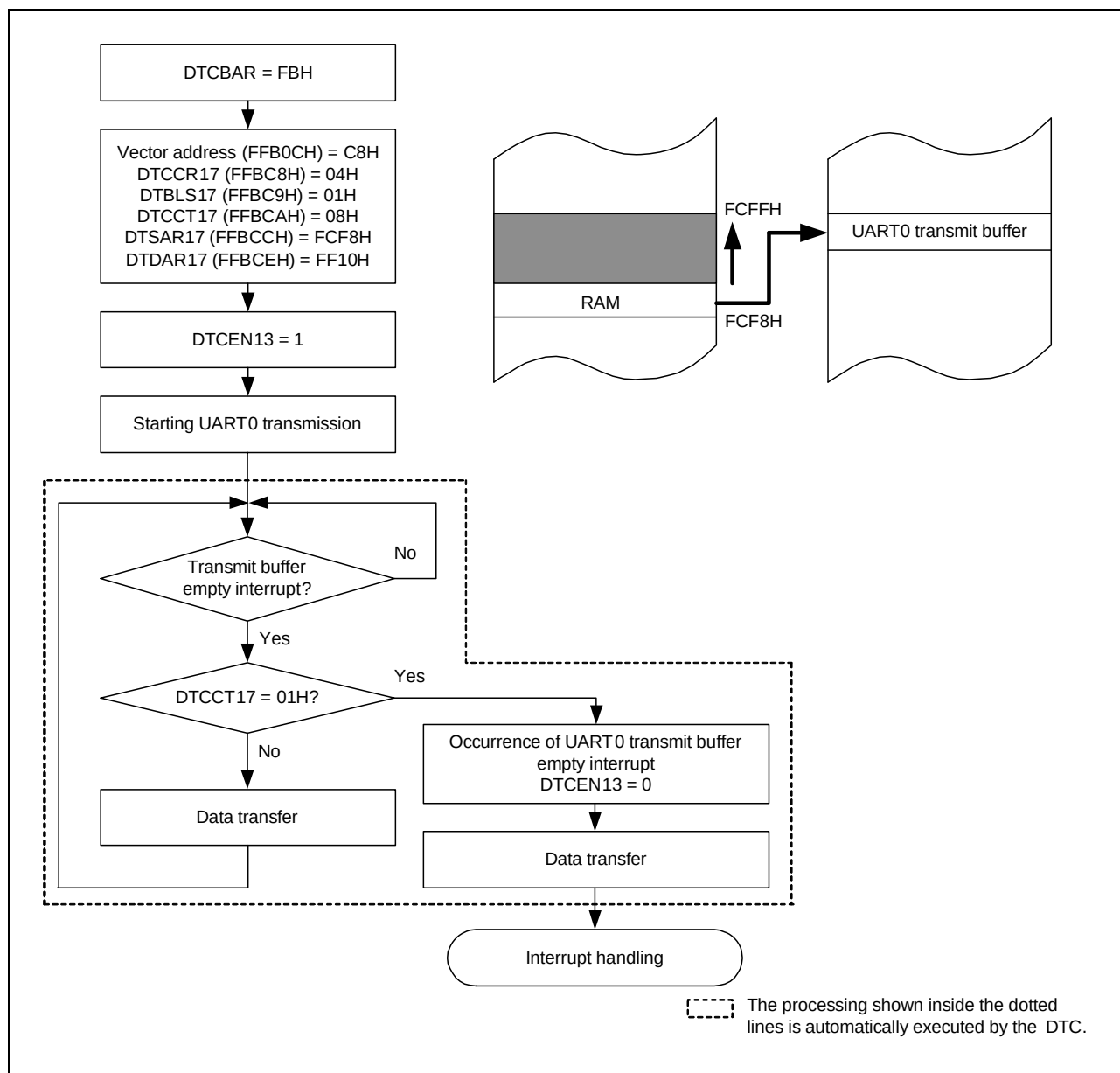
The value of the DTRL12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(2) Example 2 of using normal mode: UART0 consecutive transmission

The DTC is activated by a UART0 transmit buffer empty interrupt and the value of RAM is transferred to the UART0 transmit buffer.

- The vector address is FFB0CH and control data is allocated at FFBC8H to FFBCFH.
- Transfers 8 bytes of data at addresses from FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H).

Figure 16 - 17 Example 2 of Using Normal Mode: UART0 Consecutive Transmission



The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Start the first UART0 transmission by software. The second and subsequent transmissions proceed automatically by using transmit buffer empty interrupts to activate the DTC.

16.4.3 Repeat mode

One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (i = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0.

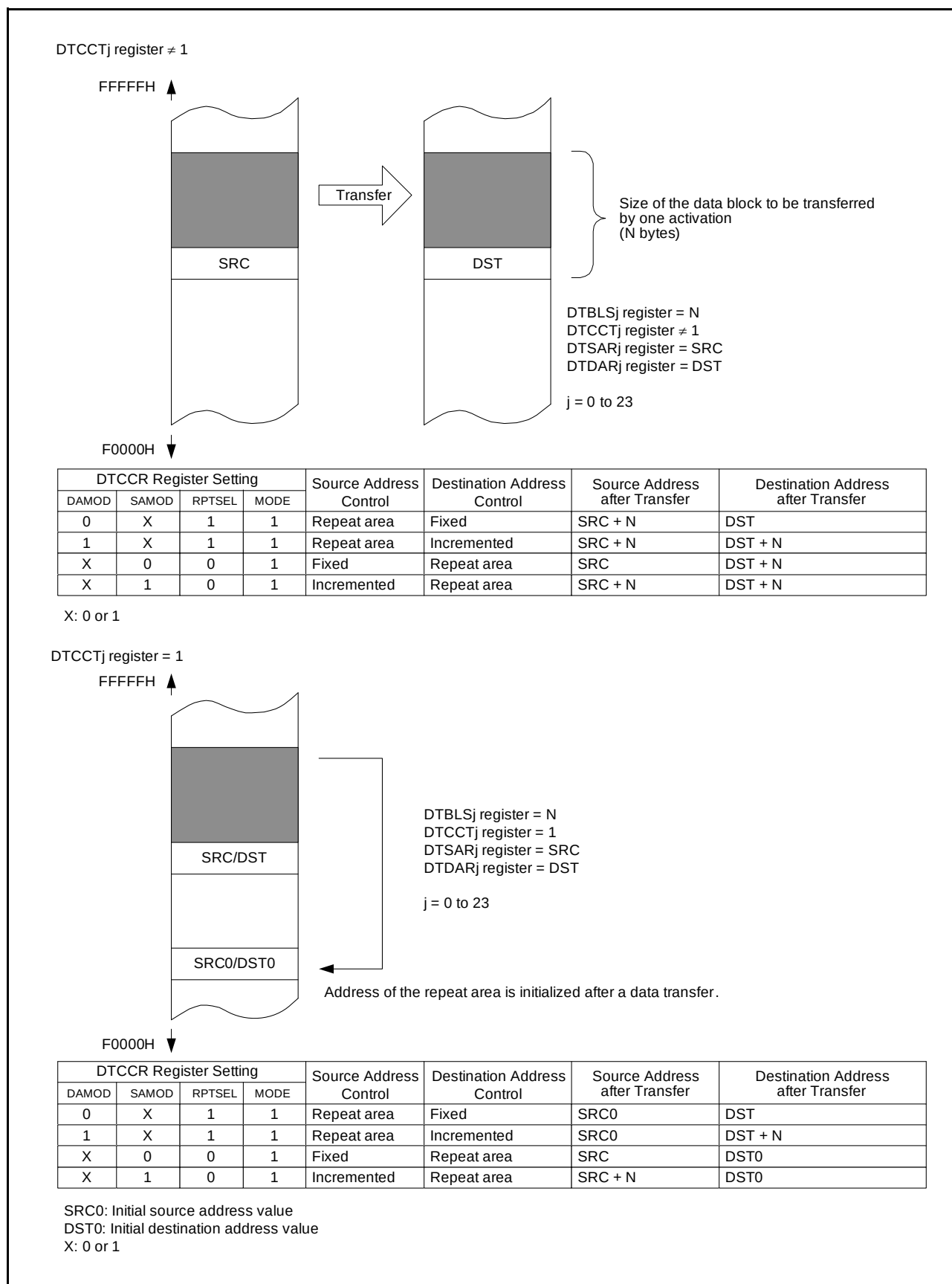
Table 16 - 6 lists register functions in repeat mode. **Figure 16 - 18** shows data transfers in repeat mode.

Table 16 - 6 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCT register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 16 - 18 Data Transfers in Repeat Mode



Caution 1. When repeat mode is used, the 8 lower-order bits of the initial value for the repeat area address must be 00H.

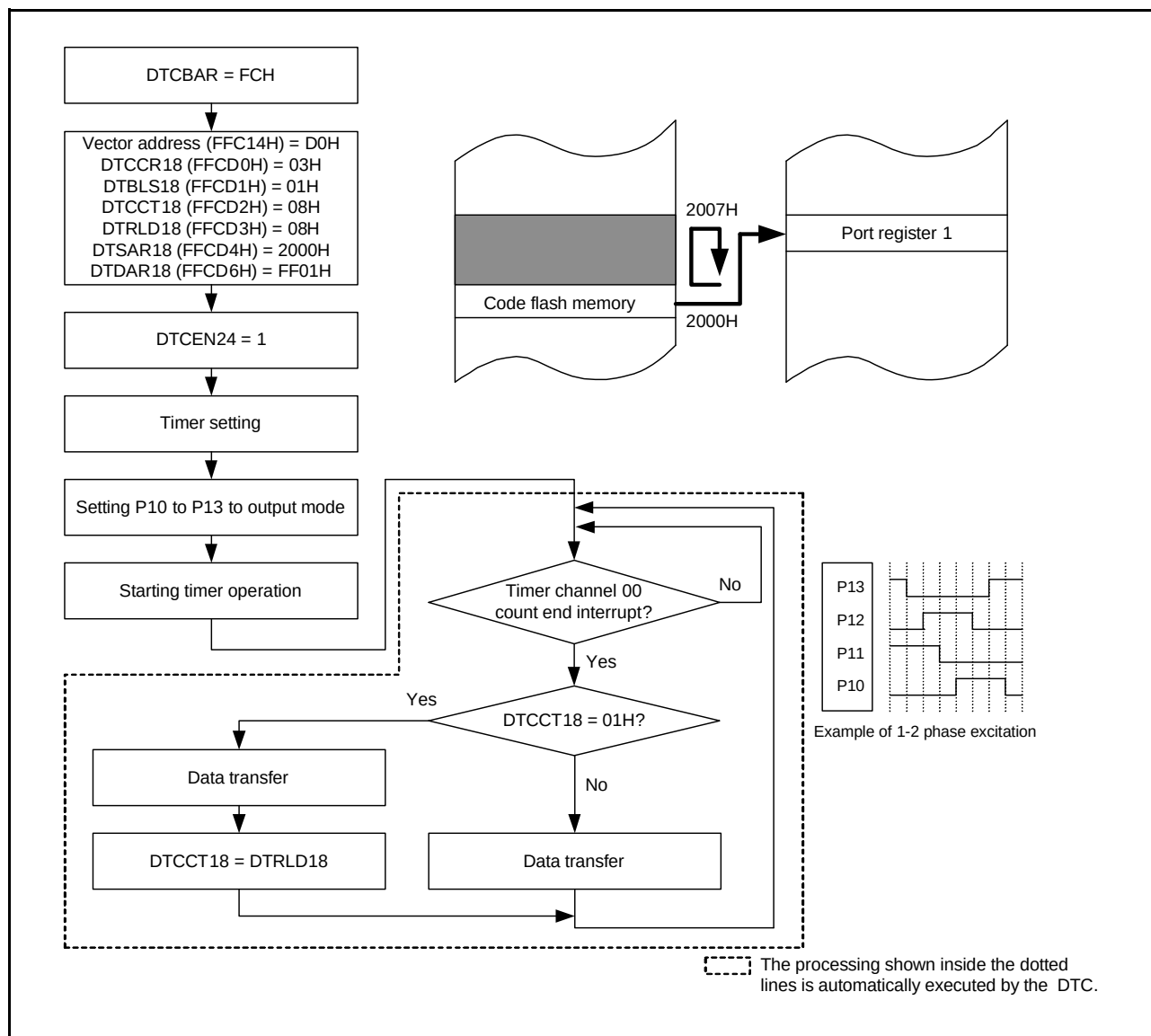
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(1) Example 1 of using repeat mode: Outputting stepping motor control pulses using port pins

The DTC is activated using the interval timer function of channel 0 of timer array unit 0, and the patterns of the motor control pulse stored in the code flash memory are transferred to the general-purpose port pins.

- The vector address is FFC14H and control data is allocated at FFCD0H to FFCD7H.
- Transfers 8-byte data at addresses from 02000H to 02007H of the code flash memory from the mirror area (F2000H to F2007H) to port register 1 (FFF01H).
- A repeat mode interrupt is disabled.

Figure 16 - 19 Example 1 of Using Repeat Mode: Outputting Stepping Motor Control Pulses Using Port Pins



To stop the output, stop the timer first and then clear DTCEN24.

16.4.4 Chain transfers

When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

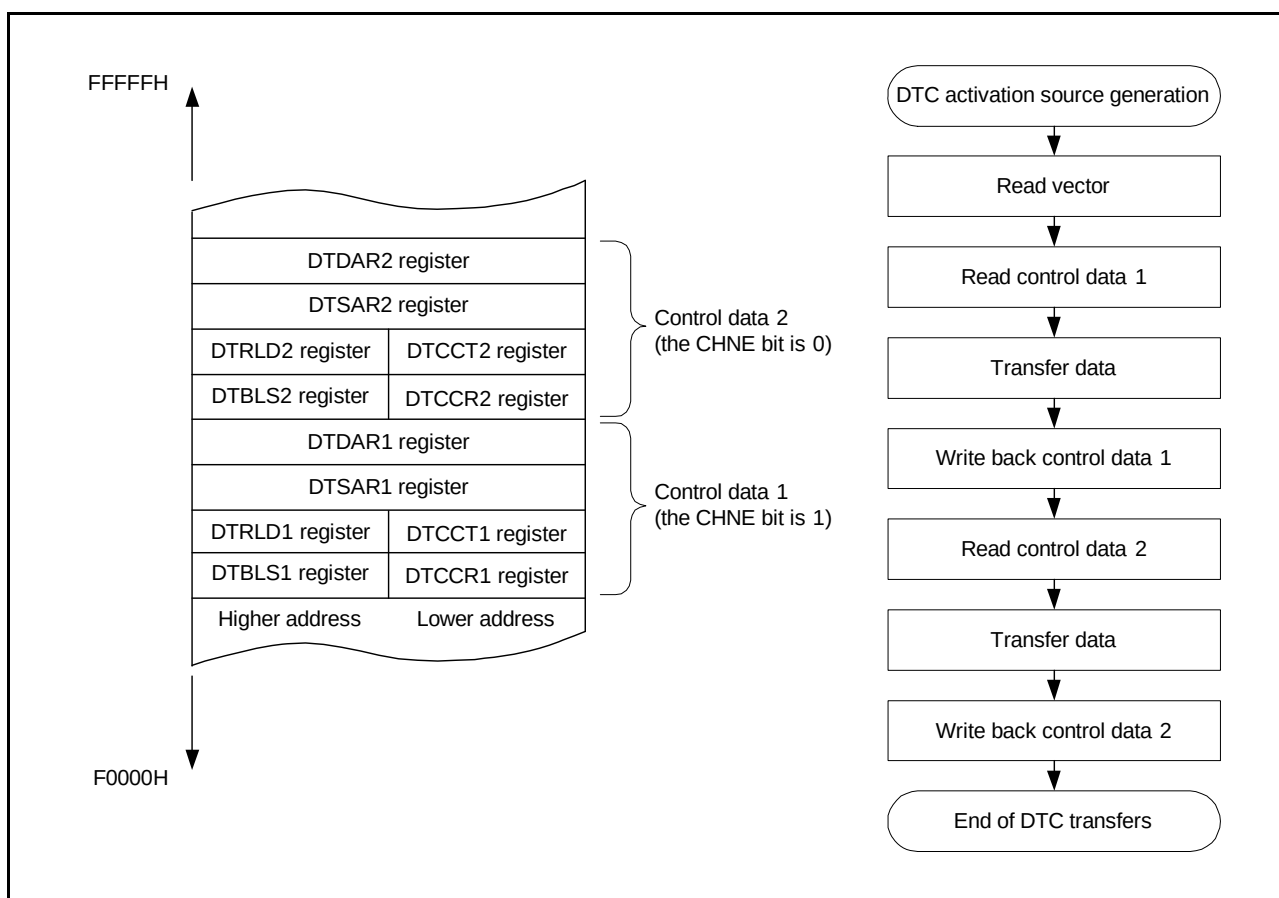
When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area.

When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read to continue a transfer after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.

Figure 16 - 20 shows data transfers during chain transfers.

Figure 16 - 20 Data Transfers during Chain Transfers

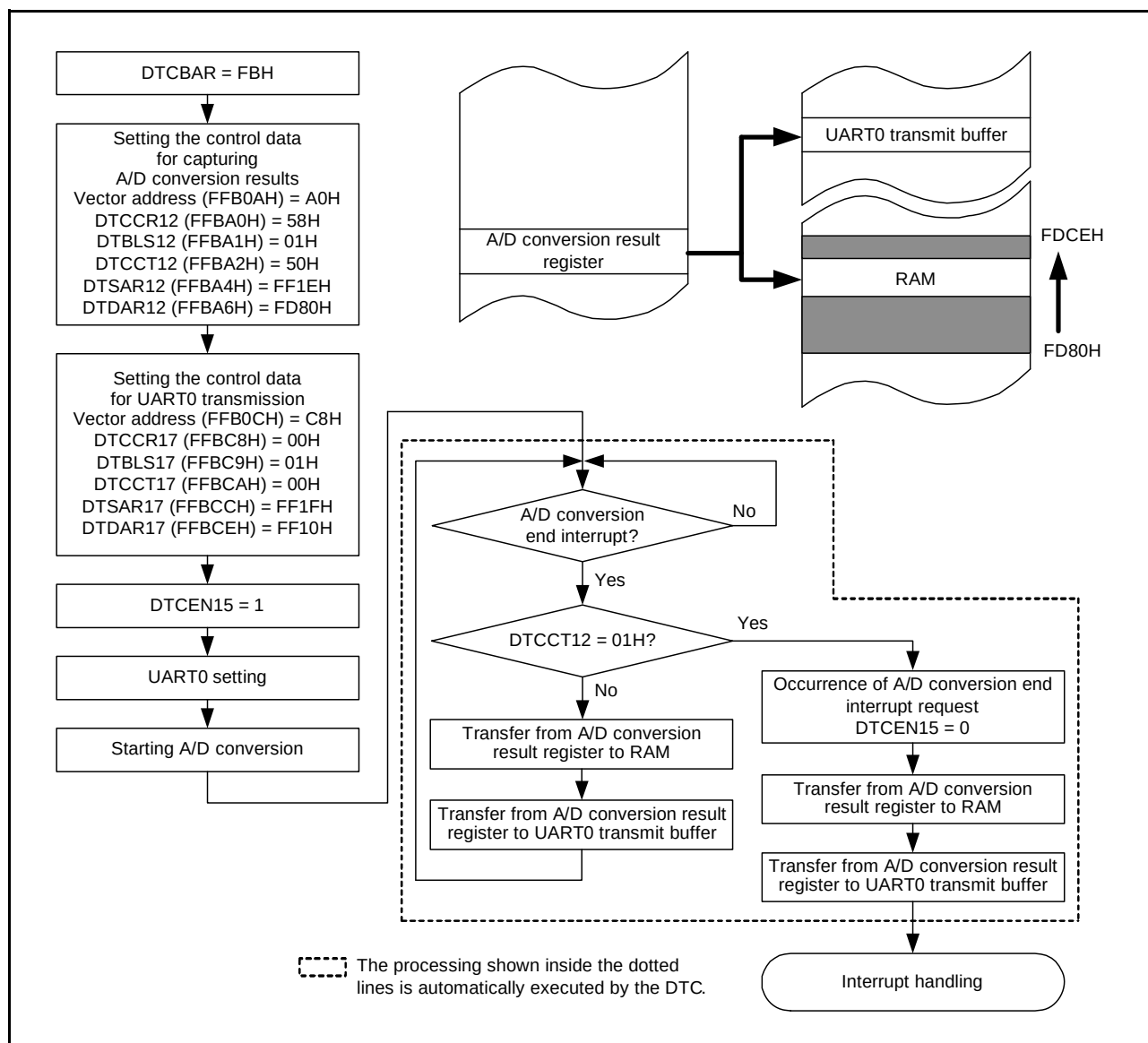


Note 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

Note 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 4) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

- (1) Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission
- The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART0.
- The vector address is FFB0AH.
 - The control data for capturing A/D conversion results is allocated at FFBA0H to FFBA7H.
 - The control data for UART0 transmission is allocated at FFBA8H to FFBAFH.
 - Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFD8FH of RAM, and transfers the 1 higher-order byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H).

Figure 16 - 21 Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART0 Transmission



16.5 Points for Caution when the DTC is to be Used

16.5.1 Setting DTC control data and vector table

- Do not access the DTC extended special function registers (2nd SFRs), the DTC control data area, the DTC vector table area, or the general-purpose register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 4) register is 0 (activation disabled).

16.5.2 Allocation of DTC control data area and DTC vector table area

The areas where the DTC control data and vector table can be allocated differ depending on the product and usage conditions.

- It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- Initialize the DTRLD register to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.
- The area where debug monitor programs are allocated when using the on-chip debugging function cannot be used as the DTC control data area or DTC vector table area. For details, see **31.4 Allocation of Memory Spaces to User Resources**.

16.5.3 DTC pending instruction

Even if a DTC transfer request is generated, DTC transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- Multiply/divide/multiply & accumulate instruction (excluding MULU)

Caution 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

16.5.4 Operation when accessing data flash memory space

When accessing the data flash space after an instruction execution from the start of DTC data transfer, a wait of three clock cycles will be inserted to the next instruction.

Instruction 1

DTC data transfer

Instruction 2 ← The wait of three clock cycles occurs.

MOV A, ! Data Flash space

16.5.5 Number of DTC execution clock cycles

Table 16 - 7 lists the operations following DTC activation and required number of cycles for each operation.

Table 16 - 7 Operations following DTC Activation and Required Number of Cycles

Vector Read	Control Data		Data read	Data Write
	Read	Write-back		
1	4	Note 1	Note 2	Note 2

Note 1. For the number of clock cycles required for control data write-back, refer to **Table 16 - 8 Number of Clock Cycles Required for Control Data Write-Back Operation**.

Note 2. For the number of clock cycles required for data read/write, refer to **Table 16 - 9 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 16 - 8 Number of Clock Cycles Required for Control Data Write-Back Operation

DTCCR Register Setting				Address Setting		Control Register to be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	X	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	X	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	X	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	X	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	X	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	X	1	1		Incremented	Written back	Written back	Written back	Written back	3
X	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; X: 0 or 1

Table 16 - 9 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special function register (SFR)	Extended special function register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait cycles ^{Note}
Data write	1	—	—	1	1	1 + number of wait cycles ^{Note}

Note The number of wait cycles differs depending on the specifications of the register to be accessed in the extended special function register (2nd SFR) area.

16.5.6 DTC response time

Table 16 - 10 lists the DTC response time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts. It does not include the number of DTC execution clock cycles.

Table 16 - 10 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 cycles	19 cycles

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the condition.

- When executing an instruction from the internal RAM
Maximum response time: 20 cycles
- When executing a DTC pending instruction (refer to **16.5.3 DTC pending instruction**)
Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the condition.
- When accessing the TRJ0 register that a wait occurs
Maximum response time: Maximum response time for each condition + 1 cycle

Remark 1 cycle: 1/fCLK (fCLK: CPU/peripheral hardware clock)

16.5.7 DTC activation sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **16.3.3 Vector table**.

16.5.8 Operation in standby mode

State	DTC Operation
HALT mode	Operable ^{Note 1}
STOP mode	DTC activation sources can be accepted ^{Note 3}
SNOOZE mode	Operable ^{Notes 2, 4, 5, 6}

Note 1. When the subsystem clock is selected as fCLK, operation is disabled if the RTCLPC bit of the OSMC register is 1.

Note 2. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected as fCLK.

Note 3. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the chip returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 4. When a transfer end interrupt from the CSIp in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the CSIp (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting the SSCm register, and writing 1 to the SSm0 bit) again.

Note 5. When a transfer end interrupt from the UARTq in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the UARTq (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting the SSCm register, and writing 1 to the SSm1 bit) again.

Note 6. When an A/D conversion end interrupt from the A/D converter in SNOOZE mode is being used as the DTC activation source, use the A/D conversion end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for the SNOOZE mode function of the A/D converter (writing 1 to the AWC bit after having written 0 to it) again.

Remark p = 00; q = 0; m = 0

Section 17 Event Link Controller (ELC)

17.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

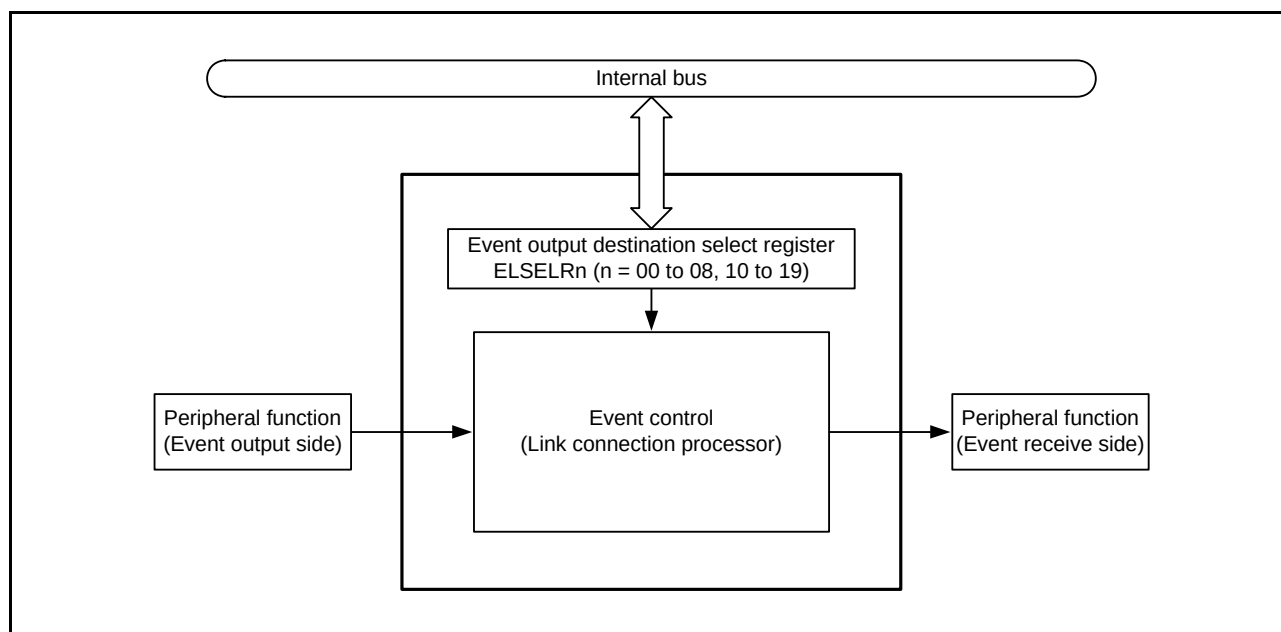
The ELC has the following functions.

- Capable of directly linking event signals from 20 types of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of seven types of peripheral functions

17.2 Configuration of ELC

Figure 17 - 1 shows the block diagram of the ELC.

Figure 17 - 1 Block Diagram of the ELC



17.3 Registers for Controlling the ELC

The following registers are used to control the ELC.

- Event output destination select register 00 (ELSELR00)
- Event output destination select register 01 (ELSELR01)**Note 1**
- Event output destination select register 02 (ELSELR02)**Note 2**
- Event output destination select register 03 (ELSELR03)
- Event output destination select register 04 (ELSELR04)**Note 1**
- Event output destination select register 05 (ELSELR05)**Note 3**
- Event output destination select register 06 (ELSELR06)**Note 4**
- Event output destination select register 07 (ELSELR07)**Note 5**
- Event output destination select register 08 (ELSELR08)
- Event output destination select register 10 (ELSELR10)
- Event output destination select register 11 (ELSELR11)
- Event output destination select register 12 (ELSELR12)
- Event output destination select register 13 (ELSELR13)
- Event output destination select register 14 (ELSELR14)**Note 3**
- Event output destination select register 15 (ELSELR15)**Note 3**
- Event output destination select register 16 (ELSELR16)
- Event output destination select register 17 (ELSELR17)**Note 1**
- Event output destination select register 18 (ELSELR18)**Note 6**
- Event output destination select register 19 (ELSELR19)**Note 6**

Note 1. This register is only mounted in 24- to 48-pin products.

Note 2. This register is only mounted in 30- to 48-pin products.

Note 3. This register is only mounted in 20- to 48-pin products.

Note 4. This register is only mounted in 48-pin products.

Note 5. This register is only mounted in 40- to 48-pin products.

Note 6. This register is only mounted in 36- to 48-pin products.

17.3.1 Event output destination select register n (ELSELRn) (n = 00 to 08, 10 to 19)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals and the function of the event output destination (event receive side) is stopped.

Table 17 - 1 lists the correspondence between ELSELRn (n = 00 to 08, 10 to 19) registers and peripheral functions, and **Table 17 - 2** lists the correspondence between values set to ELSELRn (n = 00 to 08, 10 to 19) registers and operation of link destination peripheral functions at reception.

Figure 17 - 2 Format of Event Output Destination Select Register n (ELSELRn)

Address: F0240H (ELSELR00) to F0248H (ELSELR08)
F024AH (ELSELR10) to F0253H (ELSELR19)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn3	ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	0	Event link disabled
0	0	0	1	Select operation of peripheral function 1 to link Note
0	0	1	0	Select operation of peripheral function 2 to link Note
0	0	1	1	Select operation of peripheral function 3 to link Note
0	1	0	0	Select operation of peripheral function 4 to link Note
0	1	0	1	Select operation of peripheral function 5 to link Note
0	1	1	0	Select operation of peripheral function 6 to link Note
0	1	1	1	Select operation of peripheral function 7 to link Note
Other than above				Setting prohibited

Note See Table 17 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 08, 10 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception.

Table 17 - 1 Correspondence between ELSELRn (n = 00 to 08, 10 to 19) Registers and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01 ^{Note 1}	External interrupt edge detection 1	INTP1
ELSELR02 ^{Note 2}	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04 ^{Note 1}	External interrupt edge detection 4	INTP4
ELSELR05 ^{Note 3}	External interrupt edge detection 5	INTP5
ELSELR06 ^{Note 4}	External interrupt edge detection 6	INTP6
ELSELR07 ^{Note 5}	Key return signal detection	INTKR
ELSELR08	Realtime clock fixed-cycle signal/alarm match detection	INTRTC
ELSELR10	32-bit interval timer 0 compare match	INTIT00
ELSELR11	32-bit interval timer 1 compare match	INTIT01
ELSELR12	32-bit interval timer 2 compare match	INTIT10
ELSELR13	32-bit interval timer 3 compare match	INTIT11
ELSELR14 ^{Note 3}	TAU channel 00 count end/capture end	INTTM00
ELSELR15 ^{Note 3}	TAU channel 01 count end/capture end	INTTM01
ELSELR16	TAU channel 02 count end/capture end	INTTM02
ELSELR17 ^{Note 1}	TAU channel 03 count end/capture end	INTTM03
ELSELR18 ^{Note 6}	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end	INTST0/INTCSI00/INTIIC00
ELSELR19 ^{Note 6}	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end	INTSR0/INTCSI01/INTIIC01

Note 1. This register is only mounted in 24- to 48-pin products.

Note 2. This register is only mounted in 30- to 48-pin products.

Note 3. This register is only mounted in 20- to 48-pin products.

Note 4. This register is only mounted in 48-pin products.

Note 5. This register is only mounted in 40- to 48-pin products.

Note 6. This register is only mounted in 36- to 48-pin products.

Table 17 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 08, 10 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELRn3 to ELSELRn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
0001B	1	A/D converter	A/D conversion starts
0010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
0011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
0100B	4	SNOOZE mode sequencer	Operation starts
0101B	5	Data transfer controller	Operation starts
0110B	6	Capacitive sensing unit	Operation starts
0111B	7	32-bit interval timer	Operation starts

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to fCLK using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to fCLK using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

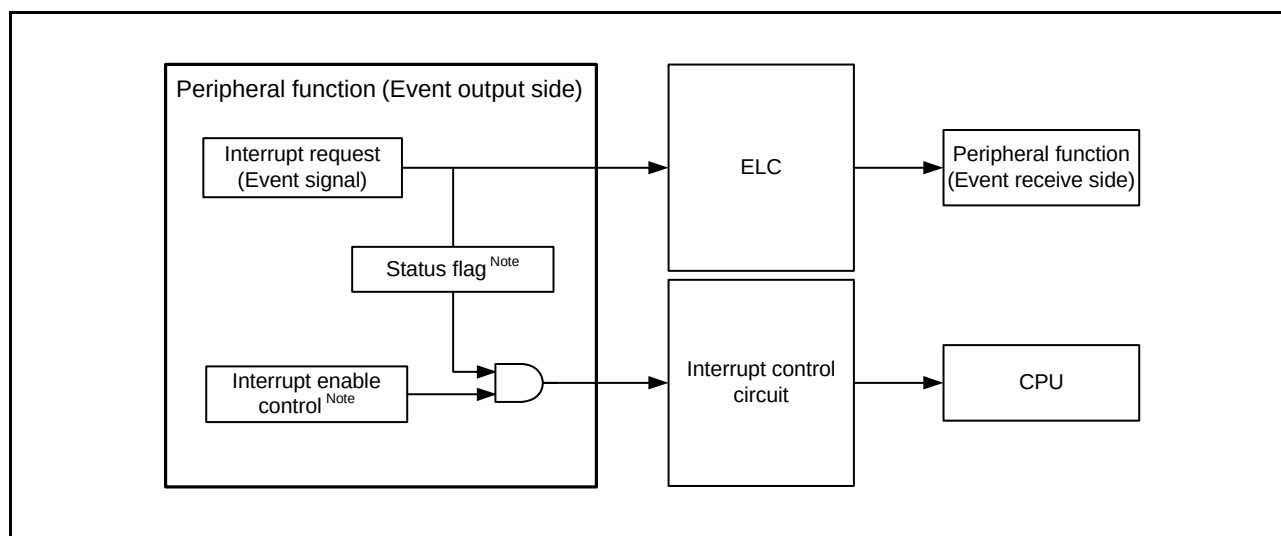
17.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 17 - 3 shows the relationship between interrupt handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event. For details, see **Table 17 - 2 Correspondence between Values Set to ELSELRn (n = 00 to 08, 10 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Figure 17 - 3 Relationship between Interrupt Handling and ELC



Note Not available in some peripheral functions.

Table 17 - 3 lists the response of peripheral functions that receive events.

Table 17 - 3 Response of Peripheral Functions that Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f _{CLK} after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f _{CLK} after an ELC event is generated.
4	SNOOZE mode sequencer	Operation starts	An event from the ELC is directly used as a trigger to start the SNOOZE mode sequencer.
5	Data transfer controller	Operation starts	An event from the ELC is directly used as a trigger to start the data transfer controller.
6	Capacitive sensing unit	Operation starts	An event from the ELC is directly used as an external trigger to start the capacitive sensing unit.
7	32-bit interval timer	Operation starts	An event from the ELC is directly used as a capture trigger for the 32-bit interval timer.

Section 18 Interrupt Functions

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing. The number of interrupt sources differs, depending on the product.

		16-pin	20-pin	24- and 25-pin	30- and 32-pin	36-pin	40- and 44-pin	48-pin
Maskable interrupts	External	2	3	5	6	6	7	10
	Internal	16	19	21	29	32	32	32

18.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For default priority, see **Table 18 - 1**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

18.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and a software interrupt. In addition, they also have up to seven reset sources (see **Table 18 - 1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 18 - 1 Interrupt Source List (1/3)

Interrupt Type	Default Priority>Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type>Note 2	48- pin	44- pin	40- pin	36- pin	32- pin	30- pin	25- pin	24- pin	20- pin	16- pin
		Name	Trigger													
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/2fIL)	Internal	00004H	(A)	√	√	√	√	√	√	√	√	√	√
	1	INTLVI	Voltage detection Note 4		00006H		√	√	√	√	√	√	√	√	√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√	√	√	√	√	√	√	√	√
	3	INTP1			0000AH		√	√	√	√	√	√	√	—	—	—
	4	INTP2			0000CH		√	√	√	√	√	—	—	—	—	—
	5	INTP3			0000EH		√	√	√	√	√	√	√	√	√	√
	6	INTP4			00010H		√	√	√	√	√	√	√	—	—	—
	7	INTP5			00012H		√	√	√	√	√	√	√	√	√	—
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	00014H	(A)	√	√	√	√	√	√	—	—	—	—
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		√	√	√	√	Note 5	Note 5	—	—	—	—
	10	INTSRE2	UART2 reception communication error		00018H		√	√	√	√	√	√	—	—	—	—
	11	—	Reserved		0001AH		—	—	—	—	—	—	—	—	—	—
	12	INTSMSE	Event output from the SNOOZE mode sequencer		0001CH		√	√	√	√	√	√	√	√	√	√
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		√	√	√	√	√	√	√	√	√	√
	14	INTTM00	End of timer channel 00 count or capture		00020H		√	√	√	√	√	√	√	√	√	—
	15	INTSRE0	UART0 reception communication error		00022H		√	√	√	√	√	√	√	√	√	√
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				√	√	√	√	√	√	√	√	√	—
	16	INTST1	UART1 transmission transfer end or buffer empty interrupt		00024H		√	√	√	√	√	√	√	√	√	—
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end		00026H		√	√	√	√	√	√	√	√	√	Note 7
	18	INTSRE1	UART1 reception communication error		00028H		√	√	√	√	√	√	√	√	√	—
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				√	√	√	√	√	√	√	—	—	—
	19	INTIICA0	End of IICA0 communication		0002AH		√	√	√	√	√	√	√	√	—	—
	20	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		0002CH		√	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6
	21	INTTM01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)		0002EH		√	√	√	√	√	√	√	√	√	—
	22	INTTM02	End of timer channel 02 count or capture		00030H		√	√	√	√	√	√	√	√	√	√

Table 18 - 1 Interrupt Source List (2/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	48- pin	44- pin	40- pin	36- pin	32- pin	30- pin	25- pin	24- pin	20- pin	16- pin
		Name	Trigger													
Maskable	23	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)	Internal	00032H	(A)	√	√	√	√	√	√	√	√	—	—
	24	INTAD	End of A/D conversion		00034H		√	√	√	√	√	√	√	√	√	√
	25	INTRTC	Fixed-cycle signal of realtime clock/alarm match detection		00036H		√	√	√	√	√	√	√	√	√	√
	26	INTITL	Interval signal of 32-bit interval timer detection		00038H		√	√	√	√	√	√	√	√	√	√
	27	INTKR	Key return signal detection	External	0003AH	(C)	√	√	√	—	—	—	—	—	—	—
	28	—	Reserved	Internal	0003CH	(A)	—	—	—	—	—	—	—	—	—	—
	29	—	Reserved		0003EH		—	—	—	—	—	—	—	—	—	—
	30	—	Reserved		00040H		—	—	—	—	—	—	—	—	—	—
	31	INTTM04	End of timer channel 04 count or capture		00042H		√	√	√	√	√	—	—	—	—	—
	32	INTTM05	End of timer channel 05 count or capture		00044H		√	√	√	√	√	—	—	—	—	—
	33	INTTM06	End of timer channel 06 count or capture		00046H		√	√	√	√	√	—	—	—	—	—
	34	INTTM07	End of timer channel 07 count or capture		00048H		√	√	√	√	√	—	—	—	—	—
	35	INTP6	Pin input edge detection	External	0004AH	(B)	√	—	—	—	—	—	—	—	—	—
	36	—			0004CH		—	—	—	—	—	—	—	—	—	—
	37	INTP8			0004EH		√	—	—	—	—	—	—	—	—	—
	38	INTP9			00050H		√	—	—	—	—	—	—	—	—	—
	39	INTFL	Reserved		00052H		√	√	√	√	√	√	√	√	√	√
	40	—	Reserved		00054H		—	—	—	—	—	—	—	—	—	—
	41	—	Reserved		00056H		—	—	—	—	—	—	—	—	—	—
	42	INTURE0	UARTA0 reception communication error	Internal	00058H	(A)	√	√	√	√	—	—	—	—	—	—
	43	—	Reserved		0005AH		—	—	—	—	—	—	—	—	—	—
	44	—	Reserved		0005CH		—	—	—	—	—	—	—	—	—	—
	45	—	Reserved		0005EH		—	—	—	—	—	—	—	—	—	—
	46	INTCTSUWR	Request to write to a configuration register of an individual CTSU channel		00060H		√	√	√	√	√	√	√	√	√	√
	47	—	Reserved		00062H		—	—	—	—	—	—	—	—	—	—
	48	INTCTSURD	Request to transfer data measured by the CTSU		00064H		√	√	√	√	√	√	√	√	√	√
	49	INTCTSUFN	CTSU measurement end		00066H		√	√	√	√	√	√	√	√	√	√
	50	—	Reserved		00068H		—	—	—	—	—	—	—	—	—	—
	51	INTUTO	UARTA0 transmission transfer end or buffer empty interrupt		0006AH		√	√	√	√	—	—	—	—	—	—
	52	INTURO	UARTA0 reception transfer end		0006CH		√	√	√	√	—	—	—	—	—	—

Table 18 - 1 Interrupt Source List (3/3)

Interrupt Type	Default Priority ^{Note 1}	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type ^{Note 2}	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
		Name	Trigger													
Maskable	53	—	Reserved	Internal	0006EH	(A)	—	—	—	—	—	—	—	—	—	—
	54	—	Reserved		00070H		—	—	—	—	—	—	—	—	—	—
	55	—	Reserved		00072H		—	—	—	—	—	—	—	—	—	—
	56	—	Reserved		00074H		—	—	—	—	—	—	—	—	—	—
	57	—	Reserved		00076H		—	—	—	—	—	—	—	—	—	—
	58	—	Reserved		00078H		—	—	—	—	—	—	—	—	—	—
Software	—	BRK	Execution of BRK instruction	—	0007EH	(D)	√	√	√	√	√	√	√	√	√	√
Reset	—	RESET	RESET pin input	—	00000H	—	√	√	√	√	√	√	√	√	√	√
		POR	Power-on-reset				√	√	√	√	√	√	√	√	√	√
		LVD	Voltage detection				√	√	√	√	√	√	√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√	√	√	√	√	√	√
		TRAP	Execution of illegal instruction				√	√	√	√	√	√	√	√	√	√
		IAW	Illegal-memory access				√	√	√	√	√	√	√	√	√	√
		RPE	RAM parity error				√	√	√	√	√	√	√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 58 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in **Figure 18 - 1**.

Note 3. When the value of bit 7 (WDTINT) in the option byte (000C0H) is 1.

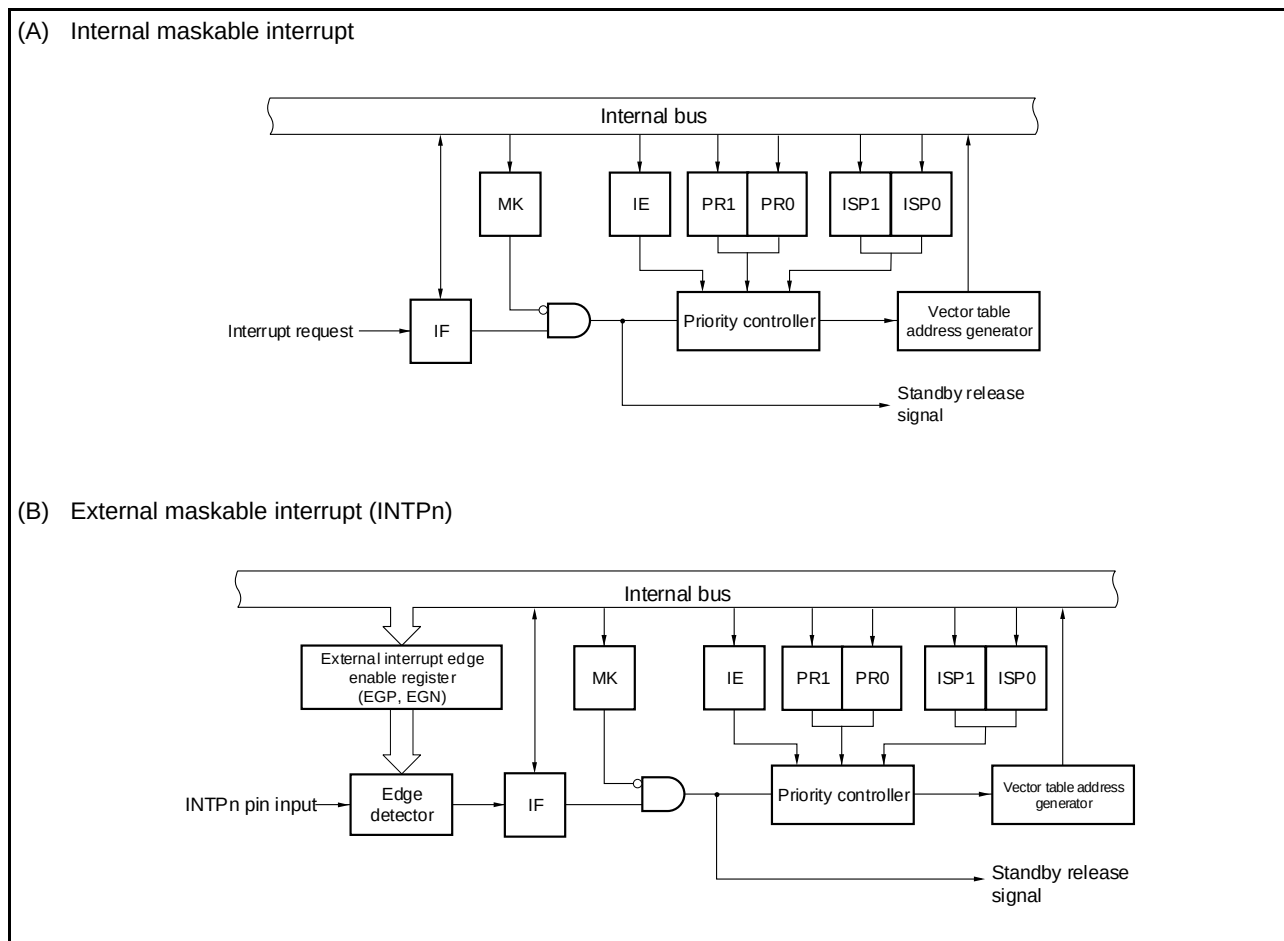
Note 4. When the value of bit 6 (LVD1SEL) in the voltage detection level register (LVIS) is 0 or the value of bit 7 (LVD0SEL) in the option byte (000C1H) is 1.

Note 5. INTSR2 is only present in this product.

Note 6. INTSR0 is only present in this product.

Note 7. INTIIC11 is only present in this product.

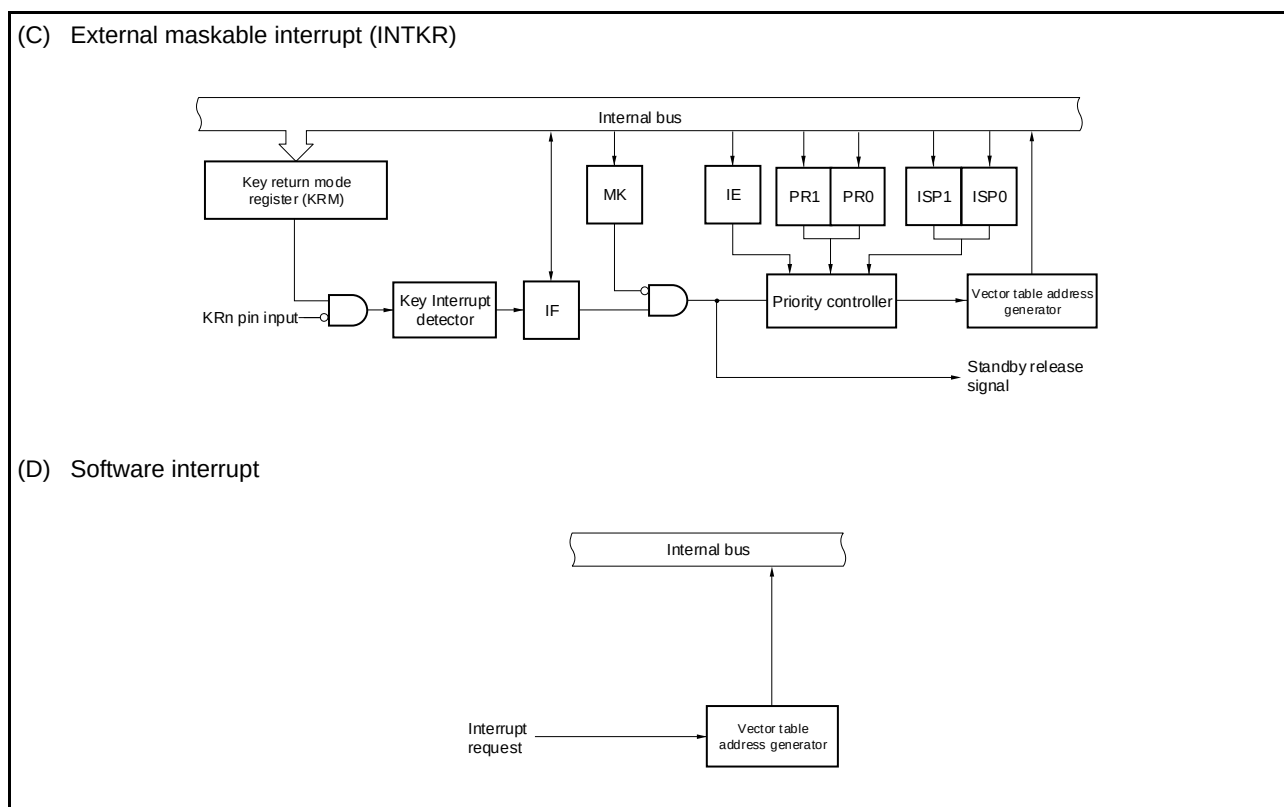
Figure 18 - 1 Basic Configuration of Interrupt Function (1/2)



- Remark 1.** IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

- Remark 2.** 16-pin: n = 0, 3
20-pin: n = 0, 3, 5
24- and 25-pin: n = 0, 1, 3 to 5
30-, 32-, 36-, 40-, and 44-pin: n = 0 to 5
48-pin: n = 0 to 6, 8 to 9

Figure 18 - 1 Basic Configuration of Interrupt Function (2/2)



- Remark 1.** IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

- Remark 2.** 40- and 44-pin: n = 0 to 3
48-pin: n = 0 to 5

18.3 Registers for Controlling the Interrupt Functions

The following registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)
- External interrupt rising edge enable registers (EGP0, EGP1), External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 18 - 2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18 - 2 Flags Corresponding to Interrupt Request Sources (1/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
		Register		Register		Register										
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√	√	√	√	√	√	√	√
INTLVI ^{Note}	LVIIIF ^{Note}		LVIMK ^{Note}		LVIPR0, LVIPR1 ^{Note}		√	√	√	√	√	√	√	√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√	√	√	√	√	√	√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√	√	√	√	√	√	√	—	—
INTP2	PIF2		PMK2		PPR02, PPR12		√	√	√	√	√	—	—	—	—	—
INTP3	PIF3		PMK3		PPR03, PPR13		√	√	√	√	√	√	√	√	√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√	√	√	√	√	√	—	—	—
INTP5	PIF5		PMK5		PPR05, PPR15		√	√	√	√	√	√	√	√	√	—

Note The DLVD0F and DLVD1F bits of the LVIM register can be used to confirm which voltage detector has issued the given interrupt, LVD0 or LVD1. For details, see **23.3.1 Voltage detection register (LVIM)**.

Table 18 - 2 Flags Corresponding to Interrupt Request Sources (2/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
		Register		Register		Register										
INTST2 ^{Note 1}	STIF2 ^{Note 1}	IF0H	STMK2 ^{Note 1}	MK0H	STPR02, STPR12 ^{Note 1}	PR00H, PR10H	√	√	√	√	√	√	—	—	—	—
INTCSI20 ^{Note 1}	CSIF20 ^{Note 1}		CSIMK20 ^{Note 1}		CSIPR020, CSIPR120 ^{Note 1}		√	√	√	√	√	√	—	—	—	—
INTIIC20 ^{Note 1}	IICIF20 ^{Note 1}		IICMK20 ^{Note 1}		IICPR020, IICPR120 ^{Note 1}		√	√	√	√	√	√	—	—	—	—
INTSR2 ^{Note 2}	SRIF2 ^{Note 2}		SRMK2 ^{Note 2}		SRPR02, SRPR12 ^{Note 2}		√	√	√	√	√	√	—	—	—	—
INTCSI21 ^{Note 2}	CSIF21 ^{Note 2}		CSIMK21 ^{Note 2}		CSIPR021, CSIPR121 ^{Note 2}		√	√	√	√	—	—	—	—	—	—
INTIIC21 ^{Note 2}	IICIF21 ^{Note 2}		IICMK21 ^{Note 2}		IICPR021, IICPR121 ^{Note 2}		√	√	√	√	—	—	—	—	—	—
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		√	√	√	√	√	√	—	—	—	—
INTSMSE	SMSEIF		SMSEMK		SMSEPR0, SMSEPR1		√	√	√	√	√	√	√	√	√	√
INTST0 ^{Note 3}	STIF0 ^{Note 3}		STMK0 ^{Note 3}		STPR00, STPR10 ^{Note 3}		√	√	√	√	√	√	√	√	√	√
INTCSI00 ^{Note 3}	CSIF00 ^{Note 3}		CSIMK00 ^{Note 3}		CSIPR000, CSIPR100 ^{Note 3}		√	√	√	√	√	√	√	√	√	√
INTIIC00 ^{Note 3}	IICIF00 ^{Note 3}		IICMK00 ^{Note 3}		IICPR000, IICPR100 ^{Note 3}		√	√	√	√	√	√	√	√	√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√	√	√	√	√	√	√	—
INTSRE0 ^{Note 4}	SREIF0 ^{Note 4}		SREMK0 ^{Note 4}		SREPR00, SREPR10 ^{Note 4}		√	√	√	√	√	√	√	√	√	√
INTTM01H ^{Note 4}	TMIF01H ^{Note 4}		TMMK01H ^{Note 4}		TMPR001H, TMPR101H ^{Note 4}		√	√	√	√	√	√	√	√	√	—

- Note 1.** If any of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.
- Note 2.** If any of the interrupt sources INTSR2, INTCSI21, and INTIIC21 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.
- Note 3.** If any of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.
- Note 4.** Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt source INTSRE0 or INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Table 18 - 2 Flags Corresponding to Interrupt Request Sources (3/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
		Register		Register		Register										
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	√	√	√	√	√	√	√	√	√	√
INTSR1 ^{Note 1}	SRIF1 ^{Note 1}		SRMK1 ^{Note 1}		SRPR01, SRPR11 ^{Note 1}		√	√	√	√	√	√	√	√	√	—
INTCSI11 ^{Note 1}	CSIF11 ^{Note 1}		CSIMK11 ^{Note 1}		CSIPR011, CSIPR111 ^{Note 1}		√	√	√	√	√	√	√	√	√	—
INTIIC11 ^{Note 1}	IICIF11 ^{Note 1}		IICMK11 ^{Note 1}		IICPR011, IICPR111 ^{Note 1}		√	√	√	√	√	√	√	√	√	√
INTSRE1 ^{Note 2}	SREIF1 ^{Note 2}		SREMK1 ^{Note 2}		SREPR01, SREPR11 ^{Note 2}		√	√	√	√	√	√	√	√	√	—
INTTM03H ^{Note 2}	TMIF03H ^{Note 2}		TMMK03H ^{Note 2}		TMPR003H, TMPR103H ^{Note 2}		√	√	√	√	√	√	√	√	—	—
INTICA0	IICAF0		IICAMK0		IICAPR00, IICAPR10		√	√	√	√	√	√	√	√	—	—
INTSR0 ^{Note 3}	SRIF0 ^{Note 3}		SRMK0 ^{Note 3}		SRPR00, SRPR10 ^{Note 3}		√	√	√	√	√	√	√	√	√	√
INTCSI01 ^{Note 3}	CSIF01 ^{Note 3}		CSIMK01 ^{Note 3}		CSIPR001, CSIPR101 ^{Note 3}		√	—	—	—	—	—	—	—	—	—
INTIIC01 ^{Note 3}	IICIF01 ^{Note 3}		IICMK01 ^{Note 3}		IICPR001, IICPR101 ^{Note 3}		√	—	—	—	—	—	—	—	—	—
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√	√	√	√	√	√	√	√	—
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√	√	√	√	√	√	√	√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√	√	√	√	√	√	√	—	—
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√	√	√	√	√	√	√	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√	√	√	√	√	√	√	√	√
INTTML	TMLIF		TMLMK		TMLPR0, TMLPR1		√	√	√	√	√	√	√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	√	—	—	—	—	—	—	—
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		√	√	√	√	√	√	—	—	—	—

- Note 1.** If any of the interrupt sources INTSR1, INTCSI11, and INTIIC11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.
- Note 2.** Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt source INTSRE1 or INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.
- Note 3.** If any of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 4 of the IF1L register is set to 1. Bit 4 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Table 18 - 2 Flags Corresponding to Interrupt Request Sources (4/4)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	25-pin	24-pin	20-pin	16-pin
		Register		Register		Register										
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	√	√	√	√	√	√	—	—	—	—
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		√	√	√	√	√	√	—	—	—	—
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√	√	√	√	√	—	—	—	—
INTP6	PIF6		PMK6		PPR06, PPR16		√	—	—	—	—	—	—	—	—	—
INTP8	PIF8		PMK8		PPR08, PPR18		√	—	—	—	—	—	—	—	—	—
INTP9	PIF9		PMK9		PPR09, PPR19		√	—	—	—	—	—	—	—	—	—
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√	√	√	√	√	√	√	√
INTURE0	UREIF0	IF2H	UREK0	MK2H	UREPR00, UREPR10	PR02H, PR12H	√	√	√	√	—	—	—	—	—	—
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		√	√	√	√	√	√	√	√	√	√
INTCTSURD	CTSURDIF	IF3L	CTSURDMK	MK3L	CTSURDPR0, CTSURDPR1	PR03L, PR13L	√	√	√	√	√	√	√	√	√	√
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		√	√	√	√	√	√	√	√	√	√
INTUT0	UTIF0		UTMK0		UTPR00, UTPR10		√	√	√	√	—	—	—	—	—	—
INTUR0	URIF0		URMK0		URPR00, URPR10		√	√	√	√	—	—	—	—	—	—

18.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when the given interrupt request is acknowledged, a reset signal is generated, or an instruction is executed.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, and IF3L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, the IF2L and IF2H registers, and the IF3L register and the 8-bit fixed value 00H are combined to form 16-bit registers IF0, IF1, IF2, and IF3, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 00H.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 18 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (1/2)

Address: FFFE0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIF	WDTIF

Address: FFFE1H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF0H	SREIF0 TMIF01H	TMIF00	STIF0 CSIF00 IICIF00	SMSEIF	0	SREIF2	SRIF2 CSIF21 IICIF21	STIF2 CSIF20 IICIF20

Address: FFFE2H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	SRIF0 CSIF01 IICIF01	IICAF0	SREIF1 TMIF03H	SRIF1 CSIF11 IICIF11	STIF1

Address: FFFE3H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
IF1H	TMIF04	0	0	0	KRIF	ITLIF	RTCIF	ADIF

Figure 18 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (2/2)

Address: FFFD0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
IF2L	FLIF	PIF9	PIF8	0	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	4	3	<2>	1	0
IF2H	0	CTSUWRIF	0	0	0	UREIF0	0	0

Address: FFFD2H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	<3>	2	<1>	<0>
IF3L	0	0	0	URIF0	UTIF0	0	CTSUFNIF	CTSURDIF

xxIFx	Interrupt request flag
0	No interrupt request signal is generated
1	Indicates the generation of the interrupt request signal and the interrupt request currently being in the active state.

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 18 - 2. Be sure to set bits that are not available to the initial value.

Caution 2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L.0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

18.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, and MK3L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers, and the MK3L register and the 8-bit fixed value 00H are combined to form 16-bit registers MK0, MK1, MK2, and MK3, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is FFH.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 18 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L) (1/2)

Address: FFFE4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK0H	SREMK0 TMMK01H	TMMK00	STMK0 CSIMK00 IICMK00	SMSEMK	0	SREMK2	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20

Address: FFFE6H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	SRMK0 CSIMK01 IICMK01	IICAMK0	SREMK1 TMMK03H	SRMK1 CSIMK11 IICMK11	STMK1

Address: FFFE7H
After reset: FFH
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
MK1H	TMMK04	0	0	0	KRMK	ITLMK	RTCMK	ADMK

Figure 18 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L) (2/2)

Address: FFFD4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
MK2L	FLMK	PMK9	PMK8	0	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H
After reset: FFH
R/W: R/W

Symbol	7	<6>	5	4	3	<2>	1	0
MK2H	0	CTSUWRMK	0	0	0	UREMK0	0	0

Address: FFFD6H
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	2	<1>	<0>
MK3L	0	0	0	URMK0	UTMK0	0	CTSUFNMK	CTSURDMK

xxMKx	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 18 - 2. Be sure to set bits that are not available to the initial value.

18.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, 2H, or 3L).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR03L register and the 8-bit fixed value FFH, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers, and the PR13L register and the 8-bit fixed value FFH are combined to form 16-bit registers PR00, PR01, PR02, PR03, PR10, PR11, PR12, and PR13, they can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is FFH.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 18 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (1/4)

Address: FFFE8H

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR00H	SREPR00 TMPR001H	TMPR000	STPR00 CSIPR000 IICPR000	SMSEPR0	0	SREPR02	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020

Figure 18 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (2/4)

Address: FFFEDH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H	TMPR100	STPR10 CSIPR100 IICPR100	SMSEPR1	0	SREPR12	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120

Address: FFFEAH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	SRPR00 CSIPR001 IICPR001	IICAPR00	SREPR01 TMPR003H	SRPR01 CSIPR011 IICPR011	STPR01

Address: FFFEEH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	SRPR10 CSIPR101 IICPR101	IICAPR10	SREPR11 TMPR103H	SRPR11 CSIPR111 IICPR111	STPR11

Address: FFFEBH
After reset: FFH
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
PR01H	TMPR004	0	0	0	KRPR0	ITLPR0	RTCPR0	ADPR0

Address: FFFE FH
After reset: FFH
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
PR11H	TMPR104	0	0	0	KRPR1	ITLPR1	RTCPR1	ADPR1

Figure 18 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (3/4)

Address: FFFD8H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PR02L	FLPR0	PPR09	PPR08	0	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PR12L	FLPR1	PPR19	PPR18	0	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H
After reset: FFH
R/W: R/W

Symbol	7	<6>	5	4	3	<2>	1	0
PR02H	0	CTSUWRPR0	0	0	0	UREPR00	0	0

Address: FFFDDH
After reset: FFH
R/W: R/W

Symbol	7	<6>	5	4	3	<2>	1	0
PR12H	0	CTSUWRPR1	0	0	0	UREPR10	0	0

Address: FFFDAH
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	2	<1>	<0>
PR03L	0	0	0	URPR00	UTPR00	0	CTSUFNPR0	CTSURDPR0

Address: FFFDEH
After reset: FFH
R/W: R/W

Symbol	7	6	5	<4>	<3>	2	<1>	<0>
PR13L	0	0	0	URPR10	UTPR10	0	CTSUFNPR1	CTSURDPR1

Figure 18 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (4/4)

xxPR1x	xxPR0x	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 18 - 2. Be sure to set bits that are not available to the initial value.

18.3.4 External interrupt rising edge enable registers (EGP0, EGP1), External interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP11.

The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 18 - 5 Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	0	0	EGP9	EGP8

Address: FFF3BH

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6, 8, 9)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 18 - 3 shows the ports corresponding to the EGPn and EGNn bits.

Table 18 - 3 Interrupt Request Signals Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	48-pin	30-, 32-, 36-, 40-, and 44-pin	24- and 25-pin	20-pin	16-pin
EGP0	EGN0	INTP0	√	√	√	√	√
EGP1	EGN1	INTP1	√	√	√	—	—
EGP2	EGN2	INTP2	√	√	—	—	—
EGP3	EGN3	INTP3	√	√	√	√	√
EGP4	EGN4	INTP4	√	√	√	—	—
EGP5	EGN5	INTP5	√	√	√	√	—
EGP6	EGN6	INTP6	√	—	—	—	—
EGP8	EGN8	INTP8	√	—	—	—	—
EGP9	EGN9	INTP9	√	—	—	—	—

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark 1. For edge detection ports, see 2.1 Functions of Port Pins.

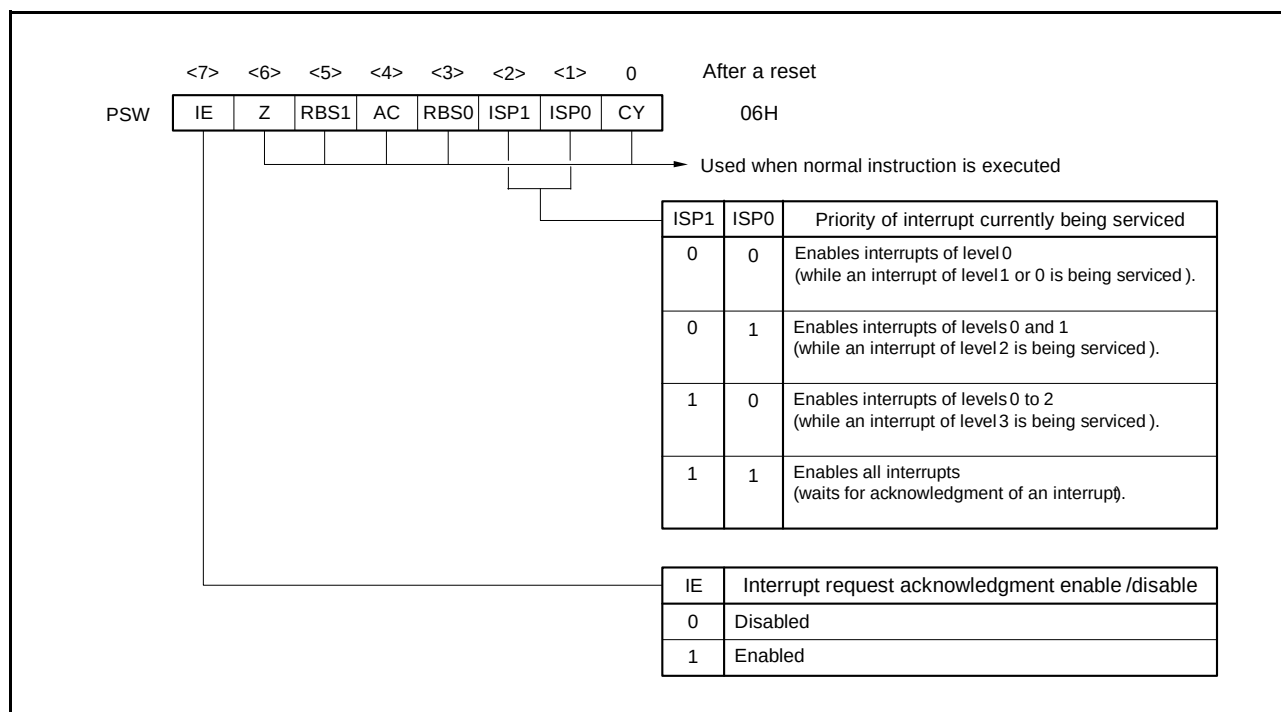
Remark 2. n = 0 to 6, 8, 9

18.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current state for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that control multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions. The value of the PSW following a reset is 06H.

Figure 18 - 6 Configuration of Program Status Word



18.4 Interrupt Servicing Operations

18.4.1 Acknowledgment of maskable interrupt requests

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

Table 18 - 4 lists the times taken from generation of a maskable interrupt request until execution of the vectored interrupt handler. See **Figure 18 - 8** and **Figure 18 - 9** for schematic views of the timing of interrupt acknowledgment.

Table 18 - 4 Times Taken from Generation of a Maskable Interrupt Request until Execution of the Vectored Interrupt Handler

	Minimum Time	Maximum Time ^{Note}
Time taken until acknowledgment of an interrupt request	9 clock cycles	16 clock cycles

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark One clock cycle is equivalent to $1/f_{CLK}$, where f_{CLK} is the frequency of the CPU clock.

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

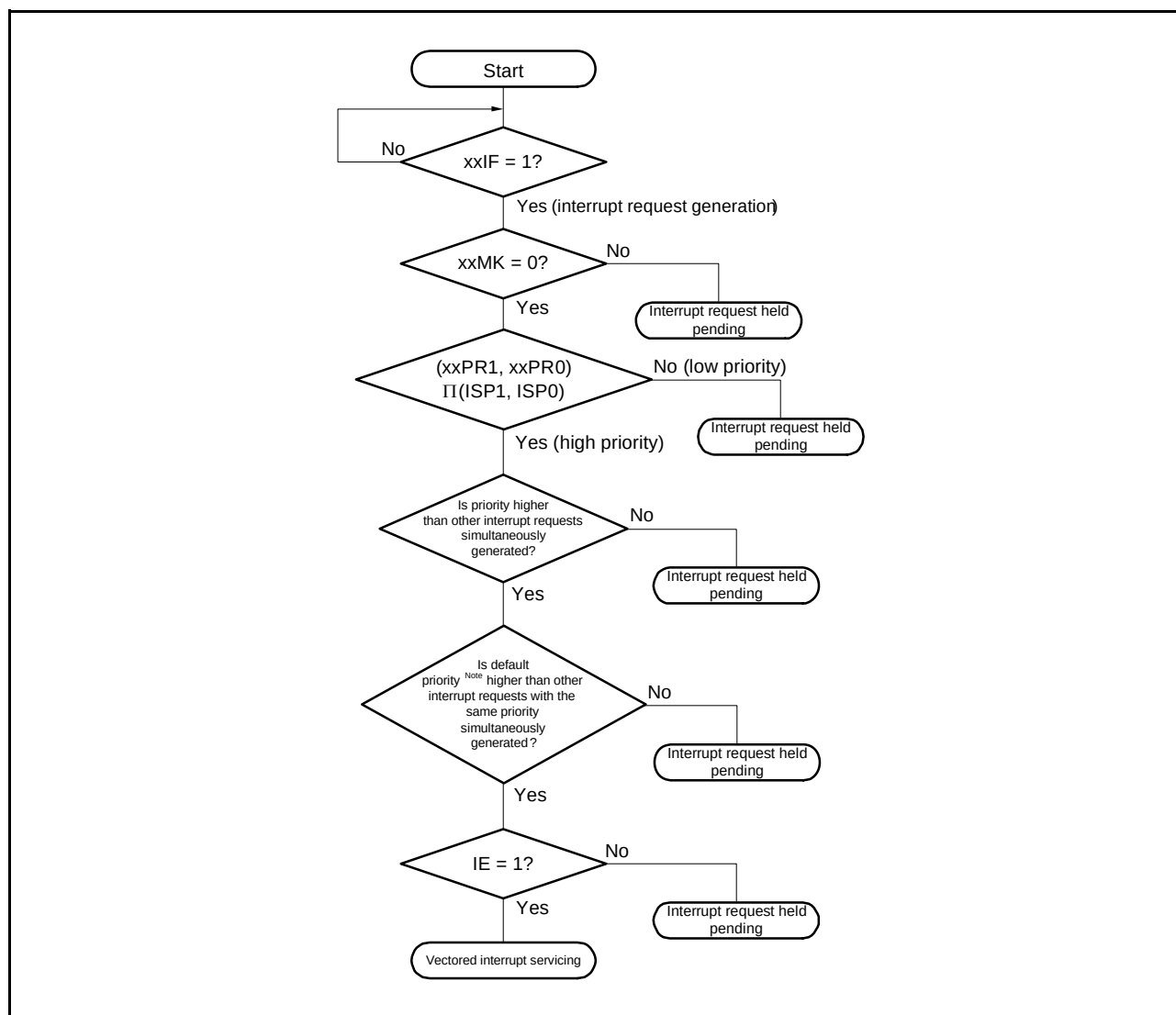
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18 - 7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is loaded into the PC and program control branches to the specified servicing.

Restoring from an interrupt is possible by using the RETI instruction.

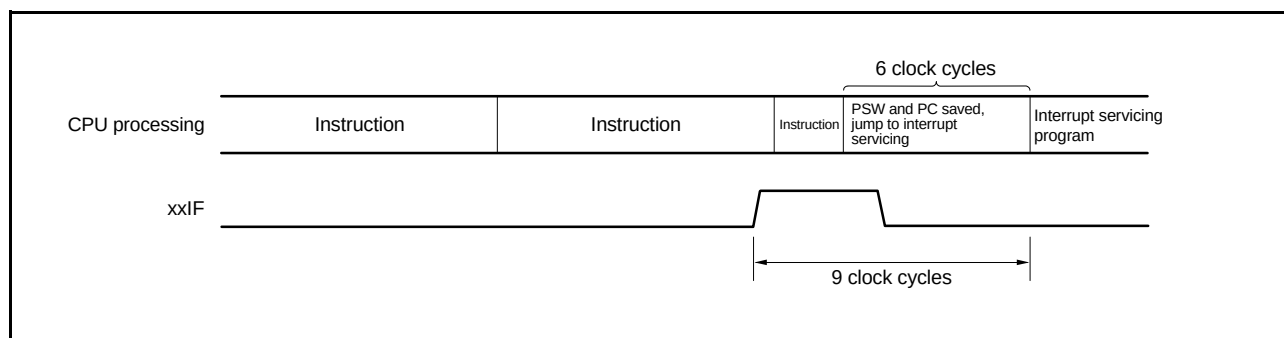
Figure 18 - 7 Interrupt Request Acknowledgment Processing Algorithm



Note For the default priority, refer to **Table 18 - 1 Interrupt Source List (1/3)**.

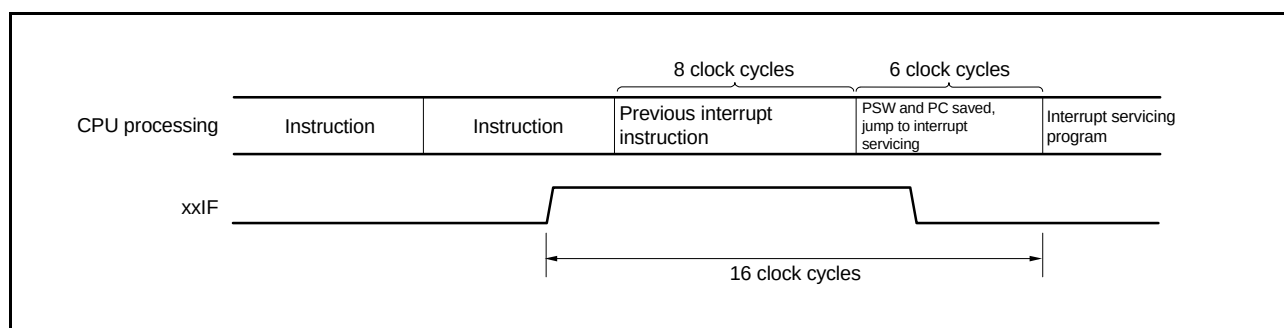
Remark	xxIF:	Interrupt request flag
	xxMK:	Interrupt mask flag
	xxPR0:	Priority specification flag 0
	xxPR1:	Priority specification flag 1
	IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
	ISP0, ISP1:	Flags that indicate the priority level of the interrupt currently being serviced (see Figure 18 - 6)

Figure 18 - 8 Minimum Time Taken until Acknowledgment of an Interrupt Request



Remark One clock cycle is equivalent to $1/f_{CLK}$, where f_{CLK} is the frequency of the CPU clock.

Figure 18 - 9 Maximum Time Taken until Acknowledgment of an Interrupt Request



Remark One clock cycle is equivalent to $1/f_{CLK}$, where f_{CLK} is the frequency of the CPU clock.

18.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled. If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), the contents of the vector table (0007EH, 0007FH) are loaded into the PC, and program control branches to the specified servicing. Restoring from a software interrupt is possible by using the RETB instruction.

Caution The RETI instruction cannot be used for restoring from the software interrupt.

18.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt. Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set the IE flag to 1 with the EI instruction during interrupt servicing to enable interrupt acknowledgment. Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing. In the interrupt enabled state, if an interrupt request with a priority higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction.

Table 18 - 5 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 18 - 10** shows multiple interrupt servicing examples.

Table 18 - 5 Relationship between Interrupt Requests Enabled for Multiple Interrupt Servicing during Interrupt Servicing

Multiple Interrupt Request Maskable Interrupt		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Processing in progress	ISP1 = 0 ISP0 = 0	√	—	—	—	—	—	—	—	√
	ISP1 = 0 ISP0 = 1	√	—	√	—	—	—	—	—	√
	ISP1 = 1 ISP0 = 0	√	—	√	—	√	—	—	—	√
Waiting for reception	ISP1 = 1 ISP0 = 1	√	—	√	—	√	—	√	—	√

Remark 1. √: Multiple interrupt servicing enabled

Remark 2. —: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for an interrupt acknowledgment (all interrupts are enabled).

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)

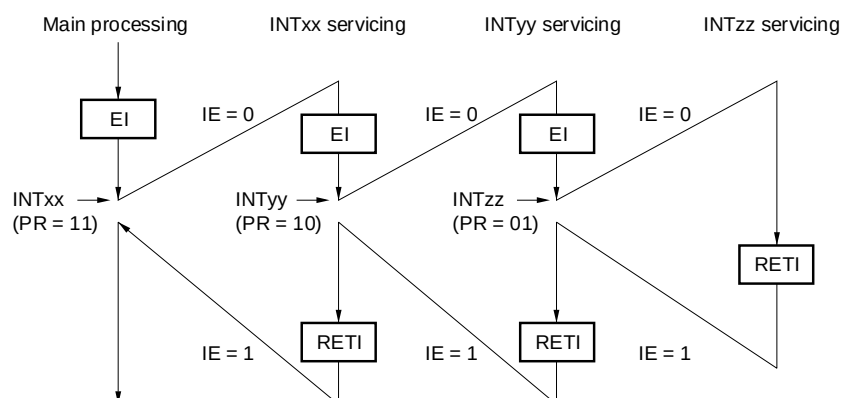
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1

PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0

PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

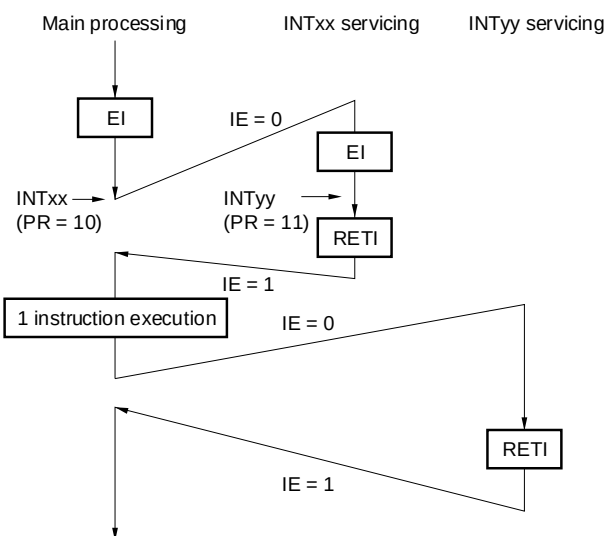
Figure 18 - 10 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control



Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)

PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1

PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0

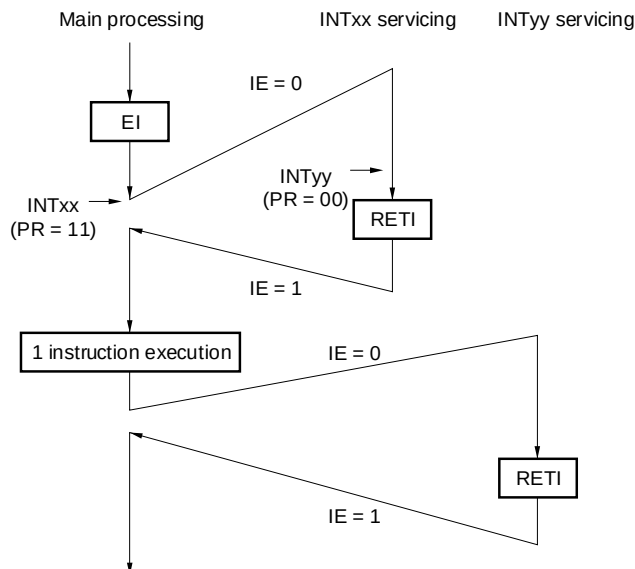
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

Figure 18 - 10 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)

PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1

PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0

PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled

IE = 1: Interrupt request acknowledgment is enabled.

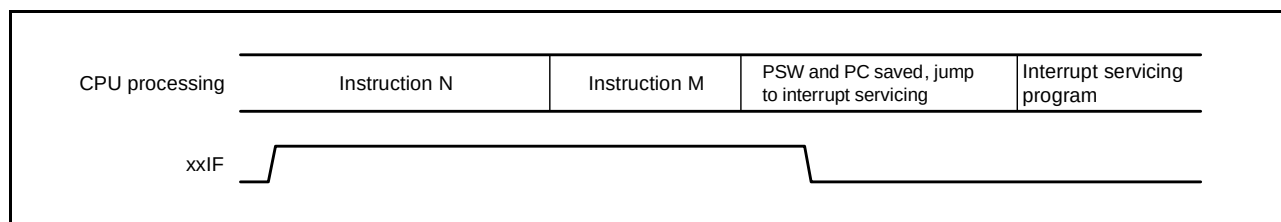
18.4.4 Interrupt request held pending

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (instructions that hold interrupt requests pending) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

Figure 18 - 11 shows the timing at which interrupt requests are held pending.

Figure 18 - 11 Holding Interrupt Requests Pending



Remark 1. Instruction N: Instruction to hold interrupt requests pending

Remark 2. Instruction M: Instruction other than the instruction to hold interrupt requests pending

Section 19 Key Interrupt Function

The number of key interrupt input channels differs, depending on the product.

	16-, 20-, 24-, 25-, 30-, 32-, and 36-pin products	40- and 44-pin products	48-pin products
Number of the key interrupt input channels	—	4	6

19.1 Functions of the Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR5).

Table 19 - 1 Assignment of the Key Interrupt Detection Pins

Key interrupt pins	Key return mode register 0 (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05

Remark Pins KR0 to KR3 are only present in the 40- and 44-pin products.
Pins KR0 to KR5 are only present in the 48-pin products.

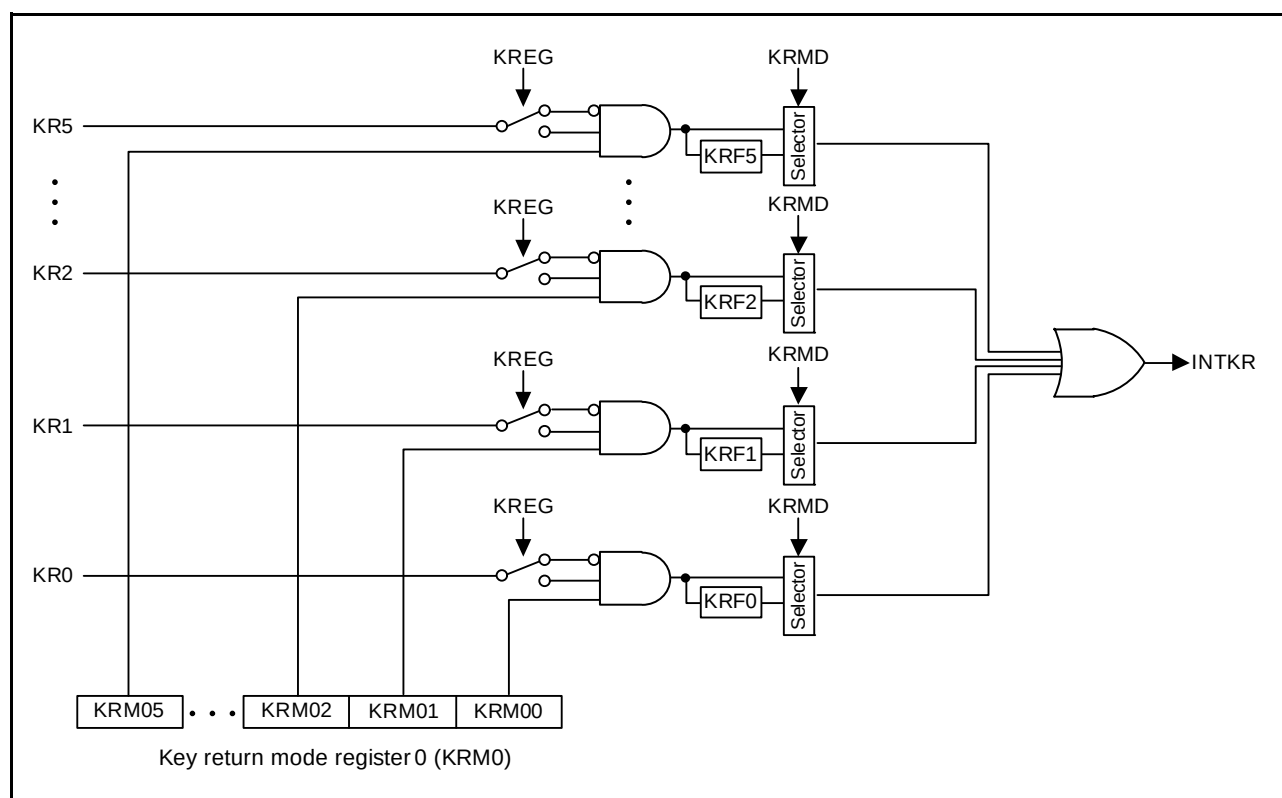
19.2 Configuration of the Key Interrupt

The key interrupt includes the following hardware blocks.

Table 19 - 2 Configuration of the Key Interrupt

Item	Configuration
Control registers	Key return control register (KRCTL) Key return mode register 0 (KRM0) Key return flag register (KRF) Port mode register 7 (PM7)

Figure 19 - 1 Block Diagram of the Key Interrupt



Remark Pins KR0 to KR3 are only present in the 40- and 44-pin products.
Pins KR0 to KR5 are only present in the 48-pin products.

19.3 Registers for Controlling the Key Interrupt

The following registers are used to control the key interrupt.

- Key return control register (KRCTL)
- Key return mode register 0 (KRM0)
- Key return flag register (KRF)
- Port mode register 7 (PM7)

19.3.1 Key return control register (KRCTL)

The KRCTL register controls the usage of the key return flags (KRF0 to KRF5) and sets the detection edge.

The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 19 - 2 Format of Key Return Control Register (KRCTL)

Address: FFF34H

After reset: 00H

R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
KRCTL	KRMD	0	0	0	0	0	0	KREG
KRMD		Usage of key return flags (KRF0 to KRF5)						
0		Does not use key return flags						
1		Uses key return flags						
KREG		Selection of detection edge (KR0 to KR5)						
0		Falling edge						
1		Rising edge						

19.3.2 Key return mode register 0 (KRM0)

The KRM0 register controls the KR0 to KR5 signals.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 19 - 3 Format of Key Return Mode Register 0 (KRM0)

Address: FFF37H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. The on-chip pull-up resistor can be applied by setting the corresponding key interrupt input pins (bits) in pull-up resistor register 7 (PU7) to 1.

Caution 2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (KREG is set to 0) or a high level (KREG is set to 1) is being input to the key interrupt input pin.

To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input high-level and low-level widths. See 34.4 AC Characteristics.

Caution 3. The pins not used in the key interrupt mode can be used as normal port pins.

Remark 1. n = 0 to 5

Remark 2. Pins KR0 to KR3 are only present in the 40- and 44-pin products.

Pins KR0 to KR5 are only present in the 48-pin products.

19.3.3 Key return flag register (KRF)

The KRF register controls the key interrupt flags (KRF0 to KRF5).

The KRF register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 19 - 4 Format of Key Return Flag Register (KRF)

Address: FFF35H

After reset: 00H

R/W: R/WNote

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 5)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing 1 has no effect. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

19.3.4 Port mode register 7 (PM7)

The PM7 register sets the input and output of port 7 in 1-bit units.

To use a key interrupt input (KR0 to KR5), set the corresponding bit in this register to 1.

The PM7 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is FFH.

The on-chip pull-up resistor can be applied by setting the corresponding key interrupt input pins (bits) in the pull-up resistor register 7 (PU7) to 1.

Figure 19 - 5 Format of Port Mode Register 7 (PM7)

Address: FFF27H

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PM7	0	0	PM75	PM74	PM73	PM72	PM71	PM70
	PM7n		I/O mode selection for PM7n pin (n = 0 to 5)					
	0		Output mode (output buffer on)					
	1		Input mode (output buffer off)					

Section 20 Standby Function

20.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

Executing a HALT instruction places this LSI chip in the HALT mode. In the HALT mode, the CPU operating clock is stopped. If the high-speed system clock oscillator, high-speed on-chip oscillator, middle-speed on-chip oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

Executing a STOP instruction places this LSI chip in the STOP mode. In the STOP mode, the high-speed system clock oscillator, high-speed on-chip oscillator, and middle-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be released by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In SNOOZE mode, the RL78/G22 is released from STOP mode and the following peripheral modules can operate without CPU intervention.

For details, see the sections on the individual modules.

- **Section 12 A/D Converter (ADC)**
- **Section 13 Serial Array Unit (SAU)**
- **Section 16 Data Transfer Controller (DTC)**
- **Section 26 SNOOZE Mode Sequencer (SMS)**
- **Section 27 Capacitive Sensing Unit (CTS2La)**

This can only be specified when the high-speed on-chip oscillator or middle-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (fCLK).

In any of these modes, registers, flags, and data memory retain all values they had immediately before the transition to the standby mode. The states of the output latches and the output buffers for I/O port pins are also retained.

(Cautions are listed on the next page.)

- Caution 1.** Shifting to the STOP mode is only enabled when the CPU is operating on the main system clock. Do not execute the STOP instruction while the CPU operates with the subsystem clock. Shifting to the HALT mode is enabled whether the CPU is operating on the main system clock or the subsystem clock.
- Caution 2.** When shifting to the STOP mode, be sure to stop the peripheral modules operating with the main system clock before executing the STOP instruction. Note that this does not apply to the peripheral modules to be used in the SNOOZE mode; that is, operations of these modules should not be stopped.
- Caution 3.** It can be selected by WDTON of the option byte or WUTMMCK0 of the subsystem clock supply mode control register (OSMC) whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see 6.1 Functions of Clock Generator (2) Subsystem clock <2> Low-speed on-chip oscillator.

20.2 Registers for Controlling the Standby Function

The following registers are used to control the standby function.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Standby mode release setting register (WKUPMD)

Remark For details of the OSMC, OSTC, and OSTS registers, see **Section 6 Clock Generator**. For registers which control the SNOOZE mode function, see the following sections.

- **Section 12 A/D Converter (ADC)**
- **Section 13 Serial Array Unit (SAU)**
- **Section 16 Data Transfer Controller (DTC)**
- **Section 26 SNOOZE Mode Sequencer (SMS)**
- **Section 27 Capacitive Sensing Unit (CTSU2La)**

20.2.1 Standby mode release setting register (WKUPMD)

The WKUPMD register is used to set the operation at the time of release from the standby mode.

The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 20 - 1 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode ^{Notes 1, 2}
0	Starting of the high-speed on-chip oscillator is at normal speed. ^{Note 3}
1	Starting of the high-speed on-chip oscillator is at high speed. ^{Note 3}

Note 1. This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.

Note 2. This register is initialized when the RL78/G22 is released from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.

Note 3. For the activation time, see **20.3.2 STOP mode**.

The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 34 Electrical Characteristics for TA = -40 to +105°C**.

20.3 Standby Function Operation

20.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock.

The operating states in the HALT mode are shown below.

Caution The interrupt request signal is used for release from the HALT mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), the HALT mode is not entered if these are the settings even when a HALT instruction is executed.

Table 20 - 1 Operating States in HALT Mode (1) (1/2)

Item \ HALT Mode Setting			When HALT Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f _{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{IM})	When CPU is Operating on X1 Clock (f _X)	When CPU is Operating on External Main System Clock (f _{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f _{IH}	Operation continues (cannot be stopped)	Operation disabled	Operation disabled	
		f _{IM}	Operation disabled	Operation continues (cannot be stopped)	Operation disabled	
		f _X	Operation disabled		Operation continues (cannot be stopped)	Cannot operate
		f _{EX}			Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f _{XT}	Retains the state before the transition to HALT mode			
		f _{EXS}				
		f _{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f _{XS}) and XR (f _{XR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM			Operation stopped (capable of operations in response to access by the DTC or SMS)			
Port (latch)			Retains the state before the transition to HALT mode (capable of operations in response to access by the DTC, SMS or ELC)			
Timer array unit			Operation enabled			
RTC			Operation enabled			
32-bit interval timer			Operation enabled			
Watchdog timer			See Section 11 Watchdog Timer (WDT) .			

Table 20 - 1 Operating States in HALT Mode (1) (2/2)

HALT Mode Setting Item		When HALT Instruction is Executed While CPU is Operating on Main System Clock			
		When CPU is Operating on High-speed On-chip Oscillator Clock (f _{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{IM})	When CPU is Operating on X1 Clock (f _X)	When CPU is Operating on External Main System Clock (f _{EX})
Clock output/buzzer output		Operation enabled			
A/D converter					
Serial array unit					
Serial interface IICA					
Serial interface UARTA					
Data transfer controller (DTC)					
SNOOZE mode sequencer					
Event link controller (ELC)		Operation-enabled function blocks can be linked			
Power-on-reset function		Operation enabled			
Voltage detection function					
External interrupt					
Key interrupt function					
Capacitive sensing unit (CTS2La)					
CRC operation function	High-speed CRC	Capable of operations in response to access by the DTC or SMS to obtain data for calculations from the RAM area			
	General-purpose CRC				
Illicit memory access detection function		Capable of operations in response to access by the DTC or SMS			
RAM parity error detection function					
RAM guard function					
SFR guard function					
True random number generator		Operation enabled			

Remark Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.

Operation disabled: Stop operation before switching to the HALT mode.

f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock

f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock

f_{EX}: External main system clock f_{XT}: XT1 clock

f_{EXS}: External subsystem clock

Table 20 - 2 Operating States in HALT Mode (2) (1/2)

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock		
			When CPU is Operating on XT1 Clock (f _{XT})	When CPU is Operating on External Subsystem Clock (f _{EXS})	When CPU is Operating on Low-speed On-chip Oscillator Clock (f _{IL})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f _{IH}	Operation disabled		
		f _{IM}			
		f _X			
		f _{EX}			
	Subsystem clock	f _{XT}	Operation continues (cannot be stopped)	Cannot operate	Operation disabled
		f _{EXS}	Cannot operate	Operation continues (cannot be stopped)	Operation disabled
		f _{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f _{XS}) and XR (f _{XSXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped		Operation continues (cannot be stopped)
CPU		Operation stopped			
Code flash memory					
Data flash memory					
RAM			Operation stopped (capable of operations in response to access by the DTC or SMS)		
Port (latch)			Retains the state before the transition to HALT mode (capable of operations in response to access by the DTC, SMS or ELC)		
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
RTC			Operation enabled		
32-bit interval timer			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Watchdog timer			See Section 11 Watchdog Timer (WDT) .		
Clock output/buzzer output			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
A/D converter			Operation disabled		
Serial array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Serial interface IICA			Operation disabled		
Serial interface UARTA			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Data transfer controller (DTC)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled

Table 20 - 2 Operating States in HALT Mode (2) (2/2)

HALT Mode Setting Item		When HALT Instruction is Executed While CPU is Operating on Subsystem Clock		
		When CPU is Operating on XT1 Clock (fxT)	When CPU is Operating on External Subsystem Clock (fexs)	When CPU is Operating on Low-speed On-chip Oscillator Clock (fil)
SNOOZE mode sequencer		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Event link controller (ELC)		Operation-enabled function blocks can be linked		
Power-on-reset function		Operation enabled		
Voltage detection function				
External interrupt				
Key interrupt function				
Capacitive sensing unit (CTSU2La)		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
CRC operation function	High-speed CRC	Operation disabled		
	General-purpose CRC	Capable of operations in response to access by the DTC or SMS to obtain data for calculations from the RAM area		
Illicit memory access detection function		Capable of operations in response to access by the DTC or SMS		
RAM parity error detection function				
RAM guard function				
SFR guard function				
True random number generator		Operation enabled		

Remark Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.
Operation disabled: Stop operation before switching to the HALT mode.

fIH: High-speed on-chip oscillator clock fIL: Low-speed on-chip oscillator clock
fIM: Middle-speed on-chip oscillator clock fx: X1 clock
fEX: External main system clock fxT: XT1 clock
fEXS: External subsystem clock

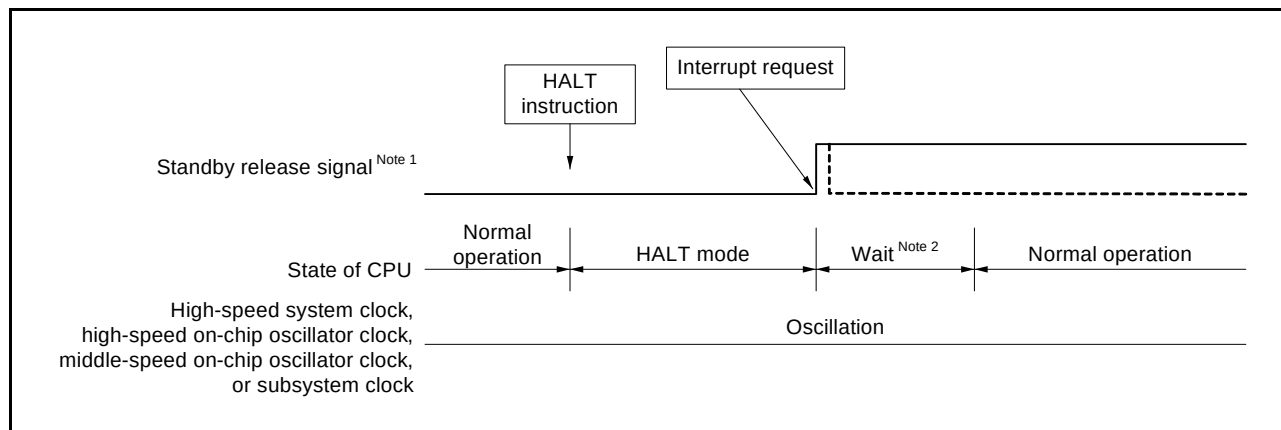
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by a non-masked interrupt request

When a non-masked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the instruction at the next address is executed.

Figure 20 - 2 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 18 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out

Main system clock: 15 to 16 clock cycles

Subsystem clock (RTCLPC = 0): 10 to 11 clock cycles

Subsystem clock (RTCLPC = 1): 11 to 12 clock cycles

- When vectored interrupt servicing is not carried out

Main system clock: 9 to 10 clock cycles

Subsystem clock (RTCLPC = 0): 4 to 5 clock cycles

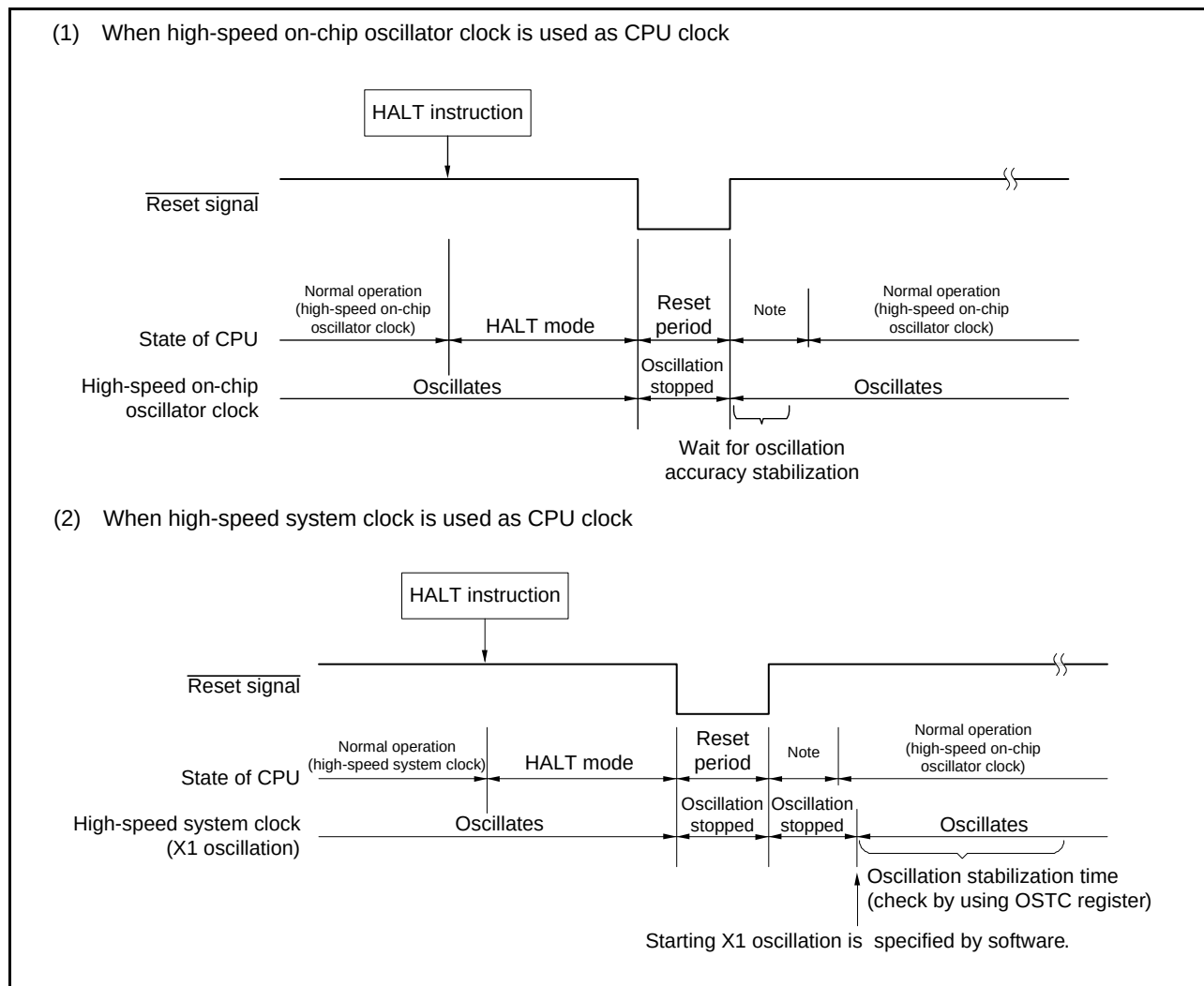
Subsystem clock (RTCLPC = 1): 5 to 6 clock cycles

Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

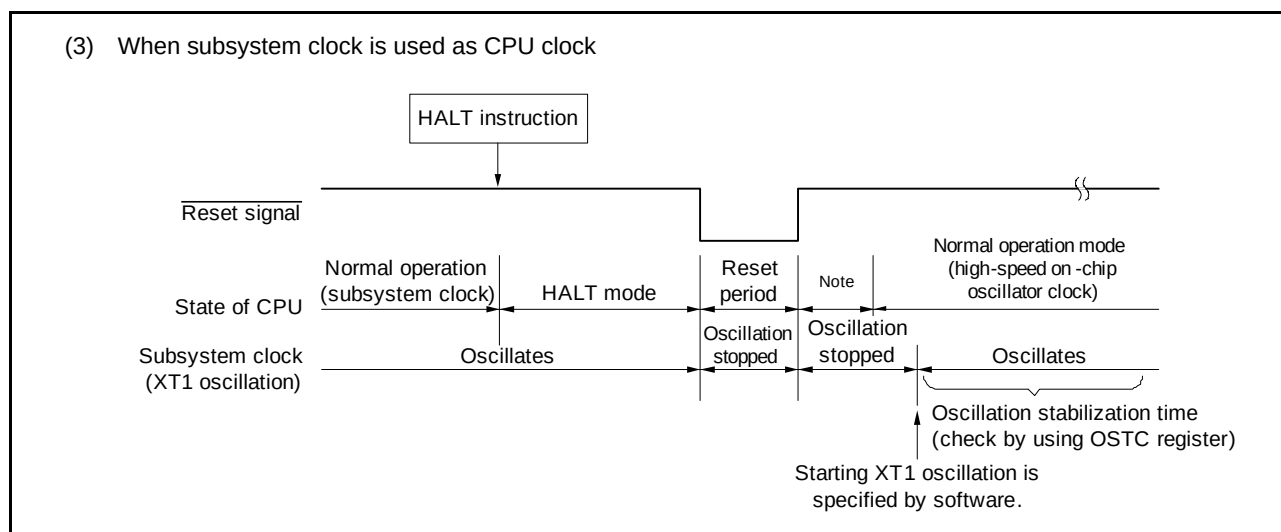
Figure 20 - 3 HALT Mode Release by Reset (1/2)



Note For the reset processing time, see **Section 21 Reset Function**.

For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 22 Power-on-Reset Circuit (POR)**.

Figure 20 - 3 HALT Mode Release by Reset (2/2)



Note For the reset processing time, see **Section 21 Reset Function**.

For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 22 Power-on-Reset Circuit (POR)**.

20.3.2 STOP mode

(1) STOP mode setting and operating states

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

Caution The interrupt request signal is used for release from the STOP mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), release from the STOP mode immediately proceeds after the transition to the STOP mode if these are the settings when a STOP instruction is executed. Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

The operating states in the STOP mode are shown below.

Table 20 - 3 Operating States in STOP Mode (1/2)

STOP Mode Setting Item			When STOP Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f _{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{IM})	When CPU is Operating on X1 Clock (f _X)	When CPU is Operating on External Main System Clock (f _{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f _{IH}	Stopped			
		f _{IM}	Stopped	Stopped	Stopped	
		f _X	Stopped			
		f _{EX}				
	Subsystem clock	f _{XT}	Retains the state before the transition to STOP mode			
		f _{EXS}				
f _{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f _{sx}) and XR (f _{sXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped				
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM						
Port (latch)			Retains the state before the transition to STOP mode			
Timer array unit			Operation stopped			
RTC			Operation enabled			
32-bit interval timer			Capable of operation when f _{sXP} is selected and RTCLPC = 0			
Watchdog timer			See Section 11 Watchdog Timer (WDT) .			
Clock output/buzzer output			Capable of operation when f _{sXP} is selected and RTCLPC = 0			
A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)			
Serial array unit			Wakeup operation is enabled only for CSIp and UARTq (switching to SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq			
Serial interface IICA			Capable of waking up in response to address matching			
Serial interface UARTA			Capable of operation when f _{sXP} is selected and RTCLPC = 0			
Data transfer controller (DTC)			DTC activation source receiving operation enabled (switching to SNOOZE mode)			
SNOOZE mode sequencer			SMS activation source receiving operation enabled (switching to SNOOZE mode)			
Event link controller (ELC)			Operation-enabled function blocks can be linked			
Power-on-reset function			Operation enabled			
Voltage detection function						
External interrupt						
Key interrupt function						
Capacitive sensing unit (CTS2La)			CTS2La activation source receiving operation enabled (switching to SNOOZE mode)			

Table 20 - 3 Operating States in STOP Mode (2/2)

STOP Mode Setting		When STOP Instruction is Executed While CPU is Operating on Main System Clock			
		When CPU is Operating on High-speed On-chip Oscillator Clock (f _{ih})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{im})	When CPU is Operating on X1 Clock (f _x)	When CPU is Operating on External Main System Clock (f _{ex})
Item	CRC operation function	Operation stopped			
	High-speed CRC				
	General-purpose CRC				
	Illicit memory access detection function				
	RAM parity error detection function				
	RAM guard function				
	SFR guard function				
	True random number generator				

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.

Operation disabled: Stop operation before switching to the STOP mode.

f_{ih}: High-speed on-chip oscillator clock f_{il}: Low-speed on-chip oscillator clock

f_{im}: Middle-speed on-chip oscillator clock f_x: X1 clock

f_{ex}: External main system clock f_{XT}: XT1 clock

f_{EXS}: External subsystem clock f_{SXP}: Low-speed peripheral clock frequency

Remark 2. p = 00; q = 0

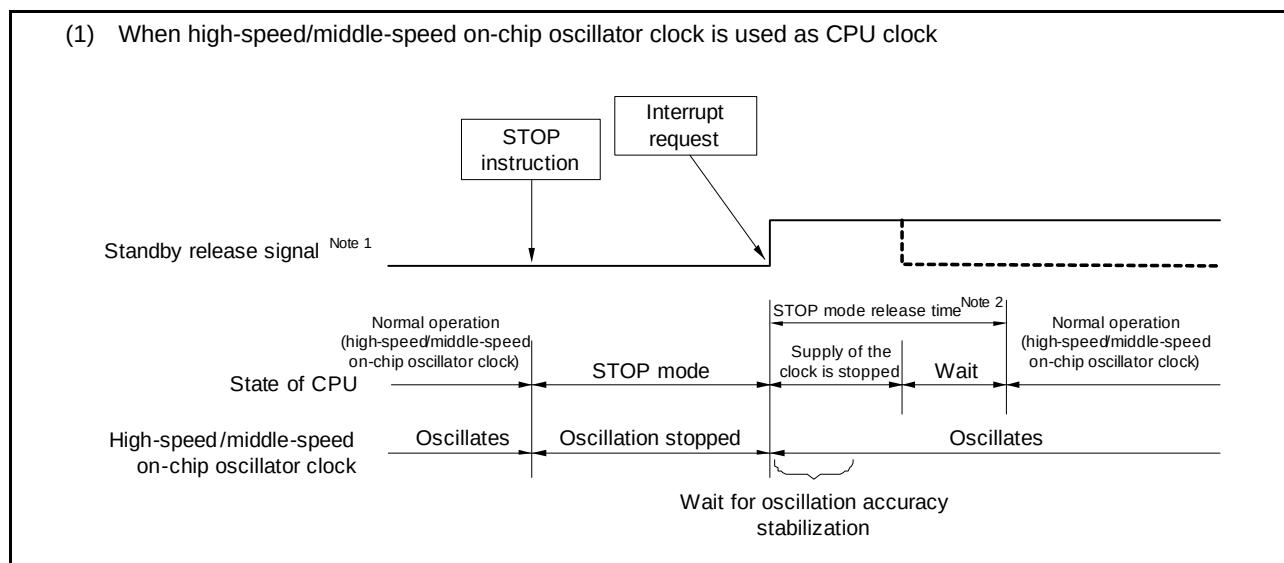
(2) STOP mode release

The STOP mode can be released by the following two sources.

(a) Release by a non-masked interrupt request

When a non-masked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the instruction at the next address is executed.

Figure 20 - 4 STOP Mode Release by Interrupt Request Generation (1/3)



(Notes, Caution, and Remarks are listed on the next page.)

Note 1. For details of the standby release signal, see **Figure 18 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

Supply of the clock is stopped:

For the high-speed on-chip oscillator clock: 3.9 to 5.2 μ s (FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)

0.6 to 0.8 μ s (FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)

The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 34 Electrical Characteristics for Ta = -40 to +105°C**.

For the middle-speed on-chip oscillator clock: 1.5 to 2.5 μ s

Wait:

(common to the high-speed/middle-speed on-chip oscillator clock)

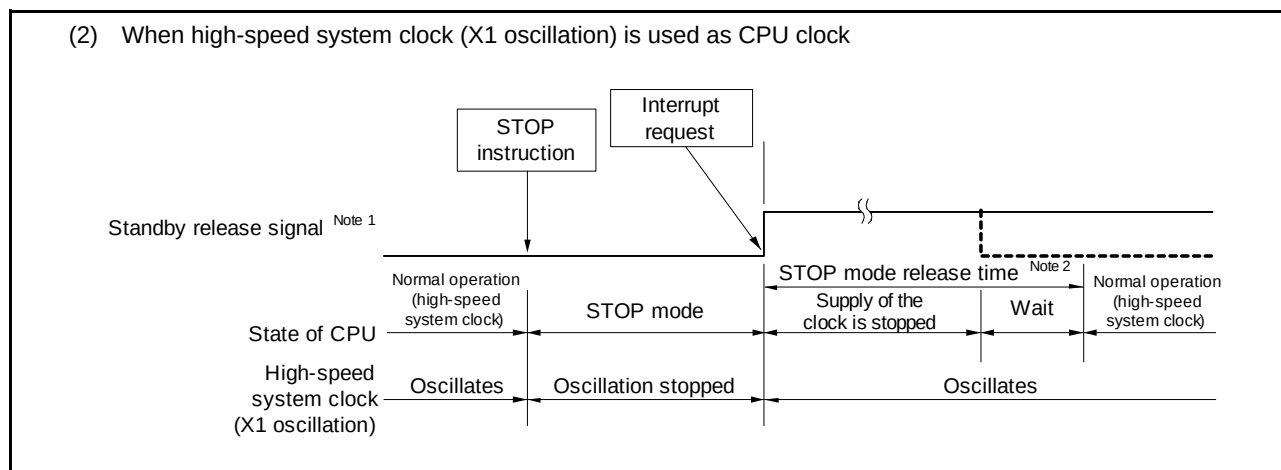
- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 20 - 4 STOP Mode Release by Interrupt Request Generation (2/3)



Note 1. For details of the standby release signal, see **Figure 18 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

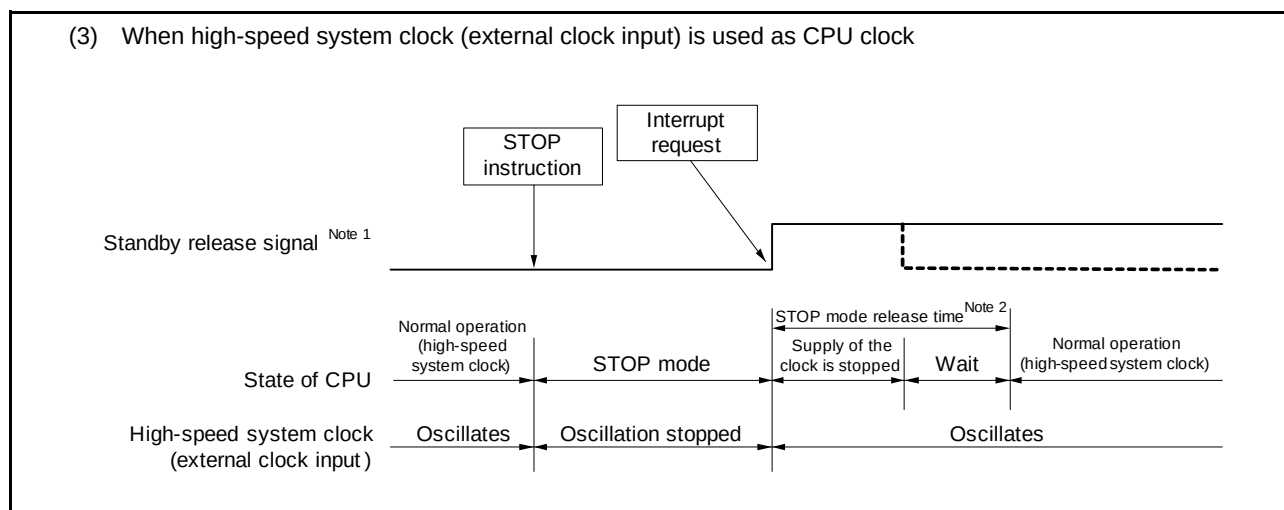
Supply of the clock is stopped:

Oscillation stabilization time (set by OSTS)

Wait:

- When vectored interrupt servicing is carried out: 10 to 11 clock cycles
- When vectored interrupt servicing is not carried out: 4 to 5 clock cycles

Figure 20 - 4 STOP Mode Release by Interrupt Request Generation (3/3)



Note 1. For details of the standby release signal, see **Figure 18 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

Supply of the clock is stopped:

50 to 51 cycles of the external clock

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the CPU clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

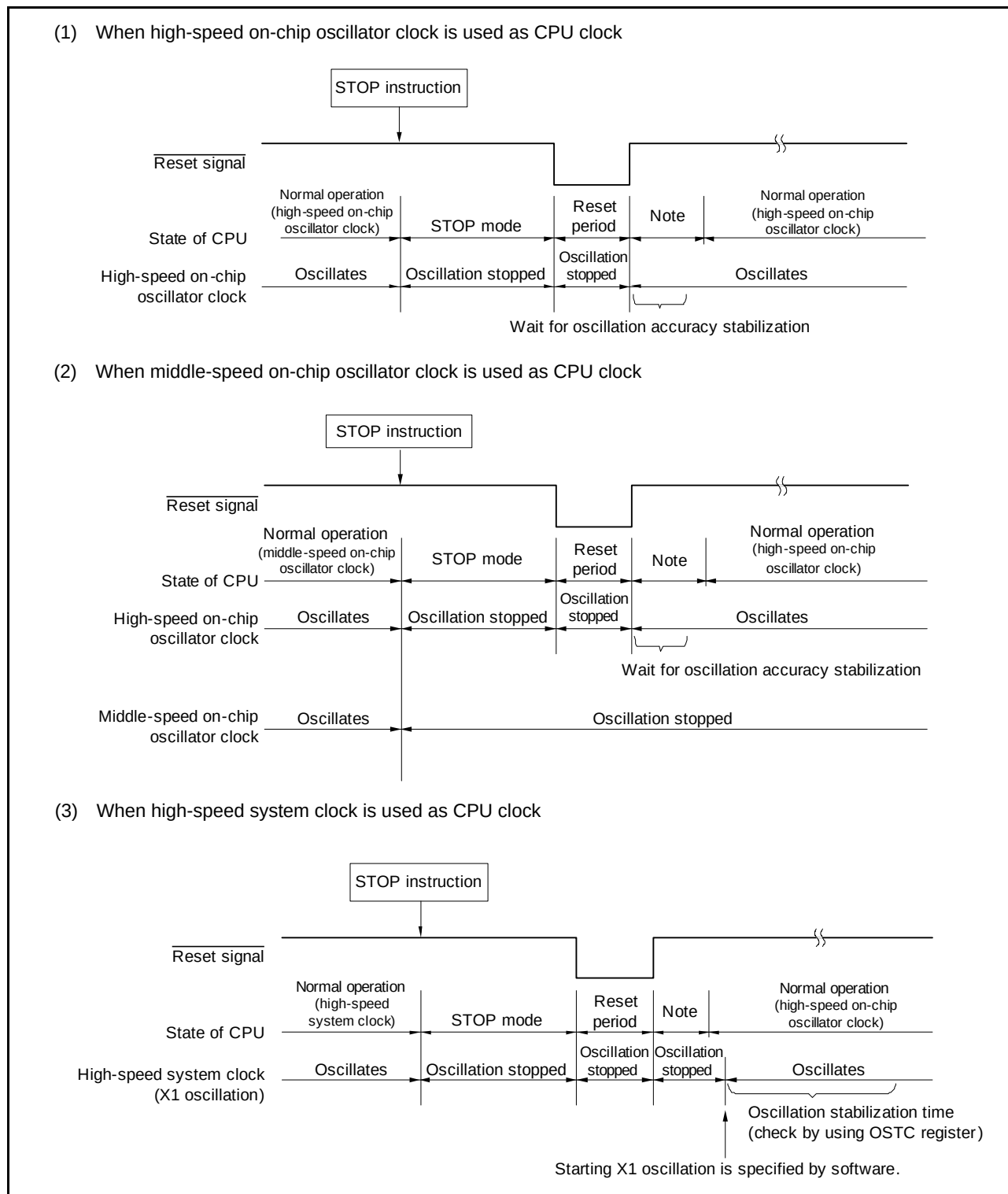
Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20 - 5 STOP Mode Release by Reset



20.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating states

The RL78/G22 can be placed in SNOOZE mode, in which operation of the following peripheral modules is selectable.

For details, see the sections on the individual modules.

- **Section 12 A/D Converter (ADC)**
- **Section 13 Serial Array Unit (SAU)**
- **Section 16 Data Transfer Controller (DTC)**
- **Section 26 SNOOZE Mode Sequencer (SMS)**
- **Section 27 Capacitive Sensing Unit (CTS2La)**

Also, the RL78/G22 can be placed in SNOOZE mode if the CPU clock before entry to SNOOZE mode is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock.

For transitions to SNOOZE mode, the following intervals of waiting are inserted.

Transition time from STOP mode to SNOOZE mode:

- For the high-speed on-chip oscillator clock: 3.9 to 5.2 μ s (FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)
0.6 to 0.8 μ s (FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)
The accuracy of the high-speed on-chip oscillator's frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 34 Electrical Characteristics for $T_A = -40$ to $+105^\circ\text{C}$.**

For the middle-speed on-chip oscillator clock^{Note}: 1.3 to 2.5 μ s

Remark The transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

For the high-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
"0.3 to 0.4 μ s" + 7 clock cycles
- When vectored interrupt servicing is not carried out:
"0.3 to 0.4 μ s" + 1 clock cycle

For the middle-speed on-chip oscillator clock^{Note}:

- When vectored interrupt servicing is carried out:
"0.6 to 1.2 μ s" + 7 clock cycles
- When vectored interrupt servicing is not carried out:
"0.6 to 1.2 μ s" + 1 clock cycle

Note This is selected when the setting of the MIOTRM register is its initial value.

The operating states in the SNOOZE mode are shown next.

Table 20 - 4 Operating States in SNOOZE Mode (1/2)

STOP Mode Setting Item			Generation of Source Conditions Which Lead to Transitions to SNOOZE Mode during STOP Mode	
			When CPU is Operating on High-speed On-chip Oscillator Clock (f _{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{IM})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f _{IH}	Operation started	Stopped
		f _{IM}	Stopped	Operation started
		f _X	Stopped	
		f _{EX}		
	Subsystem clock	f _{XT}	Operation enabled	
		f _{EXS}		
		f _{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f _{sx}) and XR (f _{sXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped	
	CPU			Operation stopped
Code flash memory				
Data flash memory				
RAM			Operation stopped (capable of operations in response to access by the DTC or SMS)	
Port (latch)			Retains the state before the transition to SNOOZE mode (capable of operations in response to access by the DTC, SMS or ELC)	
Timer array unit			Capable of operations in response to access by the DTC, SMS or ELC	
RTC			Operation enabled	
32-bit interval timer			Capable of operation when f _{sXP} is selected and RTCLPC = 0	
Watchdog timer			See Section 11 Watchdog Timer (WDT) .	
Clock output/buzzer output			Capable of operation when f _{sXP} is selected and RTCLPC = 0	
A/D converter			Operation enabled	
Serial array unit			Only CS _{IP} and USRT _q are capable of operation. Operation of the types of modules other than CS _{IP} and USRT _q is disabled.	
Serial interface IICA			Capable of waking up in response to address matching	
Serial interface UARTA			Capable of operation when f _{sXP} is selected and RTCLPC = 0	
Data transfer controller (DTC)			Operation enabled	
SNOOZE mode sequencer			Operation enabled	
Event link controller (ELC)			Operation-enabled function blocks can be linked	
Power-on-reset function			Operation enabled	
Voltage detection function				
External interrupt				
Key interrupt function				
Capacitive sensing unit (CTS _{U2La})			Operation enabled	
CRC operation function	High-speed CRC		Operation stopped	
	General-purpose CRC		Capable of operations in response to access by the DTC or SMS to obtain data for calculations from the RAM area	

Table 20 - 4 Operating States in SNOOZE Mode (2/2)

STOP Mode Setting Item	Generation of Source Conditions Which Lead to Transitions to SNOOZE Mode during STOP Mode	
	When CPU is Operating on High-speed On-chip Oscillator Clock (f _{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f _{IM})
Illicit memory access detection function	Capable of operations in response to access by the DTC or SMS	
RAM parity error detection function		
RAM guard function		
SFR guard function		
True random number generator		

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.

Operation disabled: Stop operation before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock

f_{IL}: Low-speed on-chip oscillator clock

f_{IM}: Middle-speed on-chip oscillator clock

f_X: X1 clock

f_{EX}: External main system clock

f_{XT}: XT1 clock

f_{EXS}: External subsystem clock

f_{SX}: Subsystem clock X

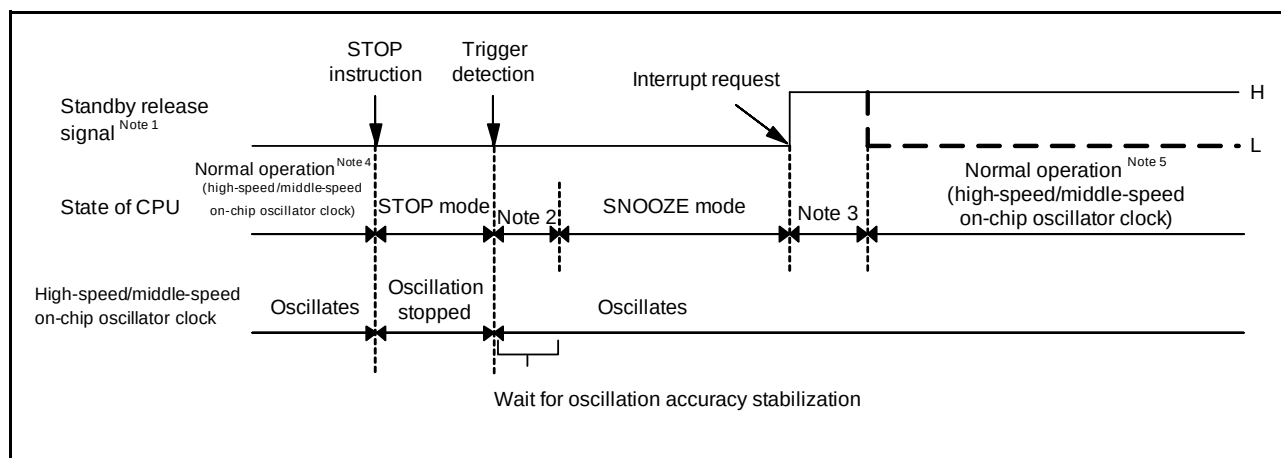
f_{SXR}: Subsystem clock XR

f_{SXP}: Low-speed peripheral clock frequency

Remark 2. p = 00; q = 0

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 20 - 6 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 18 - 1**.

Note 2. Transition time from STOP mode to SNOOZE mode

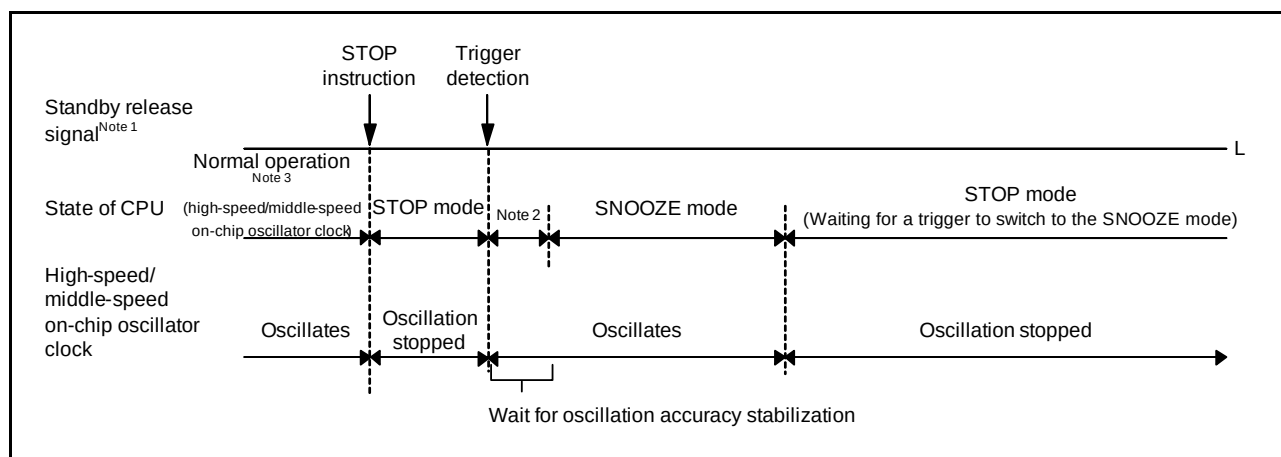
Note 3. Transition time from SNOOZE mode to normal operation

Note 4. Enable the SNOOZE mode immediately before switching to the STOP mode.

Note 5. Be sure to release the SNOOZE mode immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 20 - 7 When the Interrupt Request Signal is Not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 18 - 1**.

Note 2. Transition time from STOP mode to SNOOZE mode

Note 3. Enable the SNOOZE mode immediately before switching to the STOP mode.

Remark For details on the SNOOZE mode function, see the following sections.

- **Section 12 A/D Converter (ADC)**
- **Section 13 Serial Array Unit (SAU)**
- **Section 16 Data Transfer Controller (DTC)**
- **Section 26 SNOOZE Mode Sequencer (SMS)**
- **Section 27 Capacitive Sensing Unit (CTS2La)**

Section 21 Reset Function

A reset is triggered by any of the following events.

- (1) External reset input via the $\overline{\text{RESET}}$ pin
- (2) Internal reset due to detection of a program malfunction by the watchdog timer
- (3) Internal reset by comparison of supply voltage and detection voltage of the power-on-reset (POR) circuit
- (4) Internal reset by comparison of supply voltage and detection voltage of the voltage detectors (LVD0 and LVD1)
- (5) Internal reset due to execution of an illegal instruction^{Note}
- (6) Internal reset due to a RAM parity error
- (7) Internal reset due to illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated. A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, or on voltage detection by the POR, LVD0, or LVD1 circuit, execution of illegal instruction^{Note}, generation of a RAM parity error, or illegal-memory access, and each module is set to the state shown in **Table 21 - 1**.

Note In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

Caution 1. For an external reset, input a low level for at least 10 μs to the $\overline{\text{RESET}}$ pin.

To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for at least 10 μs within the operating voltage range shown in 34.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During generation of a reset signal, the X1 clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input becomes invalid.

The XT1 clock and external subsystem clock only stop oscillating or their inputs become invalid in the POR state.

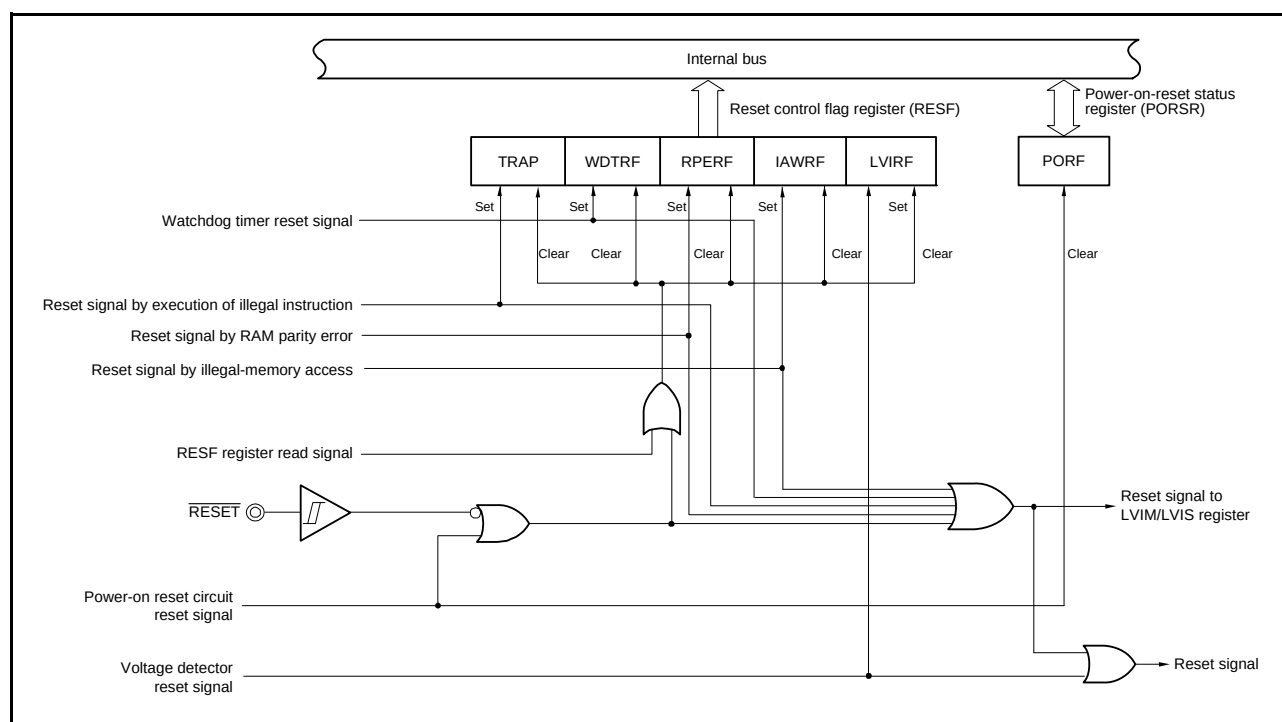
Caution 3. The port pins become the following state because SFRs and 2nd SFRs are initialized after a reset.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).
- P130: Low-level output during a reset and after the reset is released.
- Ports other than P40 and P130: High-impedance during a reset and after the reset is released.

The following registers are only initialized by a POR reset.

- RTC-related registers
- EXCLKS, OSCSELS, XTSEL, AMPHS1, and AMPHS0 bits of the CMC register

Figure 21 - 1 Block Diagram of Reset Function

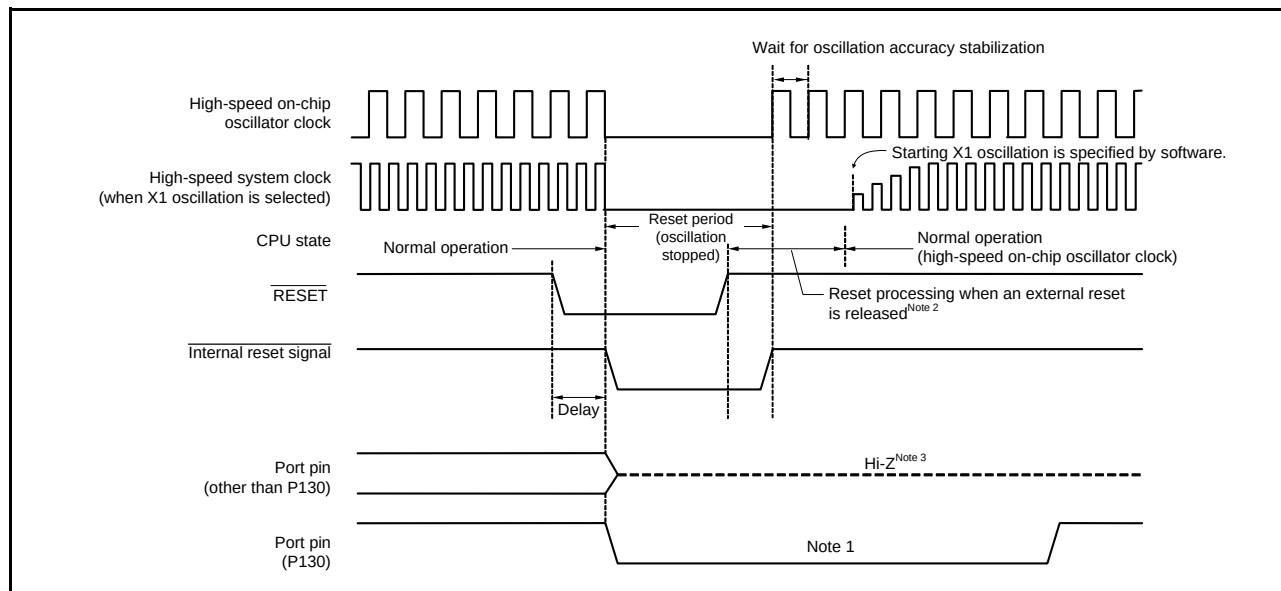


Caution An LVD0 circuit internal reset does not reset the LVD0 circuit.

21.1 Timing of Reset Operation

This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

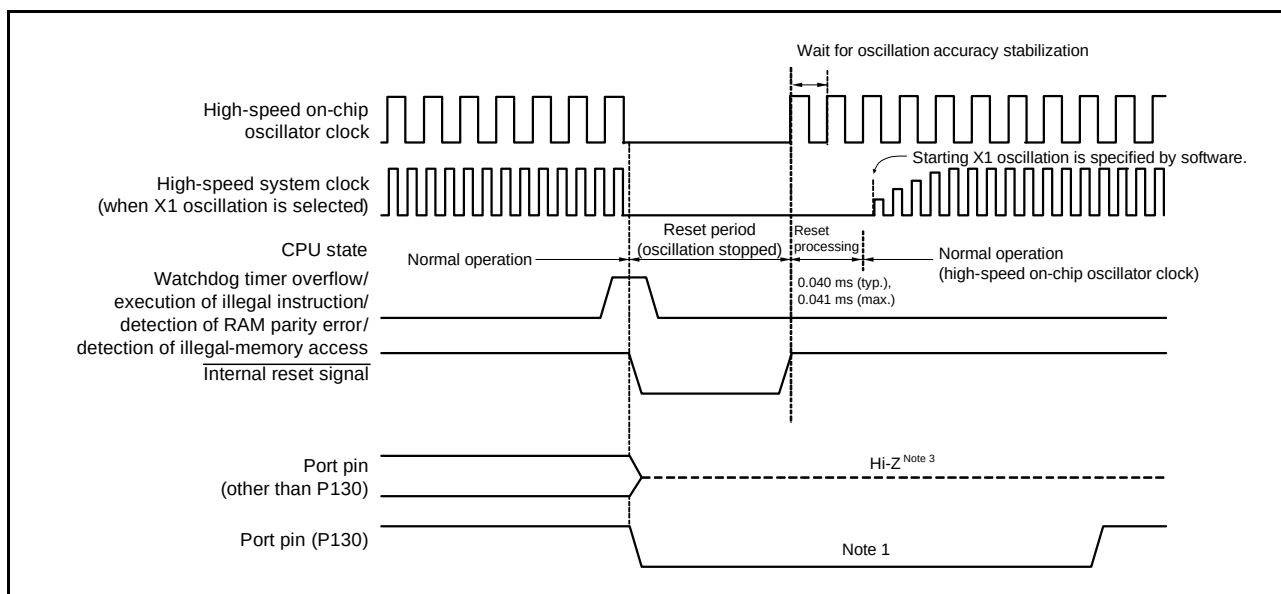
Figure 21 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



(Notes are listed on the next page.)

Release from the reset state proceeds automatically in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal-memory access. After reset processing, program execution starts with the high-speed on-chip oscillator clock as the operating clock.

Figure 21 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal-Memory Access



(Notes are listed on the next page.)

Note 1. The P130 pin outputs a low-level signal following the application of a reset. If this pin was in use as a high-level output before the application of a reset, the output signal from the P130 pin can effectively be used as the reset signal for an external device that has an active-low reset signal. To de-assert the reset signal to the external device, set the P130 pin for high-level output by software.

Note 2. Reset times (times for release from the external reset state)

The first external reset following release from the POR state:

- When the LVD is in use: 0.506 ms (typ.), 0.694 ms (max.)
- When the LVD is not in use: 0.201 ms (typ.), 0.335 ms (max.)

The second or subsequent external reset following release from the POR state:

- When the LVD is in use: 0.476 ms (typ.), 0.616 ms (max.)
- When the LVD is not in use: 0.170 ms (typ.), 0.257 ms (max.)

After power is supplied, the following voltage stabilization waiting time is required before reset processing starts after release from the external reset state.

- 4.0 ms (typ.), 9.9 ms (max.)

Note 3. The state of the P40 pin is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).

A reset from the POR circuit or by LVD0 voltage detection is released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD0}$ after the reset. After reset processing, execution of the program starts with the high-speed on-chip oscillator clock as the operating clock. For details, see **Section 22 Power-on-Reset Circuit (POR)** or **Section 23 Voltage Detector (LVD)**.

Remark V_{POR} : POR power supply rise detection voltage

V_{LVD0} : LVD0 detection voltage

Table 21 - 1 Operating States during a Reset (1/2)

Item			Operating State during a Reset
System clock			Clock supply to the CPU is stopped.
	Main system clock	f _{IH}	Operation stopped
		f _{IM}	
		f _X	Operation stopped (the X1 and X2 pins are input port mode)
		f _{EX}	Clock input invalid (the pin is input port mode)
	Subsystem clock	f _{XT}	Operable (operation stops in the POR reset state, the XT1 and XT2 pins are input port mode)
		f _{EXS}	Operable (operation stops in the POR reset state, the EXCLKS pin is input port mode)
		f _{IL}	Operation stopped
CPU			Operation stopped
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High-impedance Note
Timer array unit			Operation stopped
Realtime clock			Resets other than the POR reset: Operable POR reset: Only the values of calendar-related registers are retained.
32-bit interval timer			Operation stopped
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit			
Serial interface IICA			
Serial interface UARTA			
Data transfer controller (DTC)			
SNOOZE mode sequencer			
Event link controller (ELC)			
Power-on-reset function			
Voltage detection function			LVD0: LVD0 operation is possible following an LVD0 reset but is stopped following other types of reset. LVD1: Operation stopped

Table 21 - 1 Operating States during a Reset (2/2)

Item		Operating State during a Reset	
External interrupt		Operation stopped	
Key interrupt function			
Capacitive sensing unit (CTSU)			
CRC operation function	High-speed CRC		
	General-purpose CRC		
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Note P40 and P130 become the following states.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the on-chip pull-up resistor).
- P130: Low-level output during the reset period

Remark f_{ih}: High-speed on-chip oscillator clock f_x: X1 oscillation clock
f_{im}: Middle-speed on-chip oscillator clock f_{ex}: External main system clock
f_{xt}: XT1 oscillation clock f_{exs}: External subsystem clock
f_{il}: Low-speed on-chip oscillator clock

Table 21 - 2 States of the Hardware Blocks after a Reset is Released

Hardware		State after a Reset is Released
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set. Note
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note The contents of the PC are undefined during a reset and until the clock oscillation becomes stable after the reset is released.

Remark For the states of the special function registers (SFRs) after a reset is released, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd special function register) area**.

21.2 Registers for Controlling the Reset Function

The following registers are used to control the reset function.

- Reset control flag register (RESF)
- Power-on-reset status register (PORSR)
- Peripheral reset control register 0 (PRR0)
- Peripheral reset control register 1 (PRR1)

21.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The RESF register indicates which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

The flags TRAP, WDTRF, RPERF, IAWRF, and LVIRF are automatically cleared by any of the following event.

- Reset input via the RESET pin
- Reset by the power-on-reset (POR) circuit
- The RESF register is accessed.

Figure 21 - 4 Format of Reset Control Flag Register (RESF)

Address: FFFA8H
After reset: Undefined^{Note 1}
R/W: R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF
TRAP Internal reset request by execution of illegal instruction ^{Note 2}								
0 Internal reset request is not generated, or the RESF register is cleared.								
1 Internal reset request is generated.								
WDTRF Internal reset request by watchdog timer (WDT)								
0 Internal reset request is not generated, or the RESF register is cleared.								
1 Internal reset request is generated.								
RPERF Internal reset request by RAM parity error								
0 Internal reset request is not generated, or the RESF register is cleared.								
1 Internal reset request is generated.								
IAWRF Internal reset request by illegal-memory access								
0 Internal reset request is not generated, or the RESF register is cleared.								
1 Internal reset request is generated.								
LVIRF Internal reset request by voltage detector (LVD0 or LVD1)								
0 Internal reset request is not generated, or the RESF register is cleared.								
1 Internal reset request is generated.								

Note 1. The value after reset varies depending on the reset source. See **Table 21 - 3**.

Note 2. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

(Cautions are listed on the next page.)

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.

Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 24.3.4 RAM parity error detection.

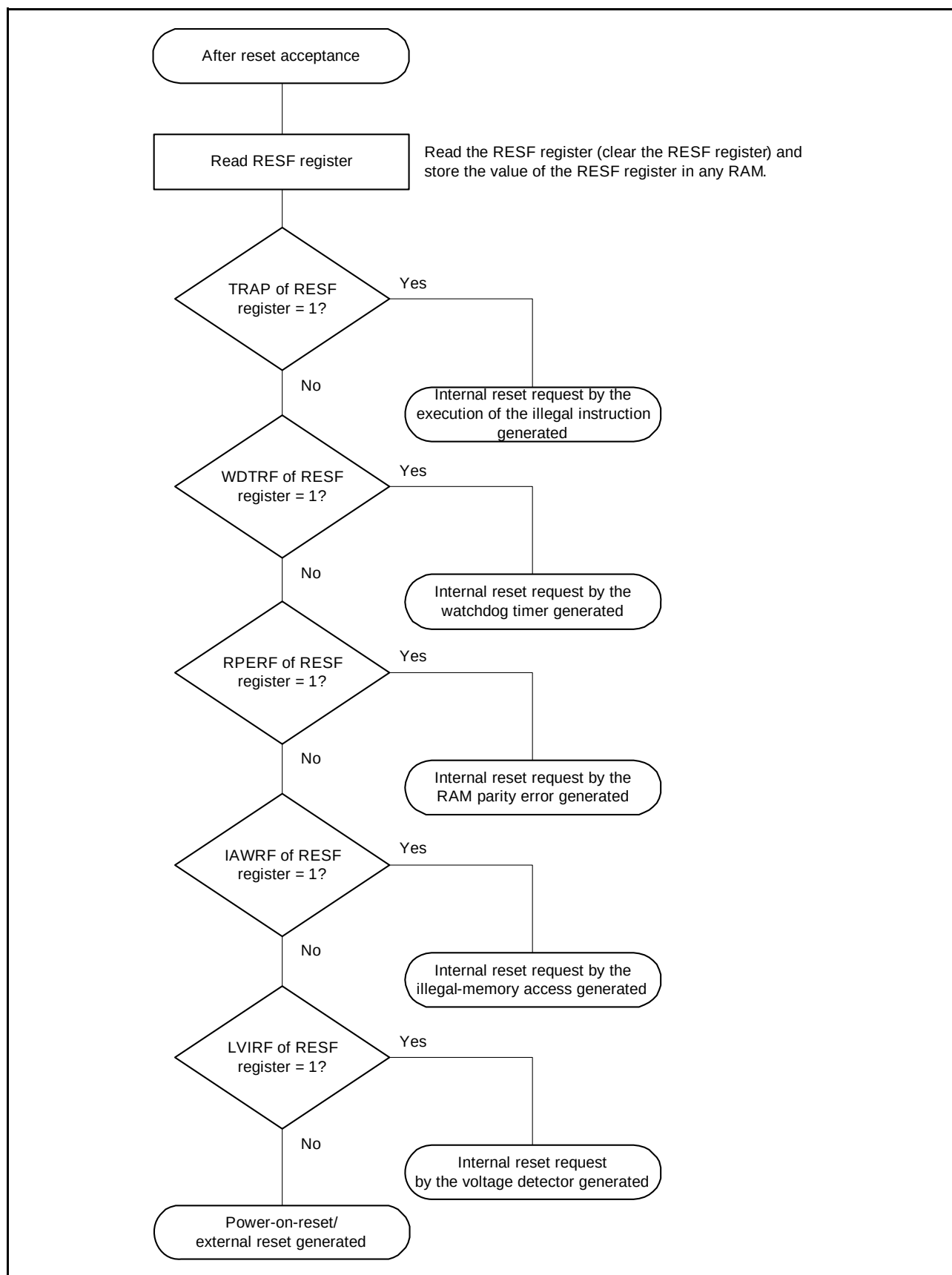
The state of the RESF register when a reset request is generated is shown in **Table 21 - 3**.

Table 21 - 3 State of the RESF Register when Reset Request is Generated

Reset Source Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal- Memory Access	Reset by LVD0 or LVD1
TRAP	Cleared (0)	Cleared (0)	Set (1)	Retained	Retained	Retained	Retained
WDTRF			Retained	Set (1)			
RPERF				Retained	Set (1)		
IAWRF					Retained	Set (1)	
LVIRF						Retained	Set (1)

The RESF register is automatically cleared after it is read by using an 8-bit memory manipulation instruction. **Figure 21 - 5** shows the procedure for checking a reset source.

Figure 21 - 5 Example of Procedure for Checking Reset Source



The above flow is an example of the procedure for checking a reset source.

21.2.2 Power-on-reset status register (PORSR)

The PORSR register is used to check the occurrence of a power-on reset.
Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 to the bit has no effect.
Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset.
The PORSR register can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of this register following a power-on reset is 00H.

Caution 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another source occurs.

Caution 2. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.

Figure 21 - 6 Format of Power-on-Reset Status Register (PORSR)

Address: F00F9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

21.2.3 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 21 - 7 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	1	<0>
PRR0	0	0	ADCRES	IICA0RES	SAU1RES	SAU0RES	0	TAU0RES
Bit n	Control resetting of the on-chip peripheral modules							
0	The corresponding on-chip peripheral module is released from the reset state.							
1	The corresponding on-chip peripheral module is in the reset state. • The SFRs for use with the corresponding on-chip peripheral module are initialized.							

Remark n = 0, 2 to 5

The on-chip peripheral modules controlled by individual bits are as follows.

Table 21 - 4 On-chip Peripheral Modules Controlled by Individual Bits in PRR0

Bit	Bit Name	Controlled On-chip Peripheral Modules
5	ADCRES	A/D converter
4	IICA0RES	Serial interface IICA0
3	SAU1RES	Serial array unit 1
2	SAU0RES	Serial array unit 0
0	TAU0RES	Timer array unit 0

Caution Be sure to set the following bits to 0.

Bits 7, 6, 4, 3, and 1 in the 16- and 20-pin products

Bits 7, 6, 3, and 1 in the 24- and 25-pin products

Bits 7, 6, and 1 in the 30-, 32-, 36-, 40-, and 48-pin products

21.2.4 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 21 - 8 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	<0>
PRR1	0	SMSRES	0	TML32RES	0	0	0	CTSURES

Bit n	Control resetting of the on-chip peripheral modules
0	The corresponding on-chip peripheral module is released from the reset state.
1	The corresponding on-chip peripheral module is in the reset state. • The SFRs for use with the corresponding on-chip peripheral module are initialized.

Remark n = 0, 4, 6

The on-chip peripheral modules controlled by individual bits are as follows.

Table 21 - 5 On-chip Peripheral Modules Controlled by Individual Bits in PRR1

Bit	Bit Name	Controlled On-chip Peripheral Modules
6	SMSRES	SNOOZE mode sequencer
4	TML32RES	32-bit interval timer
0	CTSURES	Capacitive sensing unit

Caution The UARTEAn bit in the ASIMA00 register is used to control resetting of the UARTEA. For details, see 15.2.4 Operation mode setting register 0 (ASIMAn0).

Section 22 Power-on-Reset Circuit (POR)

22.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.

The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **34.4 AC Characteristics**.

This is done by utilizing LVD0 or controlling the externally input reset signal.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing LVD0 or externally input reset signal, before the operating voltage falls below the range defined in **34.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Caution If an internal reset signal is generated by the power-on-reset circuit, the reset control flag register (RESF) and power-on-reset status register (PORSR) are cleared to 00H.

Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. For details of the RESF register, see **Section 21 Reset Function**.

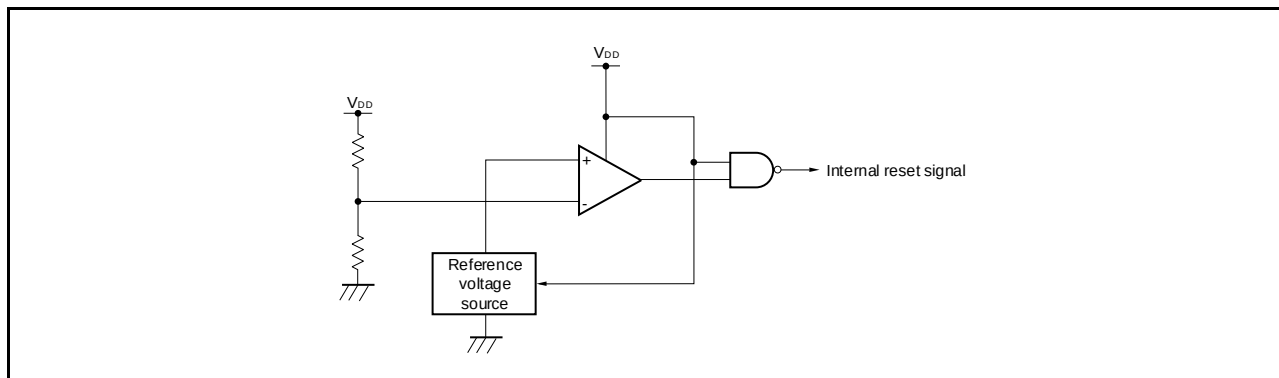
Remark 2. The power-on-reset status register (PORSR) is used to check the occurrence of an internal reset from the power-on-reset circuit. For details of the PORSR register, see **Section 21 Reset Function**.

Remark 3. V_{POR} : POR power supply rise detection voltage
 V_{PDR} : POR power supply fall detection voltage
For details, see **34.6.3 Characteristics of the POR circuit**.

22.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in **Figure 22 - 1**.

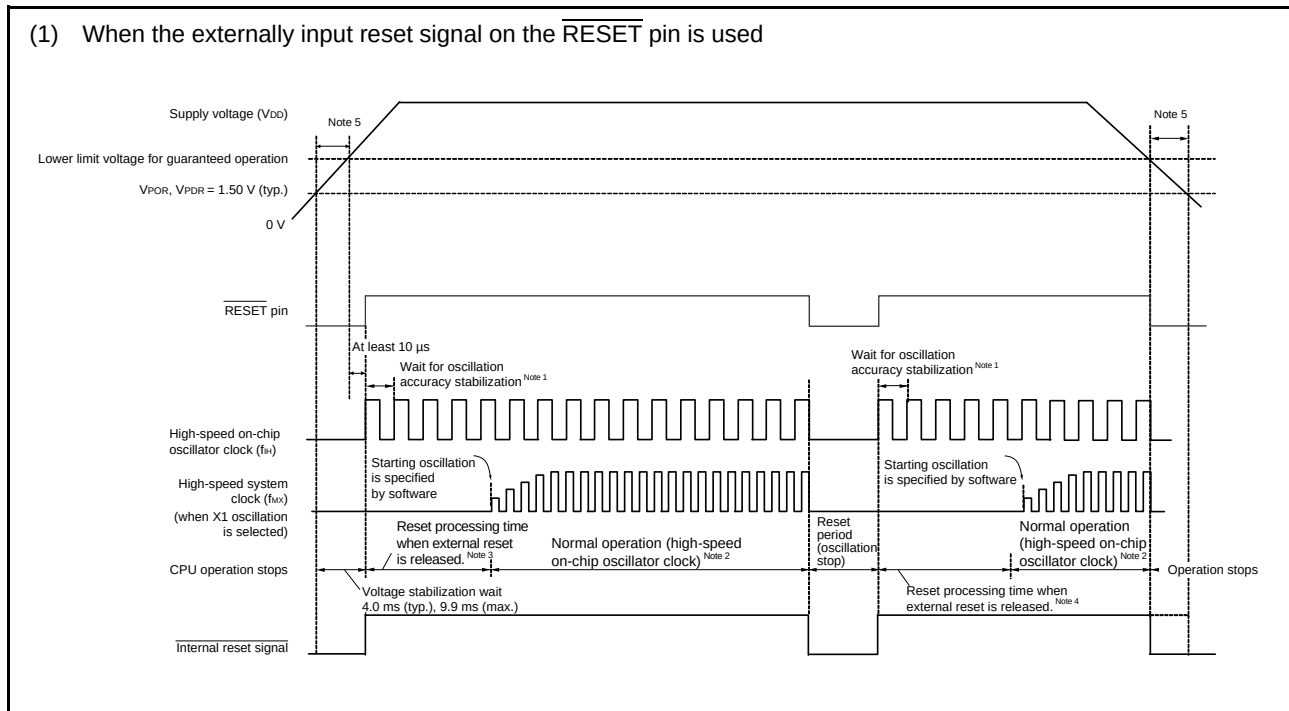
Figure 22 - 1 Block Diagram of Power-on-reset Circuit



22.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown on the following pages.

Figure 22 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)

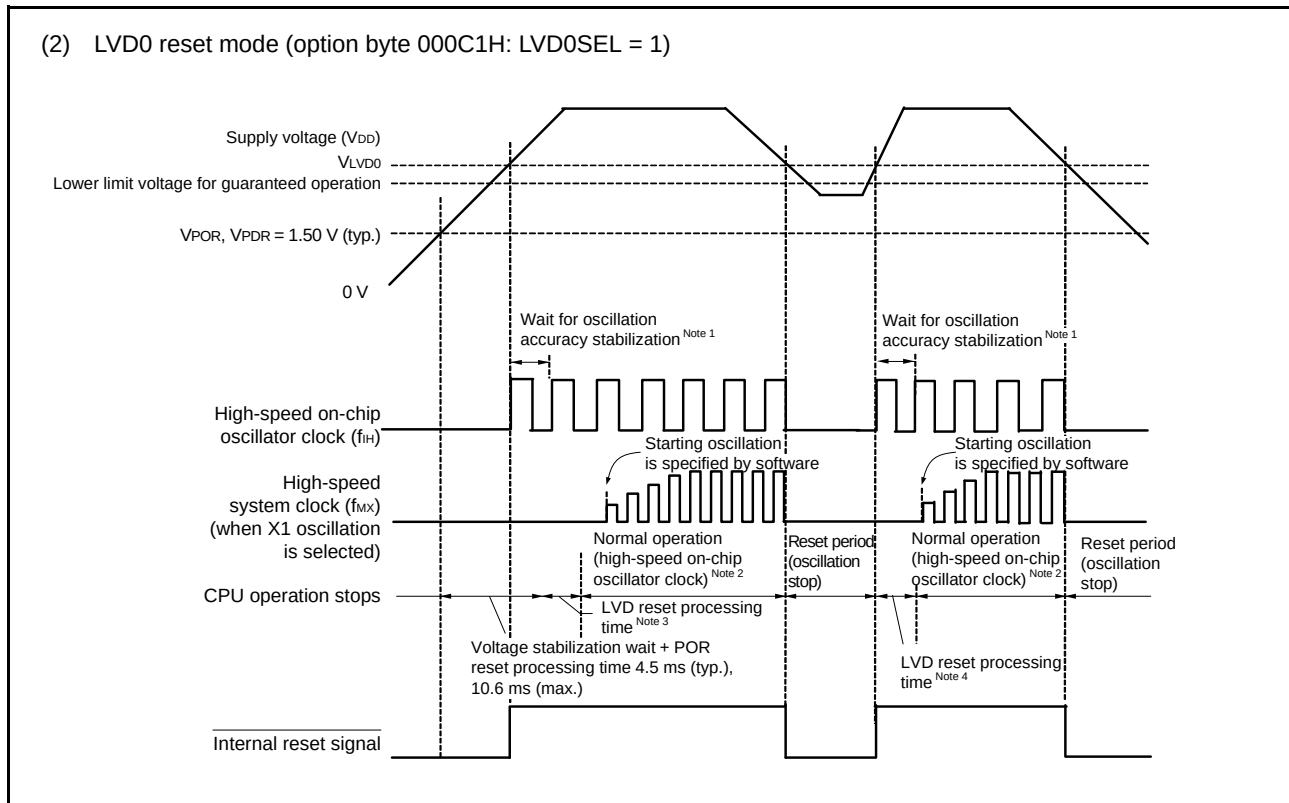


- Note 1.** The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.
- Note 2.** The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.
- Note 3.** The time until normal operation starts includes the following reset processing time when the external reset is released (release from the first external reset following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after V_{POR} (1.50 V, typ.) is reached.
With the LVD circuit in use: 0.506 ms (typ.), 0.694 ms (max.)
With the LVD circuit not in use: 0.201 ms (typ.), 0.335 ms (max.)
- Note 4.** The reset processing times in the case of the second or subsequent external reset following release from the POR state are listed below.
With the LVD circuit in use: 0.476 ms (typ.), 0.616 ms (max.)
With the LVD circuit not in use: 0.170 ms (typ.), 0.257 ms (max.)
- Note 5.** After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **34.4 AC Characteristics**. This is done by controlling the externally input reset signal.
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the range of operation. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when LVD0 is off. For details, see Section 23 Voltage Detector (LVD).

Remark V_{POR} : POR power supply rise detection voltage
 V_{PDR} : POR power supply fall detection voltage

Figure 22 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD0 detection level (V_{LVD0}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.50 V, typ.) is reached.

LVD reset processing time: 0 to 0.041 ms (max.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD0), the following LVD reset processing time is required after the LVD0 detection level (V_{LVD0}) is reached.

LVD reset processing time: 0.040 ms (typ.), 0.041 ms (max.)

Remark 1. V_{LVDH} , V_{LVDL} : LVD detection voltage

V_{POR} : POR power supply rise detection voltage

V_{PDR} : POR power supply fall detection voltage

Remark 2. When the LVD0 interrupt mode is selected (option byte 000C1H: LVD0SEL = 0), the time until normal operation starts after power is turned on is the same as the time specified in **Note 3 of Figure 22 - 2 (2)**.

Remark 3. Operation of LVD1 is stopped when power is initially supplied. LVD1 is also stopped by an internal reset.

Section 23 Voltage Detector (LVD)

23.1 Functions of Voltage Detector

Enabling, selecting the operation mode, and setting the detection voltage (VLVD0) for voltage detector 0 (LVD0) are done by using an option byte (000C1H). On the other hand, enabling, selecting the operation mode, and setting the detection voltage (VLVD1) for voltage detector 1 (LVD1) are done by using the voltage detection level register.

The voltage detectors have the following functions.

- LVD0 and LVD1 compare the supply voltage (V_{DD}) with the detection voltage (VLVD0, VLVD1), and generate an internal reset or internal interrupt signal.
- The option byte is used to select the detection voltage (VLVD0) for LVD0 from among 6 voltages. For details, see **Section 29 Option Bytes**.
- The voltage detection level register is used to select the detection voltage (VLVD1) for LVD1 from among 18 voltages.
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **34.4 AC Characteristics**. This is done by utilizing LVD0 or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing LVD0 or controlling the externally input reset signal before the voltage falls below the operating range.

The internal reset and internal interrupt signals are generated in each mode as follows.

Reset mode LVD0	Reset mode LVD1	Interrupt mode LVD0	Interrupt mode LVD1
Deasserts an internal reset signal on detecting $V_{DD} \geq VLVD0$. Generates an internal reset on detecting $V_{DD} < VLVD0$ and retains the reset state until $V_{DD} \geq VLVD0$ is detected.	Generates an internal reset on detecting $V_{DD} < VLVD1$ after LVD1 operation has been enabled.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq VLVD0$. Releases the LVD internal reset by detecting $V_{DD} \geq VLVD0$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < VLVD0$ or $V_{DD} \geq VLVD0$ after the LVD internal reset is released.	Generates an interrupt request signal (INTLVI) on detecting $V_{DD} < VLVD1$ after LVD1 operation has been enabled. After the first detection, generates an interrupt request signal (INTLVI) on detecting $V_{DD} < VLVD1$ or $V_{DD} \geq VLVD1$.

While LVD0 or LVD1 is operating, whether the supply voltage is no less than or less than the detection level can be checked by reading the voltage detection flag (LVDnF: bits 0 and 1 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **Section 21 Reset Function**.

23.2 Configuration of Voltage Detector

Figures 23 - 1 and 23 - 2 show the block diagrams of the voltage detectors.

Figure 23 - 1 Block Diagram of LVD0

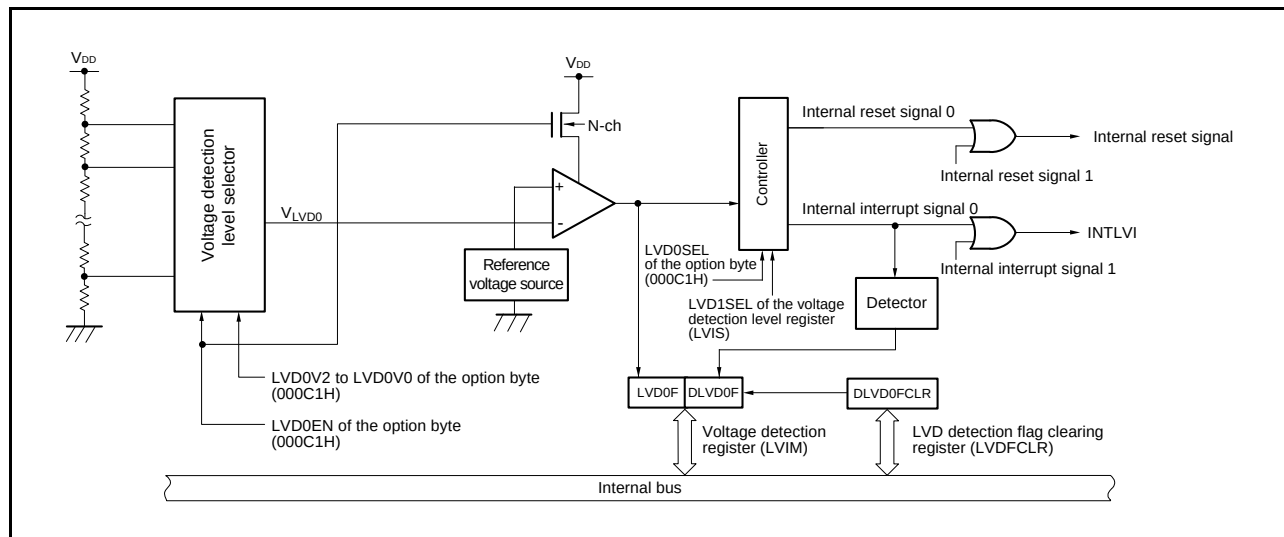
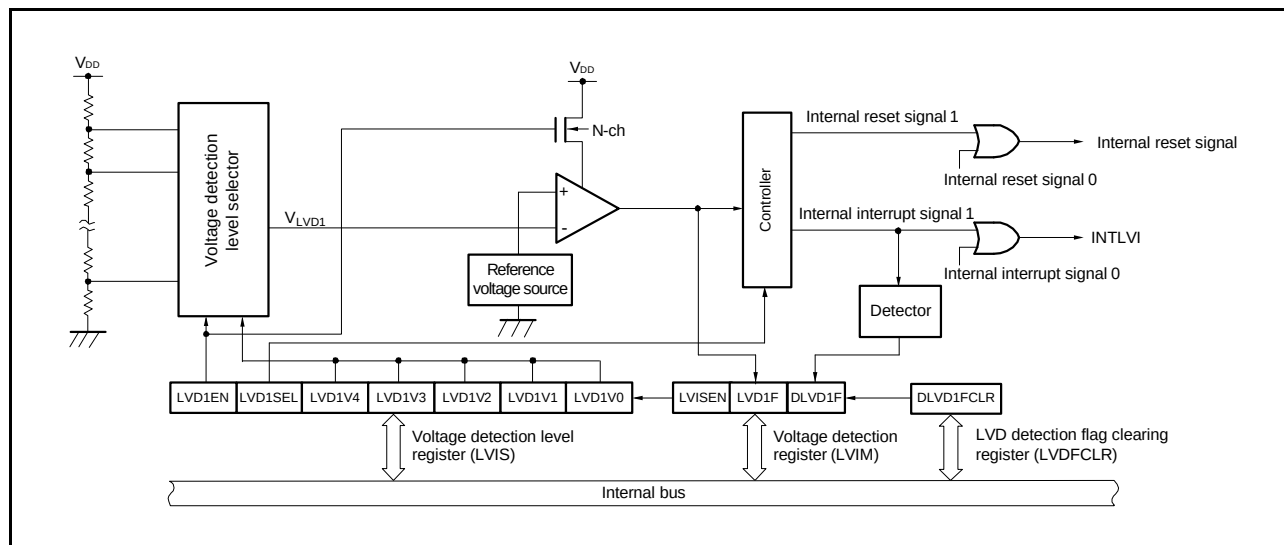


Figure 23 - 2 Block Diagram of LVD1



23.3 Registers for Controlling the Voltage Detector

The following registers are used to control the voltage detector.

- User option byte (000C1H/040C1H): See **Section 29 Option Bytes**.
- Voltage detection register (LVIM)
- LVD detection flag clearing register (LVDFCLR)
- Voltage detection level register (LVIS)

23.3.1 Voltage detection register (LVIM)

The LVIM register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the states of LVD0 and LVD1.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 23 - 3 Format of Voltage Detection Register (LVIM)

Address: FFFA9H
After reset: 00H^{Note 1}
R/W: R/W^{Notes 2, 3}

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
LVIM	LVISEN	0	0	0	DLVD1F	DLVD0F	LVD1F	LVD0F
LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)							
0	Rewriting of the LVIS register is disabled.							
1 ^{Note 4}	Rewriting of the LVIS register is enabled (reset and interrupt generation by LVD1 are masked).							
DLVDnF	LVDn detection interrupt flag (n = 0, 1)							
0	The given LVDn interrupt has not been detected.							
1	The given LVDn interrupt has been detected.							
LVDnF	Voltage detection flag (n = 0, 1)							
0	Supply voltage (V _{DD}) ≥ detection voltage (V _{LVDn}), or when LVD is off							
1	Supply voltage (V _{DD}) < detection voltage (V _{LVDn})							

Note 1. The value after a reset is 01H when LVD0 operation is enabled and the power supply voltage (V_{DD}) is less than the detection voltage (V_{LVD0}).

Note 2. Bits 0 and 1 are read-only.

Note 3. Bits 2 and 3 are read-only. These bits are cleared by using the LVD detection flag clearing register (LVDFCLR).

Note 4. While the LVISEN bit is 1, the reset and interrupt generation by LVD1 are masked. Therefore, clear the LVISEN bit to 0 after having written a new value to the LVIS register.

23.3.2 LVD detection flag clearing register (LVDFCLR)

The LVDFCLR register is used to clear the interrupt detection flags (DLVD0F and DLVD1F) of the voltage detection register (LVIM).

The LVDFCLR register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 23 - 4 Format of LVD Detection Flag Clearing Register (LVDFCLR)

Address: F0218H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	1	0
LVDFCLR	0	0	0	0	DLVD1FCLR	DLVD0FCLR	0	0

DLVD1FCLR ^{Note}	Clearing DLVD1F
0	No effect
1	Writing 1 to this bit clears the DLVD1F flag.

DLVD0FCLR ^{Note}	Clearing DLVD0F
0	No effect
1	Writing 1 to this bit clears the DLVD0F flag.

Note Only 1 can be written to this bit. Writing 0 has no effect.
The bit is read as 0 even after 1 has been written to it.

23.3.3 Voltage detection level register (LVIS)

The LVIS register is used to select the voltage detection level for LVD1.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 19H.

Figure 23 - 5 Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH
After reset: 19H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
LVIS	LVD1EN	LVD1SEL	0	LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0

LVD1EN	Enabling operation of LVD1
0	Operation stopped
1	Operation enabled

LVD1SEL	Operation mode of LVD1
0	Interrupt mode
1	Reset mode

LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0	Detection voltages for LVD1 ^{Notes 1, 3, 4}		
						Rising edge	Falling edge
1	1	1	1	1	VLVD117	1.67 V ^{Note 2}	1.63 V ^{Note 2}
1	1	1	1	0	VLVD116	1.78 V ^{Note 2}	1.74 V ^{Note 2}
1	1	1	0	1	VLVD115	1.88 V ^{Note 2}	1.84 V ^{Note 2}
1	1	1	0	0	VLVD114	1.98 V	1.94 V
1	1	0	1	1	VLVD113	2.09 V	2.04 V
1	1	0	1	0	VLVD112	2.20 V	2.15 V
1	1	0	0	1	VLVD111	2.30 V	2.25 V
1	1	0	0	0	VLVD110	2.40 V	2.35 V
1	0	1	1	1	VLVD19	2.50 V	2.45 V
1	0	1	1	0	VLVD18	2.66 V	2.60 V
1	0	1	0	1	VLVD17	2.82 V	2.76 V
1	0	1	0	0	VLVD16	2.97 V	2.91 V
1	0	0	1	1	VLVD15	3.13 V	3.06 V
1	0	0	1	0	VLVD14	3.35 V	3.27 V
1	0	0	0	1	VLVD13	3.55 V	3.47 V
1	0	0	0	0	VLVD12	3.75 V	3.67 V
0	1	1	1	1	VLVD11	3.96 V	3.88 V
0	1	1	1	0	VLVD10	4.16 V	4.08 V

Note 1. The LVD1V4 to LVD1V0 bits can only be rewritten once after release from the reset state.

Note 2. This setting can only be used when LVD0 is off.

Note 3. When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of LVD0.
(**Note 4** and **Caution** are listed on the next page.)

Note 4. If LVD0 is set to interrupt mode and the LVD0 detection voltage is greater than the LVD1 detection voltage, LVD0 becomes undefined after the LVD1 setting following release from the reset state.

Caution When the values in the LVIS register are to be changed, do so according to the procedure described in Figures 23 - 10 and 23 - 11.

23.4 Operation of Voltage Detector

23.4.1 When used as reset mode

Enabling, selecting the operation mode (reset mode: LVD0SEL = 1), and setting the detection voltage (VLVD0) for LVD0 are done by using an option byte (000C1H).

On the other hand, enabling, selecting the operation mode (reset mode: LVD1SEL = 1), and setting the detection voltage (VLVD1) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD reset mode

When LVD0 is set for the reset mode (the value of the LVD0SEL bit in the option byte is 1), the state of the internal reset being applied by LVD0 is retained until the power supply voltage (VDD) exceeds the rising voltage detection level (VLVD0) after power has been supplied. The internal reset is released when the supply voltage (VDD) exceeds the rising voltage detection level (VLVD0).

At the fall of the operating voltage, an internal reset by LVD0 is generated when the power supply voltage (VDD) falls below the falling voltage detection level (VLVD0).

Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an internal reset when the power supply voltage (VDD) falls below the voltage detection level (VLVD1). If operation of LVD1 is enabled while the power supply voltage (VDD) is lower than the voltage detection level (VLVD1), it generates an internal reset at the time its operation is enabled. If LVD1 is set for reset mode, LVD0 is placed in interrupt mode. In addition, the generation of an internal reset by LVD1 places LVD0 in reset mode.

LVD1 detection voltage can only be set once after release from the reset state.

Figures 23 - 6 and 23 - 7 show the timing of the internal reset signal generated in the LVD reset mode.

Figure 23 - 6 Timing of LVD0 Internal Reset Signal Generation

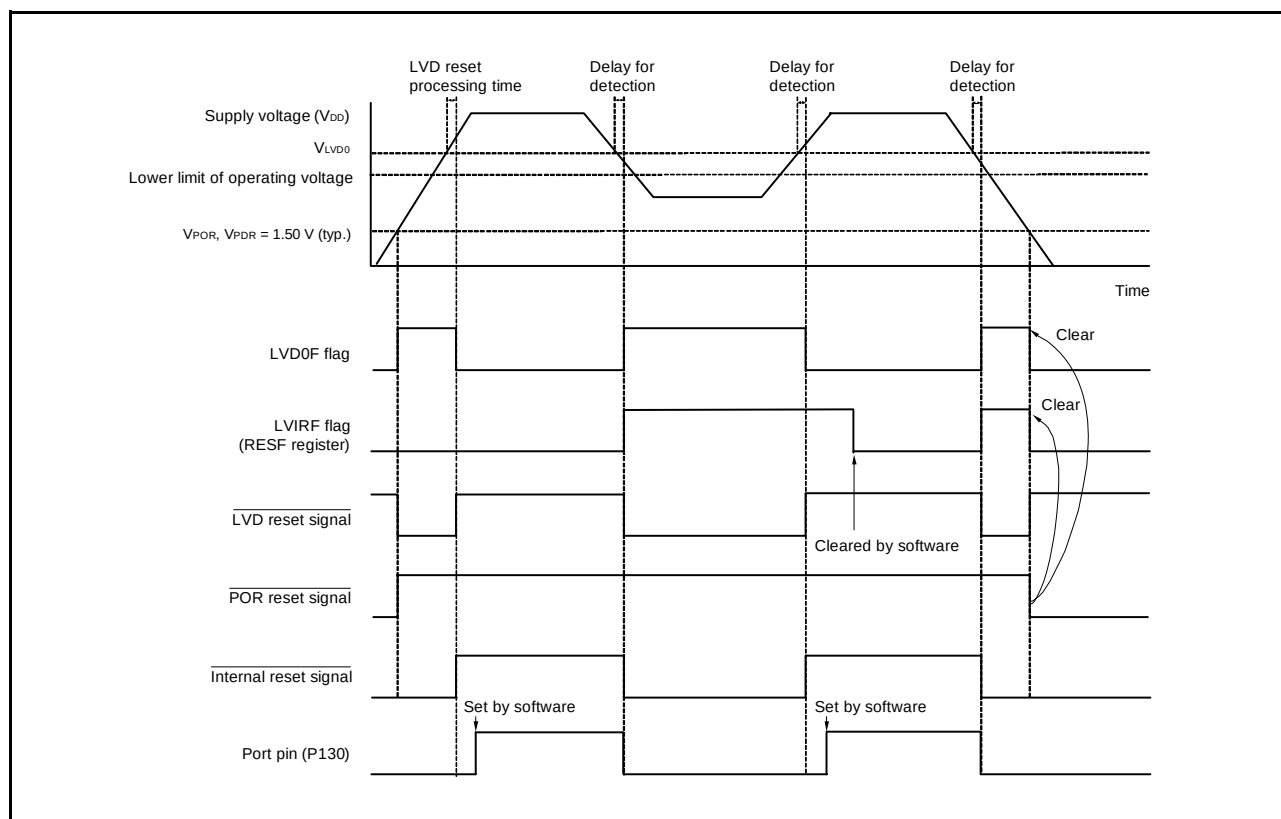
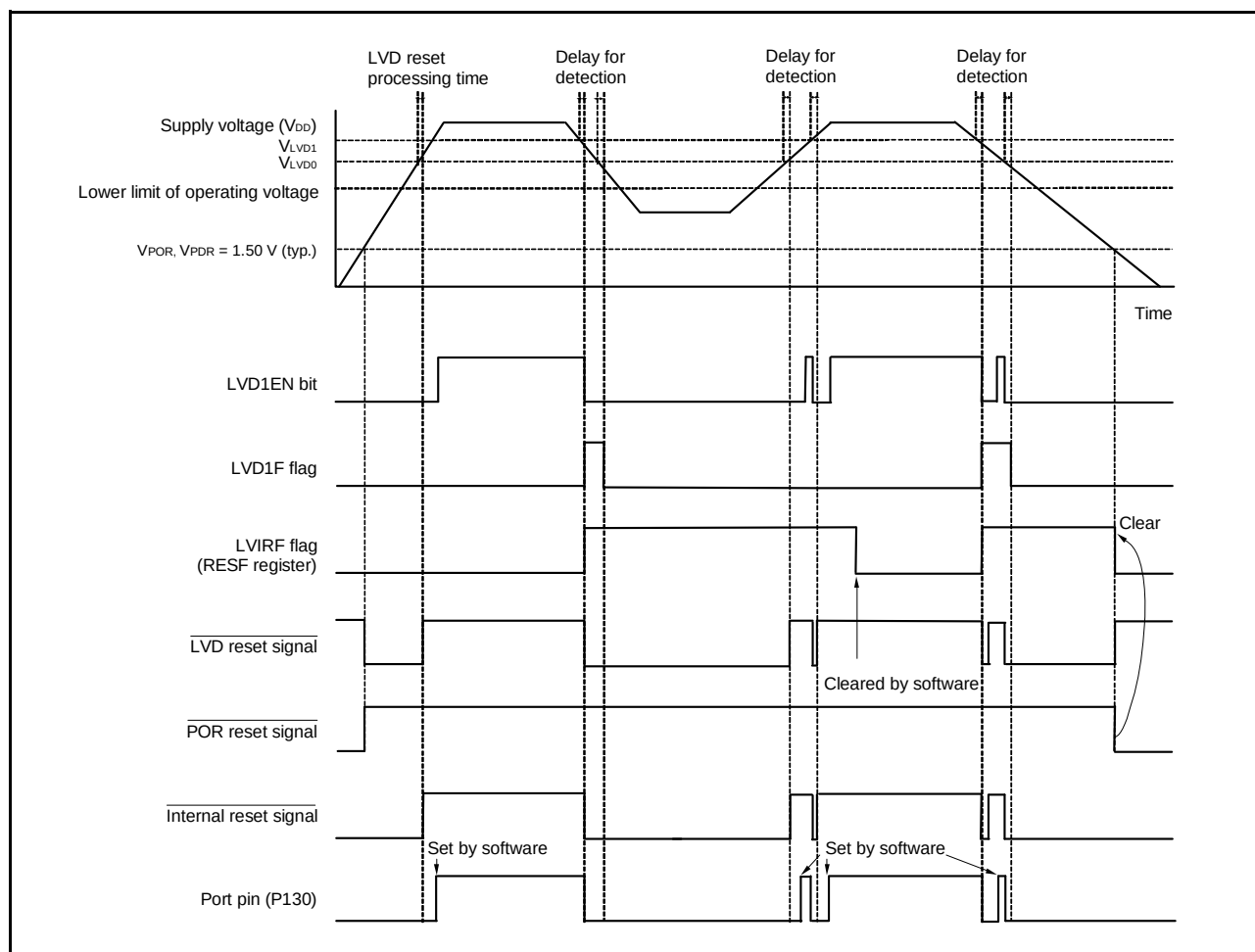


Figure 23 - 7 Timing of LVD1 Internal Reset Signal Generation



Remark LVD0: Reset mode

23.4.2 When used as interrupt mode

Enabling, selecting the operation mode (interrupt mode: LVD0SEL = 0), and setting the detection voltage (VLVD0) for LVD0 are done by using an option byte (000C1H).

On the other hand, enabling, selecting the operation mode (interrupt mode: LVD1SEL = 0), and setting the detection voltage (VLVD1) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD interrupt mode

When LVD0 is set for the interrupt mode (the value of the LVD0SEL bit in the option byte is 0), the state of the internal reset being applied by LVD0 is retained until the power supply voltage (VDD) exceeds the rising voltage detection level (VLVD0) immediately after a reset has been generated. The internal reset is released when the power supply voltage (VDD) exceeds the rising voltage detection level (VLVD0).

After the internal reset signal has been deasserted, LVD0 generates an interrupt request signal (INTLVI) if the power supply voltage (VDD) falls below the voltage detection level (VLVD0). Similarly, when the power supply voltage (VDD) rises above the voltage detection level (VLVD0), LVD0 also generates an interrupt request signal (INTLVI). When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **34.4 AC Characteristics**.

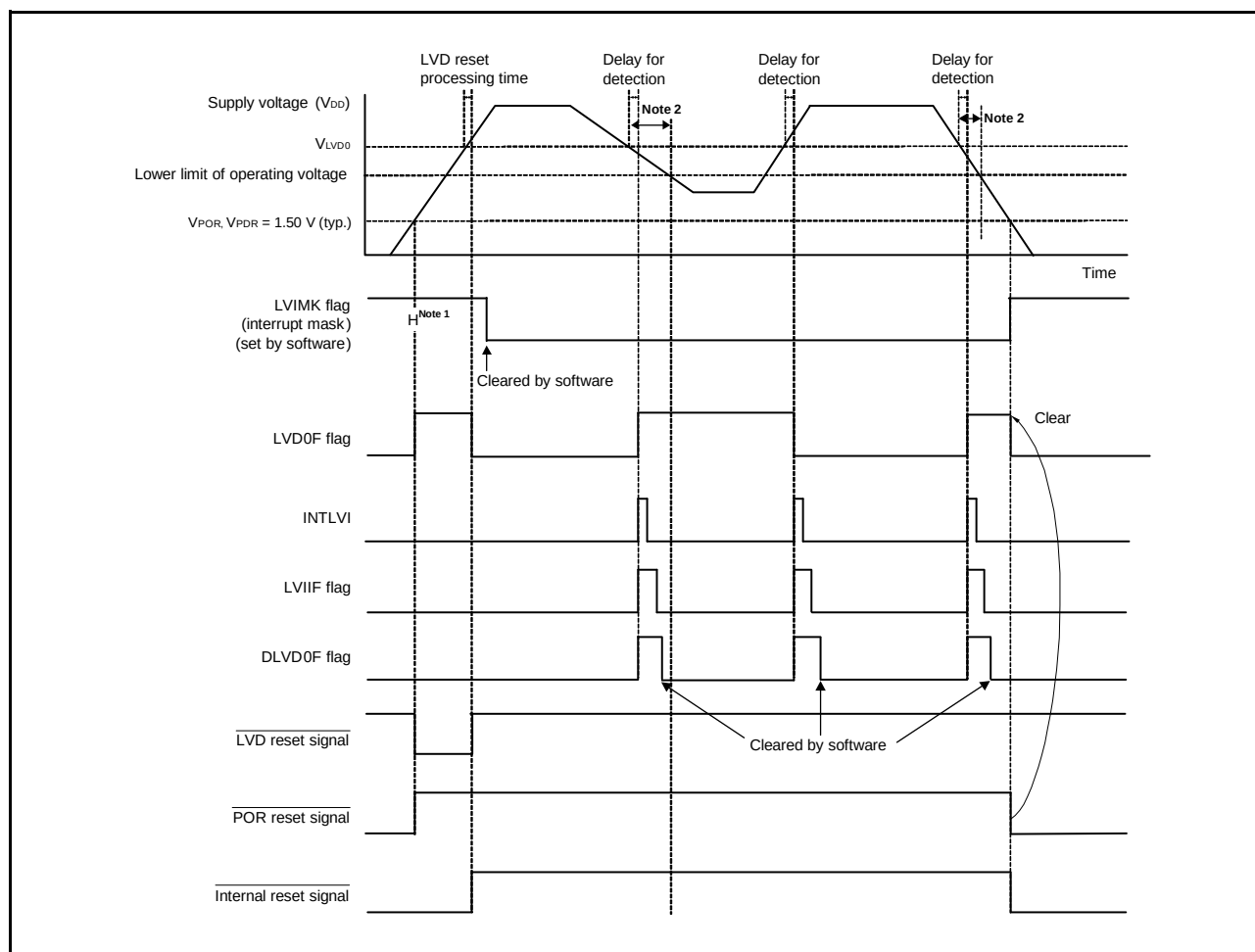
When restarting the operation, make sure that the power supply voltage has returned within the operating voltage range.

Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an interrupt request signal (INTLVI) when the power supply voltage (VDD) falls below the voltage detection level (VLVD1). Similarly, when the power supply voltage (VDD) rises above the voltage detection level (VLVD1), LVD1 also generates an interrupt request signal (INTLVI). Note that if operation of LVD1 is enabled while the power supply voltage (VDD) is lower than the voltage detection level (VLVD1), it generates an interrupt request signal (INTLVI) at the time its operation is enabled.

LVD1 detection voltage can only be set once after release from the reset state.

Figures 23 - 8 and 23 - 9 show the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 23 - 8 Timing of LVD0 Interrupt Request Signal Generation

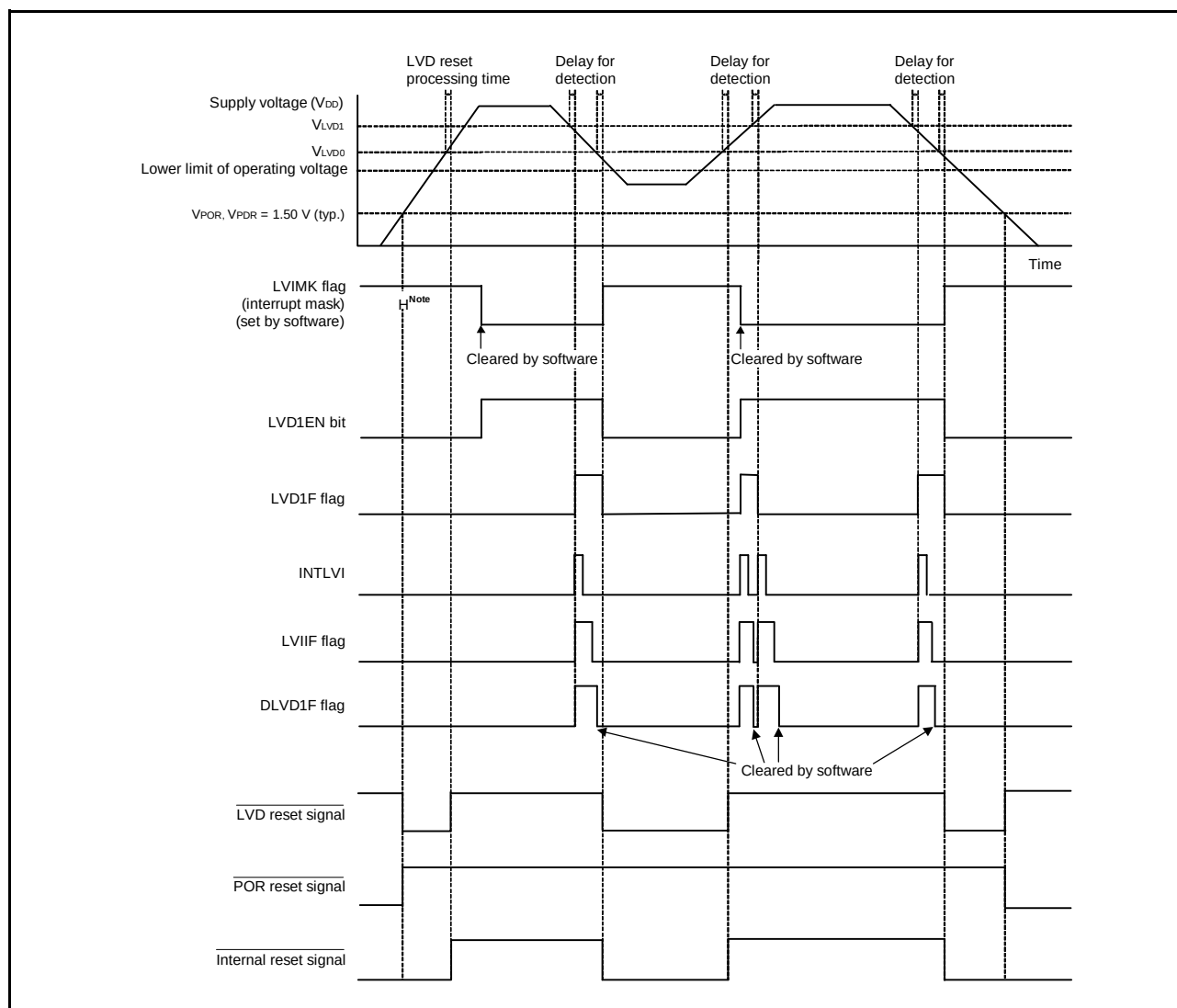


Note 1. The LVIMK flag is set to 1 by reset signal generation.

Note 2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **34.4 AC**

Characteristics. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

Figure 23 - 9 Timing of LVD1 Interrupt Request Signal Generation



Note The LVIMK flag is set to 1 by reset signal generation.

Remark LVD0: Reset mode

23.5 Points for Caution when the Voltage Detector is to be Used

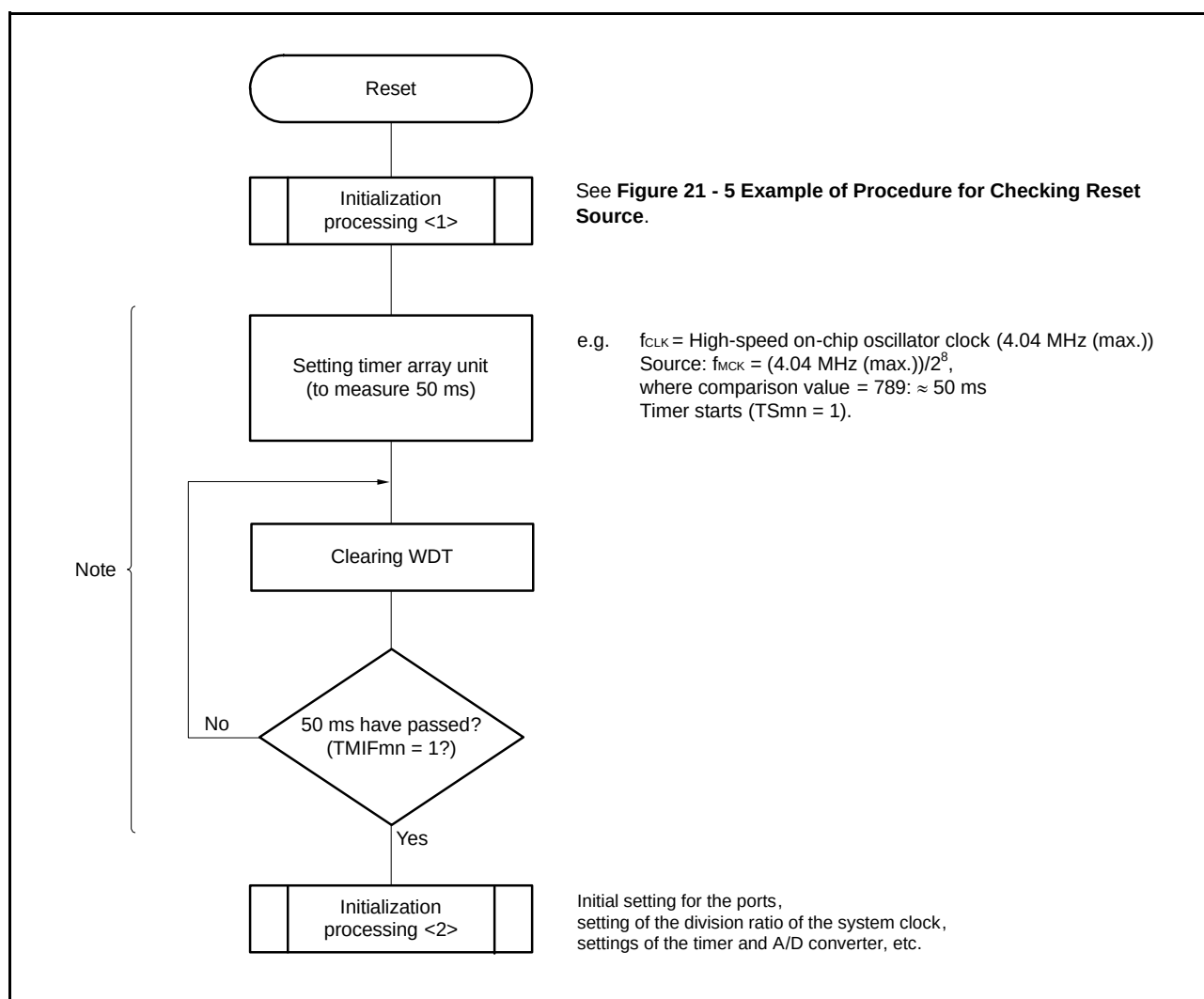
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the detection voltages for LVD0 or LVD1, the system may be repeatedly reset and released from the reset state. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 23 - 10 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of the Detection Voltages for LVD0 or LVD1



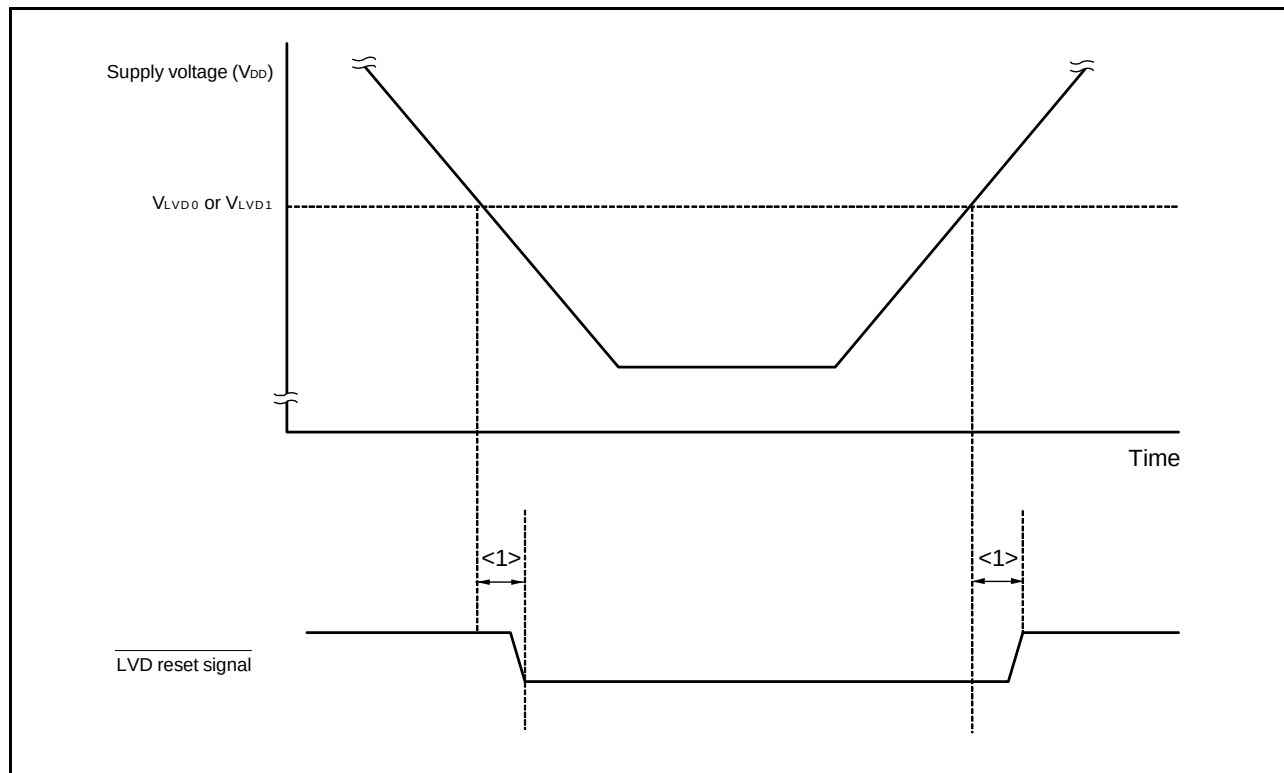
Note If reset is generated again during this period, initialization processing <2> is not started.

Remark $m = 0, 1$
 $n = 0 \text{ to } 7$

- (2) Delays from the time an LVD0 or LVD1 reset source condition is satisfied until an LVD0 or LVD1 reset has been generated and deasserted

The delay is from the time the power supply voltage (V_{DD}) becomes less than the LVD0 or LVD1 falling detection voltage (V_{LVD0} or V_{LVD1}) until the LVD0 or LVD1 reset is generated. In the same way, the delay is from the time the LVD0 or LVD1 rising detection voltage (V_{LVD0} or V_{LVD1}) becomes no greater than the power supply voltage (V_{DD}) until the LVD0 or LVD1 reset is deasserted. See **Figure 23 - 11**.

Figure 23 - 11 Delays from the Time an LVD0 or LVD1 Reset Source Condition is Satisfied until an LVD0 or LVD1 Reset has been Generated and Deasserted



$\langle 1 \rangle$: Delay for detection (500 μ s (max.))

(3) Turning power on when LVD0 is off

Use the external reset input via the $\overline{\text{RESET}}$ pin when LVD0 is off.

For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **34.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD0 is off or the interrupt mode is selected

When the operating voltage falls with LVD0 off or with the interrupt mode selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal before the voltage falls below the operating voltage range defined in **34.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

(5) Procedure for setting the LVD1 detection voltage

Follow the procedure below to set the LVD1 detection voltage. After step 3, LVD1 is enabled after the stabilization waiting time (at least 500 μs) has elapsed.

1. Set the LVIEN bit of the LVIM register to 1.
2. Set the LVD1EN bit of the LVIS register to 1 and change the setting of the LVD1V4 to LVD1V0 bits.
3. Set the LVIEN bit of the LVIM register to 0.

Section 24 Safety Functions

24.1 Overview of Safety Functions

The RL78/G22 provides the following safety functions to comply with the IEC60730 and IEC61508 safety standards. These safety functions enable the microcontroller to self-diagnose abnormalities and safely stop operating if an abnormality is detected.

(1) Flash memory CRC operation (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

The following two CRC functions are provided in the RL78/G22 and they can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General-purpose CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) Flash memory guard function

This prevents rewriting of data in the flash memory due to incorrect CPU operations.

(3) RAM parity error detection

This detects parity errors when reading RAM data.

(4) RAM guard function

This prevents rewriting of data in RAM due to incorrect CPU operations.

(5) SFR guard function

This prevents rewriting of data in the SFRs due to incorrect CPU operations.

(6) Illicit memory access detection

This detects illicit accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(7) Guard function of invalid memory access detection control register (IAWCTL)

This prevents rewriting of the invalid memory access detection control register due to incorrect CPU operations.

(8) Frequency detection

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

(9) Testing of the A/D converter

This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANLxx), temperature sensor output voltage, and internal reference voltage.

(10) Detection of the digital output signal level of the I/O pins

This is used to read the output level of an I/O pin when the pin is in the output mode.

(11) UART loopback

This is used to confirm that the transmit data is output normally by shutting off the TXDn and RXDn pins of UARTn and the TXDAm and RXDAm pins of UARTAm from the outside and connecting them within the MCU to loop back the output from the transmission shift register to the reception shift register.

Remark 1. $n = 0$ to 2, $m = 0$

Remark 2. For usage examples of the safety functions complying with the IEC60730 and IEC61508 safety standards, refer to the **application notes IEC60730/60335 Self Test Library for RL78 MCU**.

24.2 Registers for Controlling the Safety Functions

The following registers are used to control the safety functions.

- Flash memory CRC control register (CRC0CTL)
- Flash memory CRC operation result register (PGCRCL)
- CRC input register (CRCIN)
- CRC data register (CRCD)
- Code flash memory guard register (GFLASH0)
- Data flash memory guard register (GFLASH1)
- Flash security area guard register (GFLASH2)
- RAM parity error control register (RPECTL)
- Invalid memory access detection control register (IAWCTL)
 - RAM guard function
 - SFR guard function
 - Illicit memory access detection
- Guard register of IAWCTL register (GIAWCTL)
- Timer input select register 0 (TIS0)
- A/D test register (ADTES)
- Analog input channel specification register (ADS)
- Port mode select register (PMS)
- UART loopback select register (ULBS)

The content of each register is described in **24.3 Operation of Safety Functions**.

24.3 Operation of Safety Functions

24.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using a CRC to do it. The high-speed CRC provided in the RL78/G22 can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 512 μ s@32 MHz with 64-Kbyte flash memory).

The CRC generator polynomial used complies with " $X^{16} + X^{12} + X^5 + 1$ " of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Caution **The CRC operation result might differ during on-chip debugging because the monitor program is allocated.**

Remark The operation result is different between the high-speed CRC and the general-purpose CRC, because the general-purpose CRC operates in LSB first order.

24.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range. The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 1 Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0
CRC0EN		Control of CRC ALU operation						
0		Stops the operation.						
1		Starts the operation according to HALT instruction execution.						
FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range		
0	0	0	0	0	0	0000H to 03FFBH (16 Kbytes - 4 bytes)		
0	0	0	0	0	1	0000H to 07FFBH (32 Kbytes - 4 bytes)		
0	0	0	0	1	0	0000H to 0BFFBH (48 Kbytes - 4 bytes)		
0	0	0	0	1	1	0000H to 0FFFH (64 Kbytes - 4 bytes)		
Other than the above						Setting prohibited		

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

Caution Be sure to clear bit 6 to 0.

24.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register holds the high-speed CRC operation results. The PGCRCL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 24 - 2 Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H

After reset: 0000H

R/W: R/W

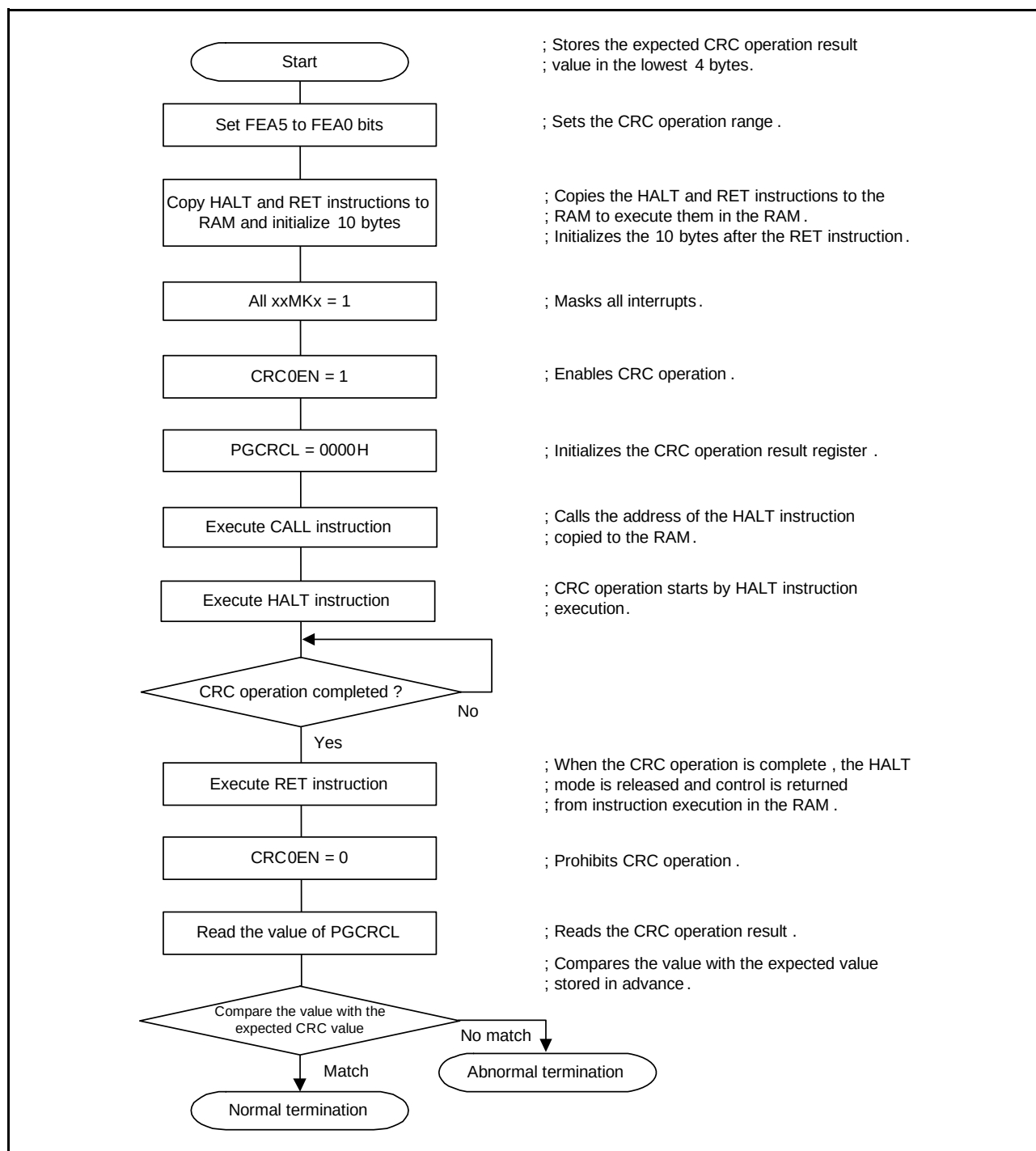
Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0
	PGCRC15 to PGCRC0		High-speed CRC operation results					
	0000H to FFFFH		Holds the high-speed CRC operation results.					

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 24 - 3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 24 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.

Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.

Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area.

Be sure to execute the HALT instruction in the RAM area.

The expected CRC value can be calculated by using the CS+ integrated development environment. See the CS+ Integrated Development Environment user's manual for details.

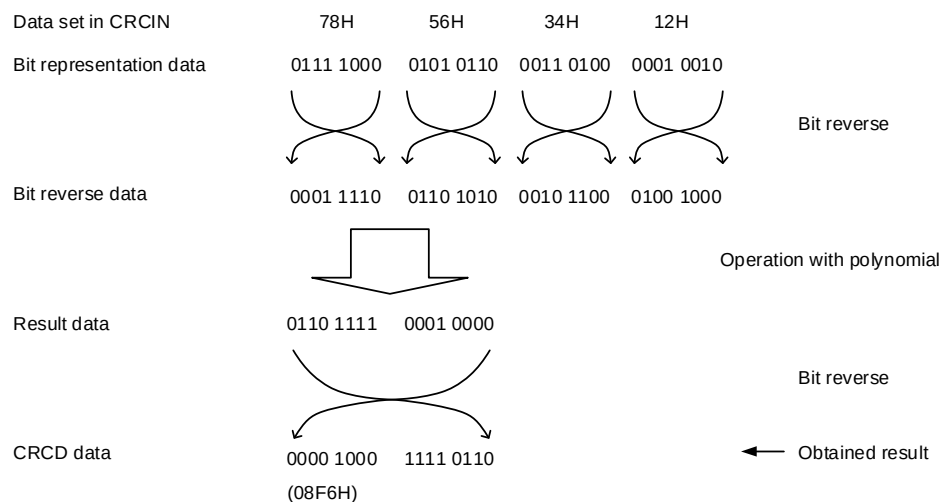
24.3.2 CRC operation (general-purpose CRC)

To guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

The general-purpose CRC handles CRC operation as a peripheral module while the CPU is operating. The general-purpose CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). The CRC calculation function in the HALT mode can be used only during DMA transfer.

The general-purpose CRC can operate either in the main system clock operation mode or in the subsystem clock operation mode.

The CRC generator polynomial used is " $X^{16} + X^{12} + X^5 + 1$ " of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

24.3.2.1 CRC input register (CRCIN)

The CRCIN is an 8-bit register to set the CRC operation data of the general-purpose CRC. The possible setting range is 00H to FFH. The CRCIN register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 4 Format of CRC Input Register (CRCIN)

Address: FFFACH
After reset: 00H
R/W: R/W

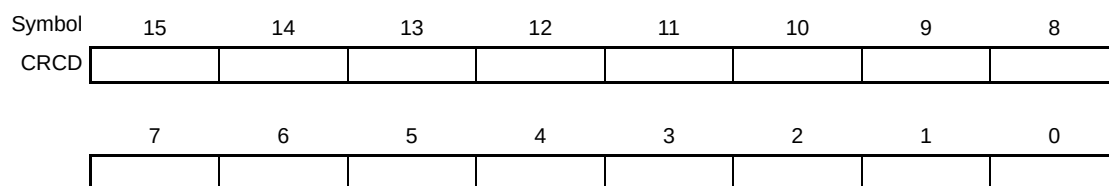
Symbol	7	6	5	4	3	2	1	0
CRCIN								
Bits 7 to 0	Function							
00H to FFH	Data input							

24.3.2.2 CRC data register (CRCD)

This register holds the CRC operation result of the general-purpose CRC. The setting range is 0000H to FFFFH. After one clock cycle of the CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register was written to, the CRC operation result is stored in the CRCD register. The CRCD register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 24 - 5 Format of CRC Data Register (CRCD)

Address: F02FAH
After reset: 0000H
R/W: R/W

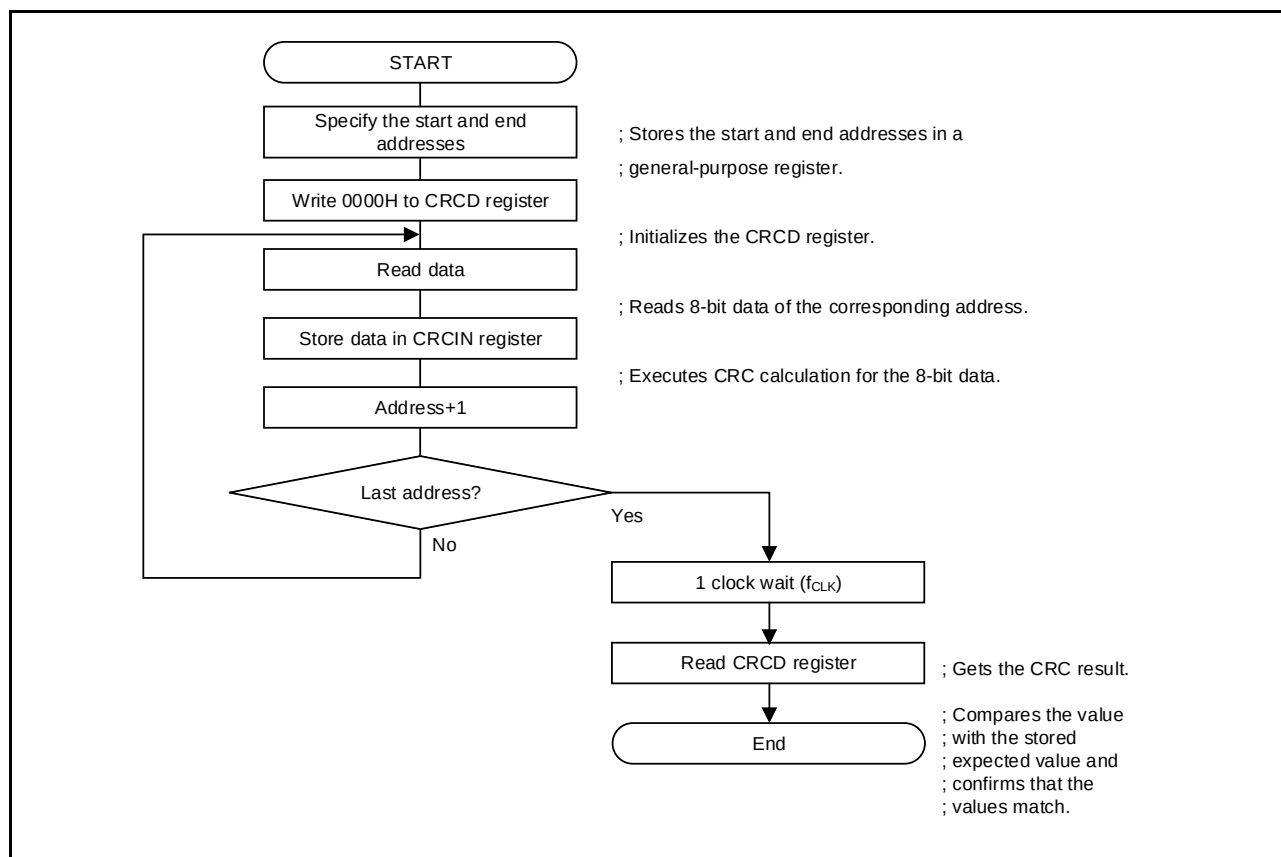


Caution 1. Read the value written to the CRCD register before writing to the CRCIN register.

Caution 2. If conflict between writing a value and storing the operation result in the CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 24 - 6 CRC Operation Function (General-Purpose CRC)



24.3.3 Flash memory guard function

To ensure safe operation, the IEC60730 standard requires protecting flash memory from rewriting of its data due to incorrect CPU operations.

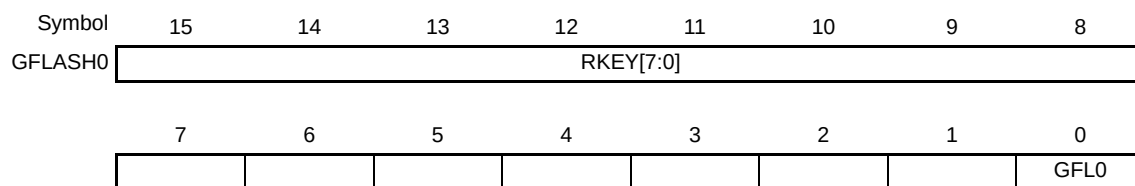
The RL78/G22 has functionality to protect data in the code flash memory, data flash memory, and the security area in the flash memory. Enabling this function disables writing to the protected area of the flash memory. Reading from the protected area is possible.

24.3.3.1 Code flash memory guard register (GFLASH0)

This register is used to protect the code flash memory against being rewritten. To allow rewriting of the code flash memory, set the GFLASH0.GFL0 bit to 0. To protect the code flash memory, set the GFLASH0.GFL0 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the code flash memory. The GFLASH0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 24 - 7 Format of Code Flash Memory Guard Register (GFLASH0)

Address: F0488H
After reset: 0000H
R/W: R/W



The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH0 register. When rewriting the GFL0 bit, set RKEY[7:0] to 30H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

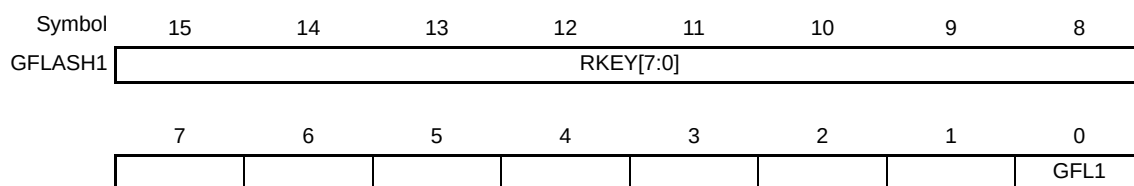
GFL0	Control of rewriting the code flash memory
0	Disables protection of the code flash memory (rewriting is allowed).
1	Enables protection of the code flash memory (rewriting is not allowed).

24.3.3.2 Data flash memory guard register (GFLASH1)

This register is used to protect the data flash memory against being rewritten. To allow rewriting of the data flash memory, set the GFLASH1.GFL1 bit to 0. To protect the data flash memory, set the GFLASH1.GFL1 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the data flash memory. The GFLASH1 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 24 - 8 Format of Data Flash Memory Guard Register (GFLASH1)

Address: F048AH
After reset: 0000H
R/W: R/W



The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH1 register. When rewriting the GFL1 bit, set RKEY[7:0] to C5H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

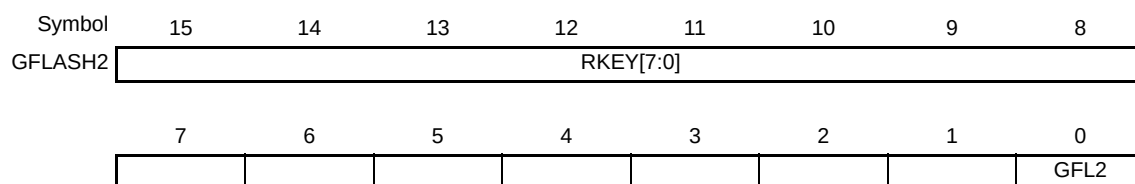
GFL1	Control of rewriting the data flash memory
0	Disables protection of the data flash memory (rewriting is allowed).
1	Enables protection of the data flash memory (rewriting is not allowed).

24.3.3.3 Flash security area guard register (GFLASH2)

This register is used to protect the flash memory's security area, which holds the security settings. To allow rewriting of the flash memory's security area, set the GFLASH2.GFL2 bit to 0.

Figure 24 - 9 Format of Flash Security Area Guard Register (GFLASH2)

Address: F048CH
After reset: 0000H
R/W: R/W



The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH2 register. When rewriting the GFL2 bit, set RKEY[7:0] to 9AH and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GFL2	Control of rewriting the flash memory's security area
0	Disables protection of the security area in the flash memory (rewriting is allowed).
1	Enables protection of the security area in the flash memory (rewriting is not allowed).

24.3.4 RAM parity error detection

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/G22's RAM. By using this RAM parity error detection, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

24.3.4.1 RAM parity error control register (RPECTL)

This register is used to control the parity error generation check bit and reset generation due to parity errors.

The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 10 Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF
	RPERDIS	Parity error reset mask flag						
	0	Enables parity error resets.						
	1	Disables parity error resets.						
	RPEF	Parity error status flag						
	0	No parity error has occurred.						
	1	A parity error has occurred.						

Caution The parity bit is appended when data is written, and the parity is checked when the data is read. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.
The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

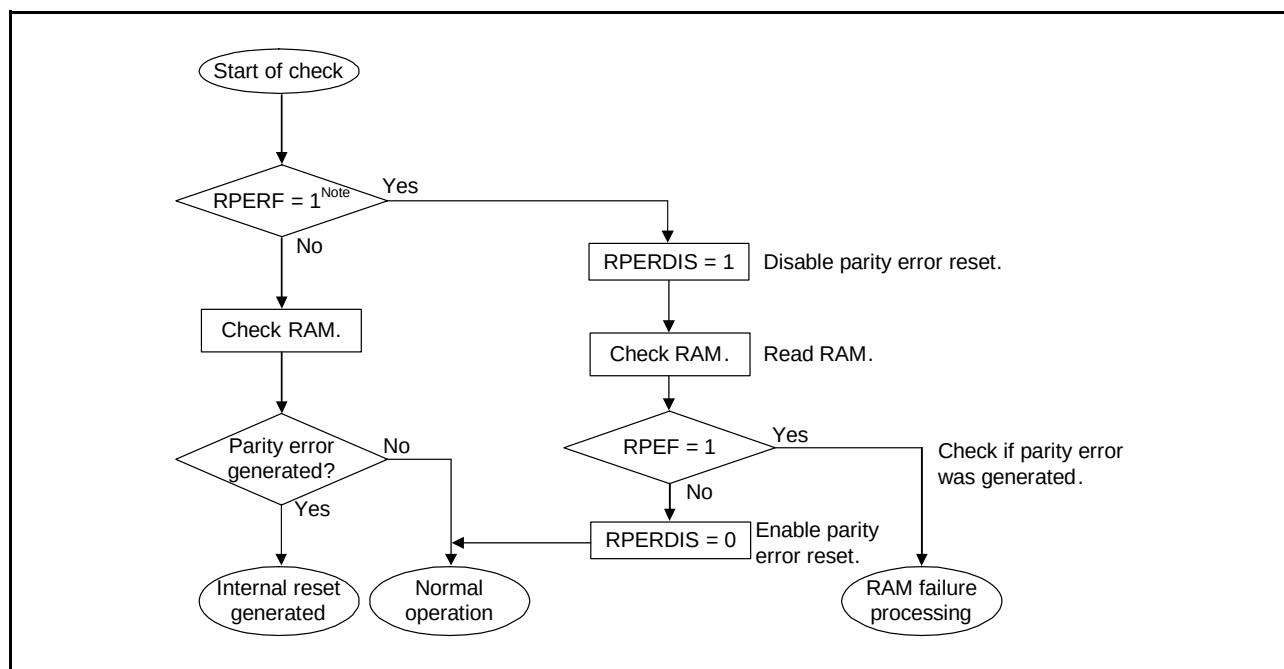
Remark 1. The parity error reset is enabled by default (RPERDIS = 0).

Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.

Remark 3. The RPEF flag in the RPECTL register is set (1) when the RAM parity error occurs and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

Remark 4. The general registers are not included for RAM parity error detection.

Figure 24 - 11 Flowchart of RAM Parity Check



Note To check the state following an internal reset triggered by a RAM parity error, see **Section 21 Reset Function**.

24.3.5 RAM guard function

To guarantee safe operation, the IEC61508 standard requires important data stored in the RAM to be protected even if a CPU malfunction occurs. The RL78/G22 has functionality to protect data in the specified memory space. Enabling this function disables writing to the specified area of the RAM. Reading from the specified area is possible.

24.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses, and protection of the RAM and SFRs. Use the GRAM1 and GRAM0 bits to protect the RAM. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 12 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	Protected area in the RAM
0	0	Disabled. Writing to the RAM is allowed.
0	1	128 bytes from the base address of the RAM
1	0	256 bytes from the base address of the RAM
1	1	512 bytes from the base address of the RAM

Remark The protection of the RAM by this register is only effective against accesses from the CPU, data transfer controller (DTC), and SNOOZE mode sequencer (SMS).

24.3.6 SFR guard function

To guarantee safe operation, the IEC61508 standard requires important data stored in the SFRs to be protected even if a CPU malfunction occurs. The RL78/G22 provides functionality to protect the data in the control registers for use with the ports, interrupts, clock control, voltage detection, and RAM parity error detection. Enabling this function disables writing to the protected area of the SFRs. Reading from the protected area is possible.

24.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the GPORT, GINT, and GCSC bits to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 13 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Protection of the port control registers
0	Disabled. Reading from and writing to the port control registers are allowed.
1	Enabled. Writing to the port control registers is not allowed. Reading from them is allowed. [Protected SFRs] PMxx, PUxx, PIMxx, POMxx, PMCAxx, PMCTxx, PFOE1, PDIDISxx

GINT	Protection of the interrupt control registers
0	Disabled. Reading from and writing to the interrupt control registers are allowed.
1	Enabled. Writing to the interrupt control registers is not allowed. Reading from them is allowed. [Protected SFRs] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Protection of the clock, voltage detector, and RAM parity error detection control registers
0	Disabled. Reading from and writing to the clock, voltage detector, and RAM parity error detection control registers are allowed.
1	Enabled. Writing to the clock, voltage detector, and RAM parity error detection control registers is not allowed. Reading from them is allowed. [Protected SFRs] CMC, CSC, OSTs, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, CKSEL, PRRx, MOCODIV, WKUPMD

24.3.7 Illicit memory access detection

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The RL78/G22 provides functionality to trigger a reset when an invalid memory area is accessed. Access to the areas indicated as "Not allowed" in **Figure 24 - 14** is detected as illicit.

Figure 24 - 14 Illicit Access Areas

		Whether or not access is allowed		
		Read	Write	Instruction fetch
FFFFFH	Special function registers (SFRs) 256 bytes	Allowed	Allowed	Not allowed
FFF00H FFEFFH	General-purpose registers 32 bytes			Allowed
FFEE0H FFEDFH	RAM		Not allowed	
FEF00H				Mirror
	Data flash memory		Allowed	Allowed
F1000H F0FFFH	Reserved			Not allowed
F0800H F07FFH	Special function registers (2nd SFRs) 2 Kbytes		Not allowed	Allowed
F0000H EFFFFH	Reserved			Not allowed
EE000H EDFFFH	Reserved		Not allowed	
yyyyyH xxxxxH	Code flash memory ^{Note}	Allowed	Allowed	
00000H				

See the next page for the details of Note in the figure.

Note The following table lists the capacity and address of the code flash memory and RAM, and the lowest address of the area to be detected as illicit when accessed of each product.

Products	Code flash memory (00000H to xxxxxH)	Lowest address of the area to be detected as illicit when accessed for reading or instruction fetching (yyyyyH)
R7F102GxC (x = 4, 6, 7, 8, A, B, C, E, F, G)	32768 × 8 bits (00000H to 07FFFH)	08000H
R7F102GxE (x = 4, 6, 7, 8, A, B, C, E, F, G)	65536 × 8 bits (00000H to 0FFFFH)	10000H

24.3.7.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the IAWEN bit to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 15 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN ^{Note}	Control of illicit memory access detection
0	Disables the detection of illicit memory accesses.
1	Enables the detection of illicit memory accesses.

Note Only writing 1 to the IAWEN bit has an effect. Writing 0 to it has no effect after the IAWEN bit has been set to 1.

Remark When WDTON = 1 (watchdog timer operation enabled) is set in the option byte (000C0H), the illicit memory access detection is enabled even if IAWEN = 0.

24.3.8 Guard function of invalid memory access detection control register

To ensure safe operation, the IEC60730 standard requires the setting for enabling or disabling illicit memory access detection to be protected from being rewritten even if a CPU malfunction occurs.

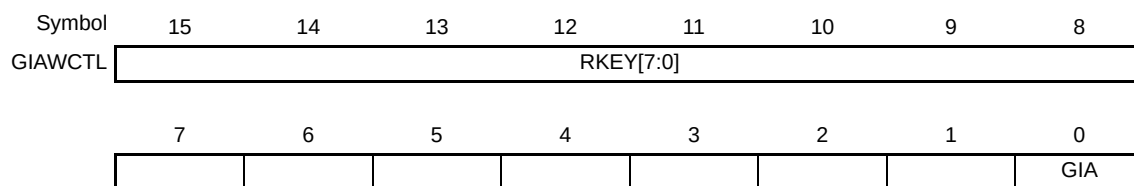
The RL78/G22 provides functionality to protect the invalid memory access detection control register (IAWCTL) against being rewritten. Enabling the protection of the invalid memory access detection control register disables writing to the given register. Reading from the protected register is possible.

24.3.8.1 Guard register of IAWCTL register (GIAWCTL)

This register is used to protect the setting for enabling or disabling the illicit memory access detection. To allow rewriting of the invalid memory access detection control register (IAWCTL), set the GIAWCTL.GIA bit to 0 to disable protection of the IAWCTL register. The GIAWCTL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 24 - 16 Format of Guard Register of IAWCTL Register (GIAWCTL)

Address: F048EH
After reset: 0000H
R/W: R/W



The RKEY[7:0] bits contain the key code to control rewriting of the GIAWCTL register. When rewriting the GIA bit, set RKEY[7:0] to C4H and then write to all 16 bits of this register at once. The RKEY[7:0] bits return 00H when read.

GIA	Control of rewriting the IAWCTL register
0	Disables protection of the IAWCTL register (rewriting is allowed).
1	Enables protection of the IAWCTL register (rewriting is not allowed).

24.3.9 Frequency detection

The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) to measure the pulse width of the input signal to channel 5 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clock frequencies to be compared>

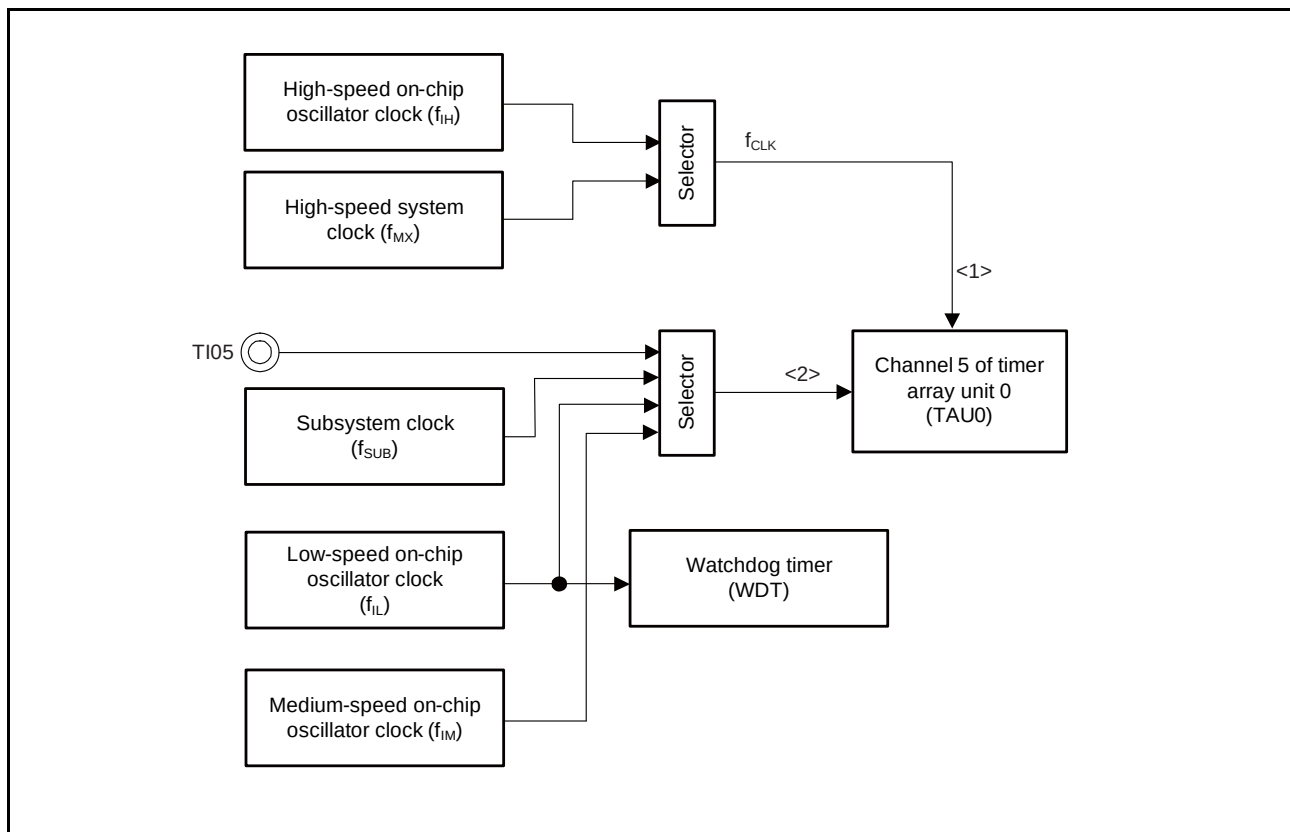
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 5 of the timer array unit

- Timer input to channel 5 (TI05)
- Low-speed on-chip oscillator clock (f_{IL})
- Subsystem clock (f_{SUB})
- Medium-speed on-chip oscillator clock (f_{IM})

Figure 24 - 17 Configuration of Frequency Detection Function



If the results of input pulse interval measurement are abnormal, the clock frequency is considered to be abnormal. For details on the input pulse interval measurement, see **7.8.4 Operation for input pulse interval measurement**.

24.3.9.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 5 of the timer array unit 0 (TAU0). The TIS0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 18 Format of Timer Input Select Register 0 (TIS0)

Address: F0074H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from the ELC
0	1	1	Medium-speed on-chip oscillator peripheral clock (fimp)
1	0	0	Low-speed on-chip oscillator clock (fil)
1	0	1	Subsystem clock (fsub)
Other than the above			Setting prohibited

24.3.10 Testing of the A/D converter

The IEC60730 standard mandates testing of the A/D converter. This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANLxx), temperature sensor output voltage, and internal reference voltage. For details on the method of checking, refer to the **application note (R01AN5607) Safety Function (A/D test)**.

Use the following procedure to check the analog multiplexer.

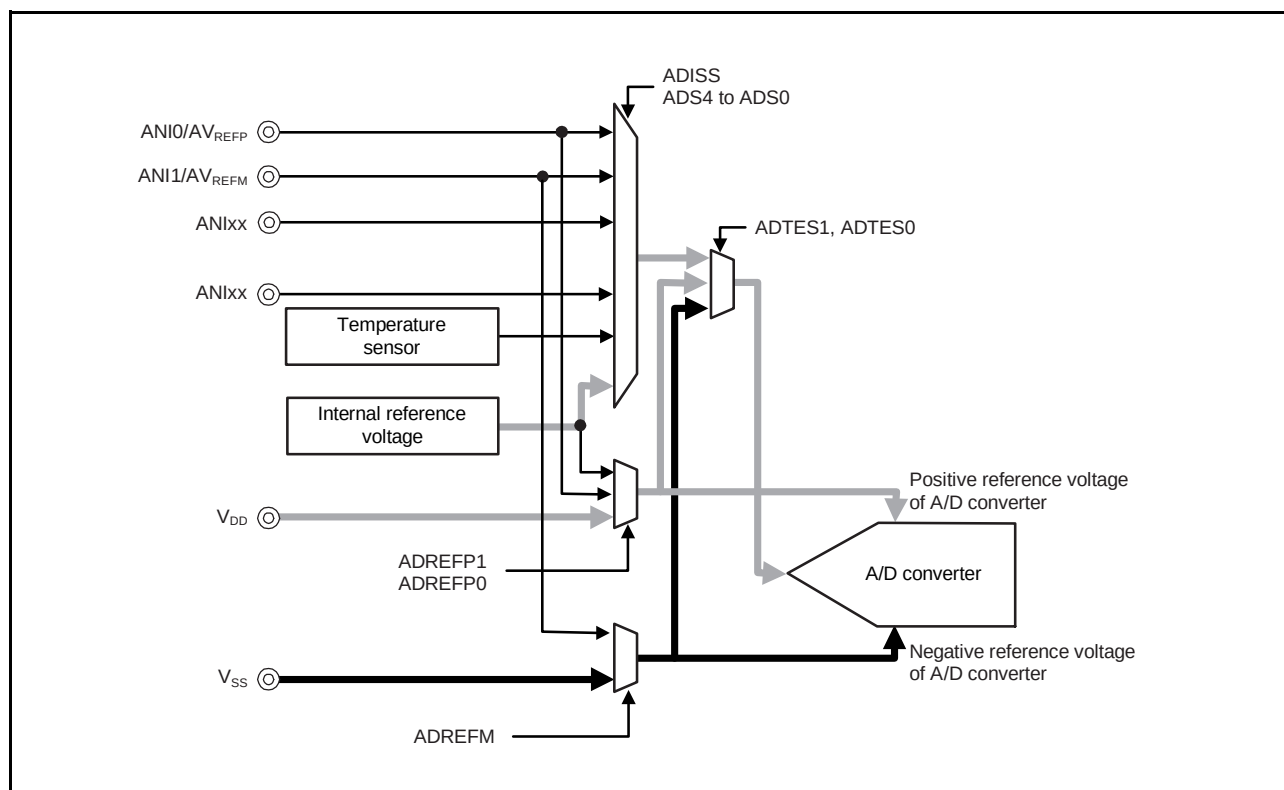
- <1> Select the ANLxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
- <2> Perform A/D conversion for the ANLxx pin (conversion result 1-1).
- <3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)
- <4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
- <5> Select the ANLxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
- <6> Perform A/D conversion for the ANLxx pin (conversion result 1-2).
- <7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)
- <8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
- <9> Select the ANLxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
- <10> Perform A/D conversion for the ANLxx pin (conversion result 1-3).
- <11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
- <12> Check that the A/D conversion result 2-1 is all zero and the A/D conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

Remark 2. The results of conversion might include an error. Consider an appropriate level of error in comparison of the results of conversion.

Figure 24 - 19 Configuration of Testing of the A/D converter



24.3.10.1 A/D test register (ADTES)

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channels (ANLxx), temperature sensor output voltage, or internal reference voltage as the target of A/D conversion. When testing the A/D converter, specify the following settings:

- Select the negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select the positive reference voltage as the target of A/D conversion for full-scale measurement.

The ADTES register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 20 Format of A/D Test Register (ADTES)

Address: F0013H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANLxx/TSCAP voltage of the CTSU/temperature sensor output voltage/internal reference voltage (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected with the ADREFM bit in the ADM2 register)
1	1	Positive reference voltage (selected with the ADREFP1 or ADREFP0 bit in the ADM2 register)
Other than the above		Setting prohibited

Caution Be sure to clear bits 7 to 2 to 0.

24.3.10.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set the A/D test register (ADTES) to 00H when measuring ANIxx, temperature sensor output voltage, or internal reference voltage. The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 21 Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

<Select mode (ADMD = 0)>

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin ^{Note 1}
0	1	0	0	0	1	ANI17	P00/ANI17 pin ^{Note 2}
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	1	1	1	0	—	TSCAP voltage of the CTSU
1	0	0	0	0	0	—	Temperature sensor output voltage
1	0	0	0	0	1	—	Internal reference voltage
Other than the above						Setting prohibited	

Note 1. The ANI16 pin is not available in the 36-, 40-, 44-, and 48-pin products.

Note 2. The ANI17 pin is not available in the 36-, 40-, 44-, and 48-pin products.

(Cautions are listed in the next page.)

- Caution 1. Be sure to clear bits 6 and 5 to 0.
- Caution 2. Select input mode for the ports which are set to analog input with the PMCA registers, using the port mode registers 0, 2, 12, and 14 (PM0, PM2, PM12, and PM14).
- Caution 3. Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control A registers 0, 2, 12, and 14 (PMCA0, PMCA2, PMCA12, and PMCA14).
- Caution 4. Only rewrite the value of the ADISS bit while the conversion operation is stopped (ADCS = 0, ADCE = 0).
- Caution 5. If using AVREFF as the positive reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.
- Caution 6. If using AVREFM as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.
- Caution 7. When ADISS is 1, the internal reference voltage cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For a detailed setting flow, see 12.7.4 Setup when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected (example for software trigger mode and one-shot conversion mode).
- Caution 8. If a transition is made to STOP mode or a transition is made to HALT mode during CPU operation with subsystem clock, do not set ADISS to 1. When ADISS is 1, the A/D converter reference voltage current (IADREF) shown in 34.3.2 Characteristics of the supply current is added.

24.3.11 Detection of the digital output signal level of the I/O pins

The IEC60730 standard mandates checking that the I/O function is operating correctly. The RL78/G22 provides functionality to read the output level of an I/O pin when the pin is in the output mode.

24.3.11.1 Port mode select register (PMS)

This register is used to select whether to read the output latch value of a port or read the output level of a pin when the pin is in the output mode (PMmn bit of port mode register (PMm) is 0). The PMS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 22 Format of Port Mode Select Register (PMS)

Address: F007BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0
PMS0	Selection of data to be read when pin is in output mode							
0	Reads the value of the Pmn register.							
1	Reads the digital output level of the pin.							

Caution 1. Be sure to clear bits 7 to 1 to 0.

Caution 2. Do not rewrite a port register (Pxx) with a 1-bit memory manipulation instruction when the setting of the PMS0 bit in the PMS register is 1. Use an 8-bit memory manipulation instruction to rewrite a port register (Pxx).

Caution 3. The setting of this register has no effect on the input port pins (P123, P124, and P137) and output port pin (P130).

Remark m = 0 to 7, 12, 14
n = 0 to 7

Caution 4. The setting of this register has no effect when the settings of the PMCAmn and PMCTmn bits are both 1. To read the digital output level of a pin with the PMCAmn bit and PMCTmn bit both set to 1, first clear the PMCAmn and PMCTmn bits to 0 and then set the PMS0 bit to 1.

Remark m = 0 to 3, 5, 7, 12, 14
n = 0 to 7

Caution 5. The setting of this register has no effect when the setting of the PDIDISmn bit is 1. To read the digital output level of a pin with the PDIDISmn bit set to 1, first clear the PDIDISmn bit to 0 and then set the PMS0 bit to 1.

Remark m = 0, 1, 5, 7, 12
n = 0 to 5, 7

24.3.12 UART loopback

The IEC60730 standard recommends to diagnose abnormalities of external interfaces (communications).

The UART loopback is used to confirm the normal output of the UART transmit data by shutting off the Rx pin from the outside and connecting it within the MCU to loop back the output from the transmission shift register to the reception shift register.

When the UART loopback is selected, the transmit data from the Tx pin is controlled by the port functions so that it does not affect its communication partner.

- Communication with negative logic
The port that also uses the Tx pin is set to the input mode (PMxx = 1) and an on-chip pull-up resistor is connected (PUxx = 1) to retain the setting of 1.
- Communication with positive logic
0 (PMxx = 0, Pxx = 0) is output from the port that also uses the Tx pin.

24.3.12.1 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback. This register has respective bits for independently controlling each UART channel. Setting the bit corresponding to each channel to 1 will select the UART loopback and loop back the output of the transmission shift register to the reception shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 23 Format of UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	3	<2>	<1>	<0>
ULBS	0	0	0	ULBS4	0	ULBS2	ULBS1	ULBS0

ULBS4	UART loopback selection Note 1
0	Inputs the state of the RxDA0 pin of serial interface UAR0 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS2	UART loopback selection
0	Inputs the state of the RxDA2 pin of serial array unit UAR2 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS1	UART loopback selection
0	Inputs the state of the RxDA1 pin of serial array unit UAR1 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS0	UART loopback selection Note 2
0	Inputs the state of the RxDA0 pin of serial array unit UAR0 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

(Notes, Caution, and Remark are listed on the next page.)

- Note 1.** For UARTA0, set the PFOE14 bit of the port function output enable register 1 (PFOE1) to 1 when using the loopback.
- Note 2.** For UART0, set the PFOE10 bit of the port function output enable register 1 (PFOE1) to 1 when using the loopback.
- Caution** **Be sure to clear bits 7, 6, 5, and 3 to 0.**
- Remark** The RxDA0 pin is only available in the 36-pin to 48-pin products.
The RxD1 pin is only available in the 20-pin to 48-pin products.
The RxD2 pin is only available in the 30-pin to 48-pin products.

Section 25 Security Functions

25.1 True Random Number Generator

25.1.1 Function of the true random number generator

The true random number generator generates 32-bit random number seeds (which are true random numbers).

25.1.2 Registers for controlling the true random number generator

The following registers are used to control the true random number generator.

- Random number seed command register 0 (TRNGSCR0)
- Random number seed data register (TRNGSDR)

25.1.2.1 Random number seed command register 0 (TRNGSCR0)

The TRNGSCR0 register controls operation of the true random number generator. Setting the TRNGST bit to 1 after having set the TRNGEN bit to 1 starts the generation of a random number seed. When the true random number generator finishes generating the random number seed, the TRNGRDY bit is set to 1.

Since the TRNGST bit serves as the trigger for starting the generation of a random number seed, it is cleared to 0 immediately after 1 having been written to it.

The TRNGSCR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 25 - 1 Format of Random Number Seed Command Register 0 (TRNGSCR0)

Address: F0542H

After reset: 00H

R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
TRNGSCR0	TRNGRDY	0	0	0	TRNGEN	TRNGST	0	0
	TRNGRDY	Random number seed generation status flag						
	0	A random number seed has not been generated or four rounds of reading from the TRNGSDR register have been completed.						
	1	A random number seed has been generated.						
	TRNGEN	Control over operation of the true random number generator						
	0	Stops the true random number generator.						
	1	Enables the true random number generator.						
	TRNGST	Trigger to start generating a random number seed						
	0	The trigger is inactive.						
	1	Starts generation of a random number seed.						

25.1.2.2 Random number seed data register (TRNGSDR)

The TRNGSDR register is an 8-bit register that holds the bytes of random number seeds generated by the true random number generator. The random number seed can be read from this register after the TRNGRDY bit is set to 1. As a random number seed consists of 32 bits, four rounds of access to the register are required for each seed. Bit 7 (TRNGRDY) of the TRNGSCR0 register is cleared to 0 following the four rounds of access.

The TRNGSDR register can be read by an 8-bit memory manipulation instruction.

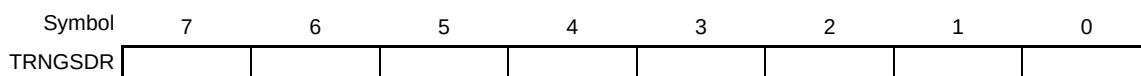
The value of this register following a reset is 00H.

Figure 25 - 2 Format of Random Number Seed Data Register (TRNGSDR)

Address: F0540H

After reset: 00H

R/W: R

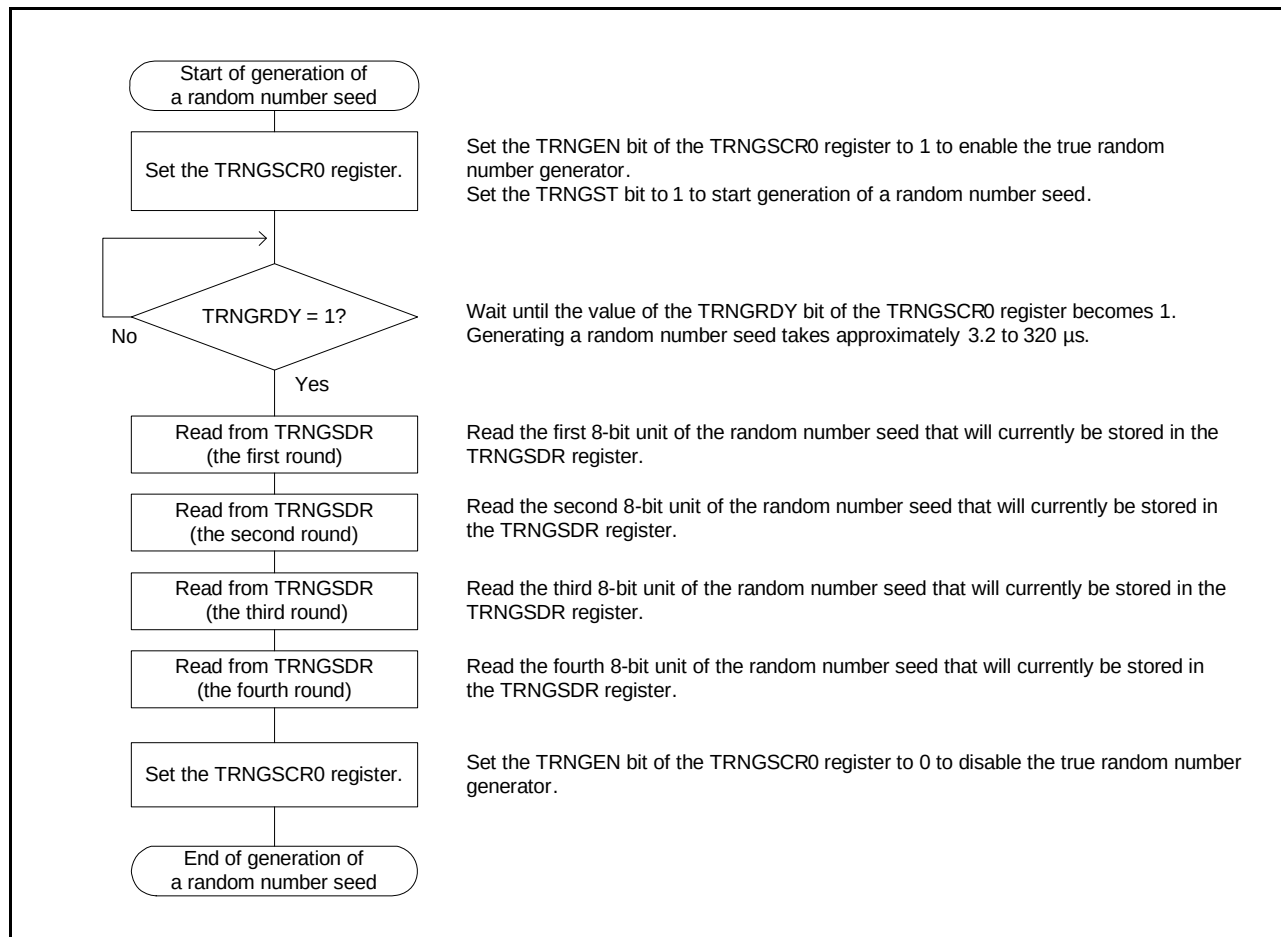


Caution When the value of the TRNGRDY bit is 0, the value in the TRNGSDR register is 00H.

25.1.3 Operations of the true random number generator

Figure 25 - 3 shows the procedure for using the true random number generator to generate a random number seed.

Figure 25 - 3 Procedure for Using the True Random Number Generator to Generate a Random Number Seed



25.2 Flash Read Protection

25.2.1 Function of flash read protection

The flash read protection function can be used to protect a specified range of the code flash memory area against read access by the CPU, DTC, or SMS. Note that fetching of instructions in the specified range by the CPU is still possible.

25.2.2 Setting of flash read protection

The settings for flash read protection are made through serial programming by using a flash memory programmer or through self-programming in the extra area.

After the settings have been made, read access to addresses in the range of blocks in the code flash memory area between the start and end of the flash read protection becomes impossible.

In addition, disabling changing of the flash read protection settings makes changes to the blocks where flash read protection starts and ends impossible. However, disabling changing of the flash read protection settings can be released by using a flash memory programmer. **Table 25 - 1** describes the settings for flash read protection and their functions.

Table 25 - 2 describes the method of setting flash read protection.

Table 25 - 1 Settings for Flash Read Protection and their Functions

Item to Be Set	Function
Block where flash read protection starts	Specifies the number of the block where the read-access disabled area starts. The specifiable values are in the range from the block numbered 001H to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection starts is part of the read-access disabled area. Setting 000H as the block where protection starts is prohibited. The initial setting is outside the specifiable range.
Block where flash read protection ends	Specifies the number of the block where the read-access disabled area ends. The specifiable values are in the range from the number of the block where the protection starts to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection ends is part of the read-access disabled area. The initial setting is outside the specifiable range.
Disabling changing of the flash read protection settings	Fixes the settings of the blocks where flash read protection starts and ends. When the "enabled" setting is made, changes to the blocks where protection starts and ends are not possible.

For details on the relationship between the addresses and block numbers, refer to **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Table 25 - 2 Method of Setting Flash Read Protection

Item to Be Set	Method of Setting	Method of Changing
Block where flash read protection starts	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection starts is not adjustable while changing of the flash read protection settings is disabled.
Block where flash read protection ends	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection ends is not adjustable while changing of the flash read protection settings is disabled.
Disabling changing of the flash read protection settings	Using a flash memory programmer or self-programming.	Fixing of the flash read protection settings can be released by using a flash memory programmer. ^{Note} If you do so, the values for the start and end blocks are initialized.

Note Release from the fixed setting is only possible when erasure of blocks is not prohibited, rewriting of boot area is not prohibited, and the code and data flash memory areas are blank.

Caution 1. The settings for flash read protection in the extra area are not readable. To confirm that the settings for flash read protection are in place, read from the read-access disabled area and confirm that FFH is returned.

Caution 2. To specify the read-access disabled area for flash read protection, be sure to specify the numbers of both the block where protection starts and the block where it ends.

Caution 3. Reading from the read-access disabled area by using an on-chip debugger is also impossible. This means that program code allocated to the read access-disabled area cannot be debugged by using the on-chip debugger. Therefore, only make the settings for flash read protection after having debugged the program code in the protected areas.

Caution 4. When a part of boot cluster 0 or boot cluster 1 is to be set as a part of the read-access disabled area, boot swapping may cause data in the read-access disabled area to be swapped with data in the read access-enabled area. To prevent this, when setting a part of boot cluster 0 or boot cluster 1 as part of the read-access disabled area, make the setting for prohibiting the rewriting of boot area so as to prohibit boot swapping itself.

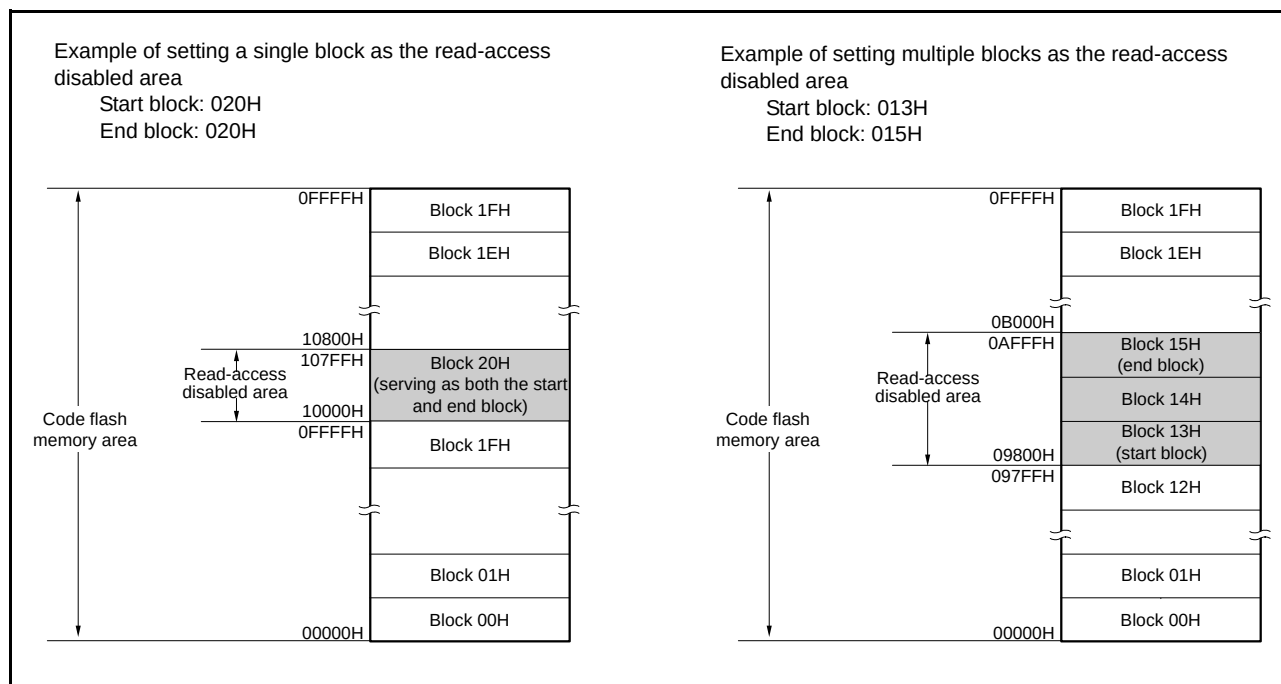
25.2.3 Operation

Reading by the CPU, DTC, or SMS from the area to which read access has been disabled with the use of flash read protection always returns FFH.

Fetching of instructions in the read-access disabled area by the CPU is still possible. Note that even if program code is to be executed from the read-access disabled area, it is unable to read data that have been placed in the read-access disabled area. Place data for use with code to be executed from the read-access disabled area in areas that are not protected.

Figure 25 - 4 shows examples of setting read-access disabled areas for the application of flash read protection.

Figure 25 - 4 Examples of Setting Read-Access Disabled Areas for the Application of Flash Read Protection



25.3 Unique ID

25.3.1 Function of a unique ID

A unique ID is a unique value that is allocated to an individual product and stored in the extra area.

A unique ID is entered for each product at the time of manufacturing the MCUs. Changing the entered ID is not possible.

The data length is 16 bytes (128 bits).

Caution **The value of a unique ID is not a random number.**

25.3.2 ASCII codes representing the product name

Product names are also stored as strings of ASCII codes in the extra area.

Since the product names include indicators of the number of pins and capacity of the flash memory for the given product, conditional branch processing on the basis of the product name is possible.

Table 25 - 3 shows the places of the unique ID, basic product name, and indicators of the number of pins and capacity of the flash memory in the memory map.

Table 25 - 3 Memory Map of the Unique ID and Product Name

Address	Item Name	Value to Be Entered
EFFC0H to EFFCFH	Unique ID	The unique value allocated to an individual product
EFFD5H	ASCII codes representing the product name	52H: "R"
EFFD6H		37H: "7"
EFFD7H		46H: "F"
EFFD8H		31H: "1"
EFFD9H		30H: "0"
EFFDAH		32H: "2"
EFFDBH		47H: "G"
EFFDCH		34H: "4" (16 pins) 36H: "6" (20 pins) 37H: "7" (24 pins) 38H: "8" (25 pins) 41H: "A" (30 pins) 42H: "B" (32 pins) 43H: "C" (36 pins) 45H: "E" (40 pins) 46H: "F" (44 pins) 47H: "G" (48 pins)
EFFDDH		43H: "C" (32-Kbyte flash memory) 45H: "E" (64-Kbyte flash memory)

Section 26 SNOOZE Mode Sequencer (SMS)

The SNOOZE mode sequencer sequentially handles 32 processes that have been set in advance. This function can be started by a signal from another peripheral function as the trigger even in the standby state. This means the sequencer can run through the processes independently of the CPU. Doing so with the CPU placed in the standby state reduces power consumption.

26.1 Functions of the SNOOZE Mode Sequencer

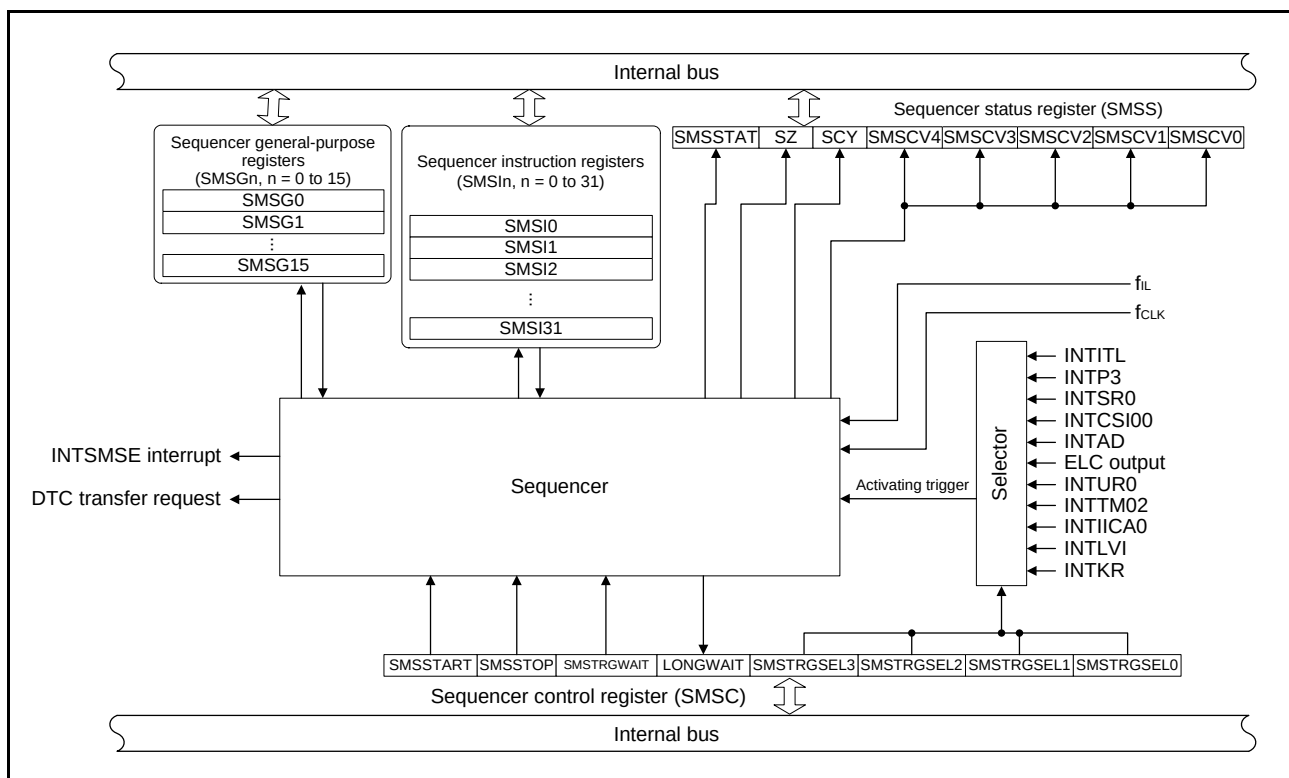
The SNOOZE mode sequencer can handle processing independently of the CPU and has the following functions.

- Sequentially handling a total of 32 processes with the use of desired commands from among 21 different ones
- Able to operate even if the CPU is in the standby state
- Waking the CPU up from the standby state
- Directly starting the data transfer controller (DTC) up
- Access to the RAM and special function register (SFR) area for the peripheral functions
- 16-bit addition and subtraction
- Branch processing
- Automatically switching the source clock for use in waiting to the low-speed on-chip oscillator clock to ensure long enough wait times

26.2 Configuration of the SNOOZE Mode Sequencer

Figure 26 - 1 is a block diagram of the SNOOZE mode sequencer.

Figure 26 - 1 Block Diagram of the SNOOZE Mode Sequencer



26.3 Registers for Controlling the SNOOZE Mode Sequencer

The following registers are used to control the SNOOZE mode sequencer.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Sequencer instruction registers p (SMSIp) (p = 0 to 31)
- Sequencer general-purpose registers n (MSGn) (n = 0 to 15)
- Sequencer control register (SMSC)
- Sequencer status register (SMSS)

26.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the SNOOZE mode sequencer is to be used, be sure to set bit 6 (SMSEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 26 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>
PER1	0	SMSEN	0	TML32EN	DTCEN	UTAEN	0	CTSUEN

SMSSEN	Control of supply of an input clock to the SNOOZE mode sequencer
0	Stops supply of an input clock. • The SFRs used by the SNOOZE mode sequencer cannot be written. • When an SFR used by the SNOOZE mode sequencer is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. • The SFRs used by the SNOOZE mode sequencer can be read and written.

Caution 1. Do not change the value of the SMSEN bit of the PER1 register from 1 to 0 while the SNOOZE mode sequencer is operating.

Caution 2. The functions that are mounted depend on the product. For details on the PER1 register, see the description in Section 6 Clock Generator.

26.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

Set bit 6 (SMSRES) of this register to 1 to reset the SNOOZE mode sequencer.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 26 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH

After reset: 00H

R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	<0>
PRR1	0	SMSRES	0	TML32RES	0	0	0	CTSURES
	SMSRES		Control resetting of the SNOOZE mode sequencer					
	0		The SNOOZE mode sequencer is released from the reset state.					
	1		The SNOOZE mode sequencer is in the reset state. • The SFRs for use with the SNOOZE mode sequencer are initialized.					

Caution Be sure to clear bits 7, 5, and 3 to 1 to 0.

26.3.3 Sequencer instruction registers p (SMSIp) (p = 0 to 31)

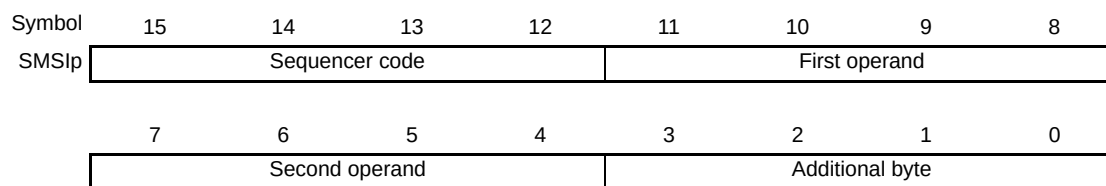
Each SMSIp register is a 16-bit register for holding a command to be handled by the SNOOZE mode sequencer. The commands are stored in the total of 32 registers from SMSI0 to SMSI31. The sequencer sequentially handles the commands in order from the SMSI0 register. The contents of each register are referred to as a sequencer code, which consists of an indicator of the processing, a first operand, a second operand, and an additional byte or nybble, the use of which depends on the sequencer code. For details on how the sequencer handles each form of processing, see **26.5 Commands for Use in Processing by the Sequencer**. In addition, after executing processing specified by an SMSIp register, the sequencer executes the processing specified either by the next SMSIp register or by the SMSIp register at a branch destination. The SMSIp register currently in use can be confirmed by reading the SMSCV0 to SMSCV4 bits of the sequencer status register (SMSS).

Table 26 - 1 shows the correspondences between the memory addresses of the SMSIp registers and values of the SMSCV0 to SMSCV4 bits. **Table 26 - 2** lists the types of processing specified by the SMSIp registers.

The SMSIp registers can be set by a 16-bit memory manipulation instruction.
The value of each SMSIp register following a reset is 0000H.

Figure 26 - 4 Format of Sequencer Instruction Registers p (SMSIp)

Address: F0380H, F0381H (SMSI0) to F03BEH, F03BFH (SMSI31)
After reset: 0000H
R/W: R/W



Remark p = 0 to 31

Table 26 - 1 Correspondences between the Memory Addresses of the SMSIp Registers and Values of the SMSCV0 to SMSCV4 Bits

SMSIp	Address	SMSCV[4:0]
SMSI15	F039EH, F039FH	01111B
SMSI14	F039CH, F039DH	01110B
SMSI13	F039AH, F039BH	01101B
SMSI12	F0398H, F0399H	01100B
SMSI11	F0396H, F0397H	01011B
SMSI10	F0394H, F0395H	01010B
SMSI9	F0392H, F0393H	01001B
SMSI8	F0390H, F0391H	01000B
SMSI7	F038EH, F038FH	00111B
SMSI6	F038CH, F038DH	00110B
SMSI5	F038AH, F038BH	00101B
SMSI4	F0388H, F0389H	00100B
SMSI3	F0386H, F0387H	00011B
SMSI2	F0384H, F0385H	00010B
SMSI1	F0382H, F0383H	00001B
SMSI0	F0380H, F0381H	00000B

SMSIp	Address	SMSCV[4:0]
SMSI31	F03BEH, F03BFH	11111B
SMSI30	F03BCH, F03BDH	11110B
SMSI29	F03BAH, F03BBH	11101B
SMSI28	F03B8H, F03B9H	11100B
SMSI27	F03B6H, F03B7H	11011B
SMSI26	F03B4H, F03B5H	11010B
SMSI25	F03B2H, F03B3H	11001B
SMSI24	F03B0H, F03B1H	11000B
SMSI23	F03AEH, F03AFH	10111B
SMSI22	F03ACH, F03ADH	10110B
SMSI21	F03AAH, F03ABH	10101B
SMSI20	F03A8H, F03A9H	10100B
SMSI19	F03A6H, F03A7H	10011B
SMSI18	F03A4H, F03A5H	10010B
SMSI17	F03A2H, F03A3H	10001B
SMSI16	F03A0H, F03A1H	10000B

Caution 1. Only set the SMSIp registers while the operation of the sequencer is stopped. Re-writing the SMSIp registers while the sequencer is handling the commands results in an undefined operation of the sequencer.

Caution 2. No register follows the SMSI31 register once the processing it defines has finished. Therefore, set the SMSI31 register for processing for termination command or interrupt plus termination command to stop processing by the sequencer, or for branch processing so that the processing at the branch destination register is run.

Table 26 - 2 Types of Processing Specified by the SMSIp Registers

Name of Processing	Operation ^{Note 4}	Sequencer Code	First Operand (4 Bits)	Second Operand (4 Bits)	Additional Byte (4 Bits)
8-bit data transfer 1	[MSGn + Byte] ← MSGm	0000	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	Byte ^{Note 2}
8-bit data transfer 2	MSGm ← [MSGn + Byte]	0001	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	Byte ^{Note 2}
16-bit data transfer 1	[MSGn + Byte] ← MSGm	0010	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	Byte ^{Note 2}
16-bit data transfer2	MSGm ← [MSGn + Byte]	0011	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	Byte ^{Note 2}
1-bit data setting	[MSGn + Byte].bit ← 1	0100	nth of MSGn ^{Note 1}	bit ^{Note 2}	Byte ^{Note 2}
1-bit data clearing	[MSGn + Byte].bit ← 0	0101	nth of MSGn ^{Note 1}	bit ^{Note 2}	Byte ^{Note 2}
1-bit data transfer	SCY ← [MSGn + Byte].bit	0110	nth of MSGn ^{Note 1}	bit ^{Note 2}	Byte ^{Note 2}
Word addition	MSGn, SCY ← MSGn + MSGm	0111	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	0000
Word subtraction	MSGn, SCY ← MSGn - MSGm	0111	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	0001
Word comparison	MSGn - MSGm	0111	nth of MSGn ^{Note 1}	mth of MSGm ^{Note 1}	0010
Logical shift right	SCY ← MSGn.0, MSGm.15 ← 0, MSGn.m-1 ← MSGn.m	0111	nth of MSGn ^{Note 1}	0000	0011
Branch 1 (SCY = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 1	1000	\$addr5 ^{Note 3}		0000
Branch 2 (SCY = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 0	1000	\$addr5 ^{Note 3}		0001
Branch 3 (SZ = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 1	1000	\$addr5 ^{Note 3}		0010
Branch 4 (SZ = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 0	1000	\$addr5 ^{Note 3}		0011
Wait	Holding processing pending for a certain period	1001	IM1		IM2
Conditional wait 1 (bit = 1)	SMSS[4:0] ← SMSS[4:0] if [MSGn + Byte].bit = 1	1010	nth of MSGn ^{Note 1}	Bit ^{Note 2}	Byte ^{Note 2}
Conditional wait 2 (bit = 0)	SMSS[4:0] ← SMSS[4:0] if [MSGn + Byte].bit = 0	1011	nth of MSGn ^{Note 1}	Bit ^{Note 2}	Byte ^{Note 2}
Termination	SMSS[4:0] ← 0, Stopping the sequencer	1111	0000	0000	0000
Interrupt plus termination	SMSS[4:0] ← 0, Stopping the sequencer after issuing an interrupt	1111	0000	0000	0001
DTC activation	Output of a DTC activating source signal	1111	0000	0000	0010

Note 1. Specify values in the range from 0 to 15 (from 0000B to 1111B) for n and m.

Note 2. Specify values in the range from 0 to 7 (from 0000B to 0111B) for the bytes.

Note 3. This is an 8-bit displacement value. Specify a relative address in the range from 0 to 31 (0000 0000B to 0001 1111B, 1111 1111B to 1110 0001B).

Note 4. For details on the terms, see 26.5 Commands for Use in Processing by the Sequencer.

26.3.4 Sequencer general-purpose registers n (MSGn) (n = 0 to 15)

Each MSGn register is a 16-bit general-purpose register for use with the sequencer and is used for the storage of data, judgement values for use in calculations, source and destination addresses for data manipulation, and so on. Note that the MSG0 and MSG15 registers have the fixed values 0000H and FFFFH, respectively.

The sequencer has direct access to the MSGn registers for several types of processing. In such cases, use the suffix n in MSGn as the internal data addresses to specify the desired MSGn registers for the sequencer.

The MSG0 to MSG15 registers can be read by a 16-bit memory manipulation instruction. The MSG1 to MSG14 registers can be set by a 16-bit memory manipulation instruction.

The value of each of the MSG1 to MSG14 registers following a reset is 0000H. The MSG0 and MSG15 registers have the fixed values 0000H and FFFFH, respectively.

Figure 26 - 5 Format of Sequencer General-Purpose Registers n (MSGn)

Address: F03C0H, F03C1H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
MSG0	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Address: F03C2H, F03C3H (MSG1) to F03DCH, F03DDH (MSG14)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
MSGn								
	7	6	5	4	3	2	1	0

Remark n = 1 to 14

Address: F03DEH, F03DFH
After reset: FFFFH
R/W: R

Symbol	15	14	13	12	11	10	9	8
MSG15	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1

Table 26 - 3 Correspondences between the SMSGn Registers, Memory Addresses, and the Internal Data Addresses for the Sequencer.

SMSGn	Addresses	Internal Data Addresses for the Sequencer
SMSG15	F03DEH, F03DFH	1111B
SMSG14	F03DCH, F03DDH	1110B
SMSG13	F03DAH, F03DBH	1101B
SMSG12	F03D8H, F03D9H	1100B
SMSG11	F03D6H, F03D7H	1011B
SMSG10	F03D4H, F03D5H	1010B
SMSG9	F03D2H, F03D3H	1001B
SMSG8	F03D0H, F03D1H	1000B
SMSG7	F03CEH, F03CFH	0111B
SMSG6	F03CCH, F03CDH	0110B
SMSG5	F03CAH, F03CBH	0101B
SMSG4	F03C8H, F03C9H	0100B
SMSG3	F03C6H, F03C7H	0011B
SMSG2	F03C4H, F03C5H	0010B
SMSG1	F03C2H, F03C3H	0001B
SMSG0	F03C0H, F03C1H	0000B

26.3.5 Sequencer control register (SMSC)

The SMSC register controls operation of the SNOOZE mode sequencer. Specifically, this register is used for control over starting and stopping of operation of the sequencer and holding the activating trigger for the sequencer pending, and for setting the source clock for counting in waiting triggered by the wait commands and the trigger for starting operation of the sequencer.

The SMSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 26 - 6 Format of Sequencer Control Register (SMSC) (1/2)

Address: F03E0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
SMSC	SMSSTART	SMSSTOP	SMSTRGWAIT	LONGWAIT	SMSTRGSEL3	SMSTRGSEL2	SMSTRGSEL1	SMSTRGSEL0
SMSSTART Note 1	Control over operation of the SNOOZE mode sequencer							
0	The sequencer is stopped. Condition for setting to 0: Setting the SMSSTOP bit to 1							
1	The sequencer is waiting for the selected activating trigger or operating. Condition for setting to 1: Writing 1 to this bit							
SMSSTOP Notes 1, 2	Control over operation of the SNOOZE mode sequencer							
0	—							
1	Trigger for forcibly terminating processing by the sequencer Terminates processing by the sequencer and forcibly places the sequencer in the stopped state. Specifically, the sequencer is stopped on completion of the processing being handled at the time the SMSSTOP bit is set to 1.							
SMSTRGWAIT AITNote 3	Control over holding the activating trigger pending							
0	Disables holding the activating trigger pending							
1	Enables holding the activating trigger pending On reception of the selected activating trigger, the trigger is held pending. Clearing the SMSTRGWAIT bit to 0 enables the trigger being held pending, so the sequencer starts operating.							

Note 1. Writing 0 to the SMSSTART and SMSSTOP bits has no effect.

Note 2. The SMSSTOP bit is always read as 0.

Note 3. Only set the SMSTRGWAIT bit to 1 while the value of the SMSSTART bit is 0, that is, while the sequencer is stopped.

Figure 26 - 6 Format of Sequencer Control Register (SMSC) (2/2)

LONGWAIT	Status flag indicating the source clock for counting for use with the wait command
0	Stops supply of the low-speed on-chip oscillator clock.
1	Enables supply of the low-speed on-chip oscillator clock.

SMSTRGS EL3	SMSTRGS EL2	SMSTRGS EL1	SMSTRGS EL0	Selected Activating Trigger for the SNOOZE Mode Sequencer ^{Note 1}
0	0	0	0	Interval detection interrupt (INTITL) from the 32-bit interval timer
0	0	0	1	Pin input edge detection interrupt (INTP3)
0	0	1	0	UART0 reception transfer end interrupt (INTSR0)
0	0	1	1	CSI00 transfer end interrupt (INTCSI00)
0	1	0	0	A/D conversion end interrupt (INTAD)
0	1	0	1	ELC output signal
0	1	1	0	UARTA0 reception transfer end interrupt (INTUR0) ^{Note 2}
0	1	1	1	Timer channel 02 count or capture end interrupt (INTTM02)
1	0	0	0	IICA0 communication end interrupt (INTIICA0) ^{Note 3}
1	0	0	1	Voltage detection interrupt (INTLVI)
1	0	1	0	Key return signal detection interrupt (INTKR) ^{Note 4}

Note 1. Only set the SMSTRGSEL3 to SMSTRGSEL0 bits to 1 while the value of the SMSSTART bit is 0, that is, while the sequencer is stopped.

Note 2. This is only selectable in products with 36 to 48 pins.

Note 3. This is only selectable in products with 24 to 48 pins.

Note 4. This is only selectable in products with 40 to 48 pins.

26.3.6 Sequencer status register (SMSS)

The SMSS register indicates the state of the sequencer. This register can be used to confirm whether the sequencer is operating or stopped, the values of the sequencer's zero and carry flags, and the number of the SMSIp register that holds the command currently being executed by the sequencer.

The SMSS register can be read by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 26 - 7 Format of Sequencer Status Register (SMSS)

Address: F03E1H
After reset: 00H
R/W: R

Symbol	<7>	6	5	4	3	2	1	0
SMSS	SMSSTAT	SZ	SCY	SMSCV4	SMSCV3	SMSCV2	SMSCV1	SMSCV0

SMSSTAT	State of operation of the SNOOZE mode sequencer						
0	The SNOOZE mode sequencer is stopped.						
1	The SNOOZE mode sequencer is operating.						

SZ	Zero flag of the SNOOZE mode sequencer						
0	The result of the most recent operation by the sequencer was not 0.						
1	The result of the most recent operation by the sequencer was 0.						

SCY	Carry flag of the SNOOZE mode sequencer						
0	The operation most recent operation by the sequencer did not produce carrying or borrowing.						
1	The operation most recent operation by the sequencer produced carrying or borrowing.						

SMSCV4	SMSCV3	SMSCV2	SMSCV1	SMSCV0	State of processing by the SNOOZE mode sequencer		
0 to 31					These bits indicate the number of the SMSIp register from which the sequencer is reading the command to be handled or has read the command that is currently being processed. Example: When the value of the SMSCV[4:0] bits is 3, the processing set up in the SMSI3 register is currently being read or its execution is in progress.		

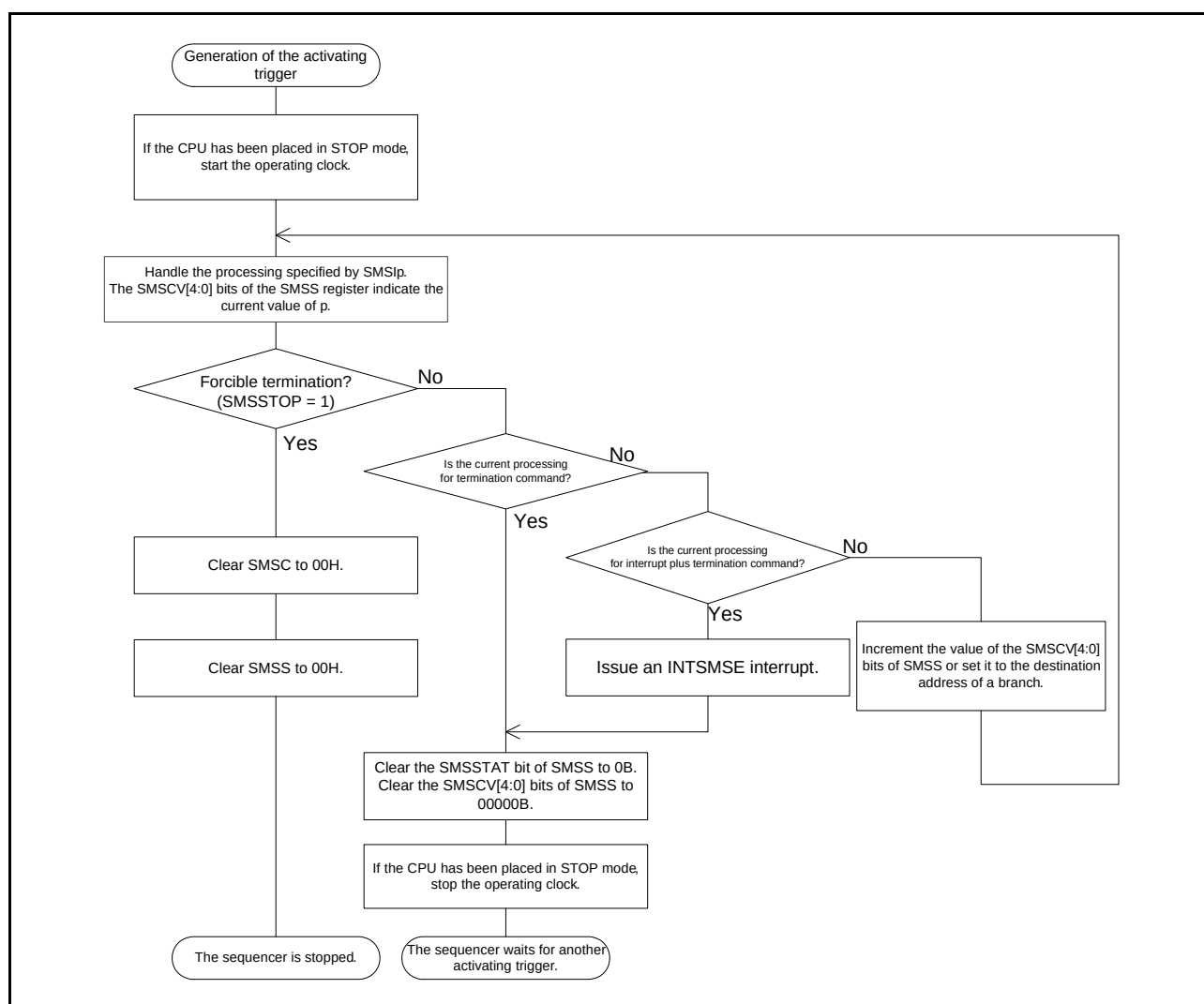
26.4 Operations of the SNOOZE Mode Sequencer

26.4.1 Internal operations of the SNOOZE mode sequencer

Sequencing by the SNOOZE mode sequencer starts in response to the activating trigger specified by the SMSTRGSEL0 to SMSTRGSEL4 bits of the SMSC register. Following activation, the sequencer handles the processing specified by the SMSIp register, and then handles the processing specified by the SMSIp register indicated by the SMSCV0 to SMSCV4 bits of the SMSS register. After execution of the processing for termination command or interrupt plus termination command, the sequencer has finished one round of processing and waits for another activating trigger. Moreover, setting the SMSSTOP bit of the SMSC register to 1 to generate a trigger for forcible termination leads to the sequencer being stopped.

Figure 26 - 8 shows the flow of internal operations of the SNOOZE mode sequencer.

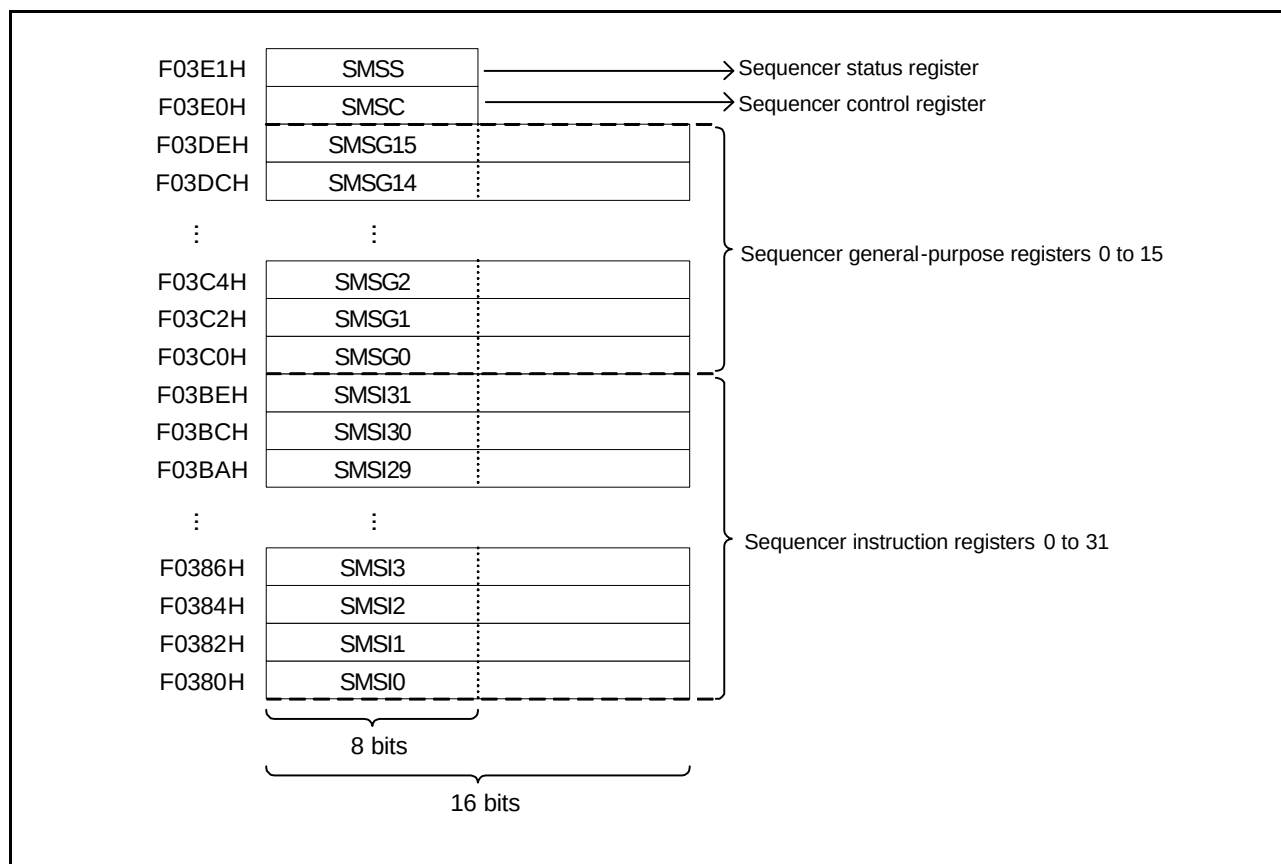
Figure 26 - 8 Flow of Internal Operations of the SNOOZE Mode Sequencer



26.4.2 Memory space allocated to the sequencer

The registers of the SNOOZE mode sequencer are in the second SFR space as shown in **Figure 26 - 9**. The sequencer handles the 16-bit processing codes stored at even addresses in the range from F0380H to F03BFH as being at five-bit addresses from 0 to 31 (00000B to 11111B). The sequencer also has direct access to the 16-bit data for processing stored at even addresses in the range from F03C0H to F03DFH as being at four-bit addresses from 0 to 15 (0000B to 1111B).

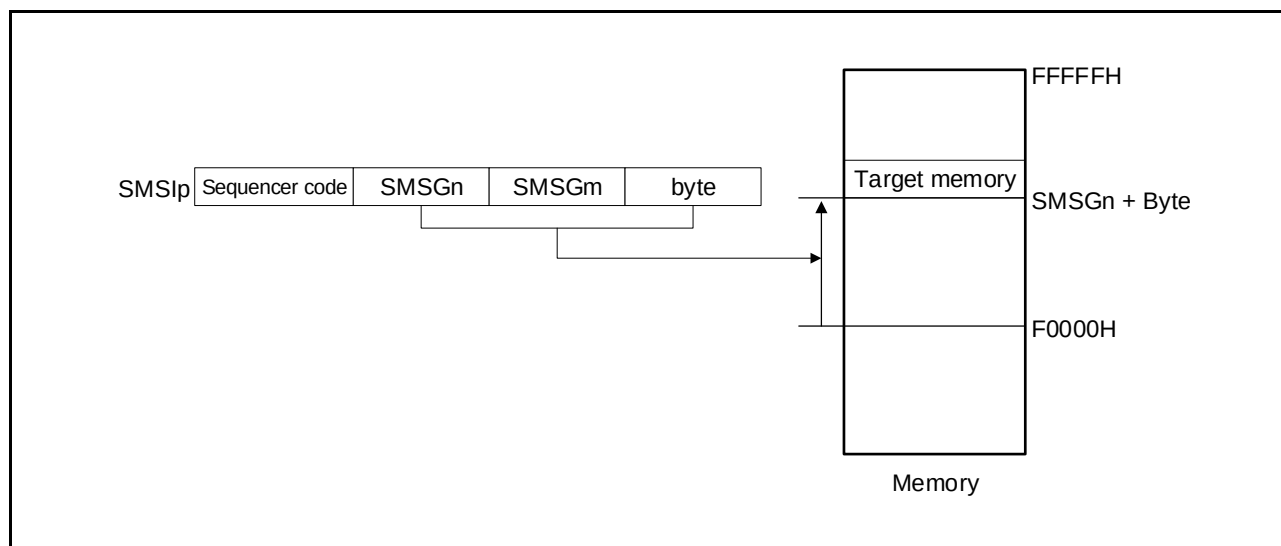
Figure 26 - 9 Memory Space Allocated to the SNOOZE Mode Sequencer



Moreover, the sequencer also has indirect access with some limitations^{Note} to the memory space from F0000H to FFFFFH through use of the SMSGn registers.

- Note** Access by the SNOOZE mode sequencer to the areas listed below is prohibited.
- General-purpose register area
 - Mirror area
 - Data flash memory area
 - Area to which the monitor program for debugging is allocated (when on-chip debugging is in use)
→ For details, see **31.4 Allocation of Memory Spaces to User Resources**.
 - Sequencer instruction registers 0 to 31 (SMSI0 to SMSI31)
 - Sequencer control register (SMSC)
 - Sequencer status register (SMSS)
 - SMSSEN bit of peripheral enable register 1 (PER1)
 - SMSRES bit of peripheral reset control register 1 (PRR1)

Figure 26 - 10 Memory Space to which the Sequencer Has Access



26.4.3 Sequencer flags

The sequencer has flags that are set or reset in response to the results of operations.

(a) Sequencer zero flag (SZ)

The SZ flag is an internal flag of the sequencer. The flag is set to 1 when the result of addition, subtraction, or comparison is 0. Otherwise, the flag is cleared to 0. The flag is only for use in the internal processing by the sequencer.

For details, see **26.5 Commands for Use in Processing by the Sequencer**.

(b) Sequencer carry flag (SCY)

The SCY flag reflects the state of addition or subtraction producing an overflow or underflow, the value of the shifted-out bit in logical shifting processing, or the result of 1-bit data transfer. The flag is only for use in the internal processing by the sequencer.

For details, see **26.5 Commands for Use in Processing by the Sequencer**.

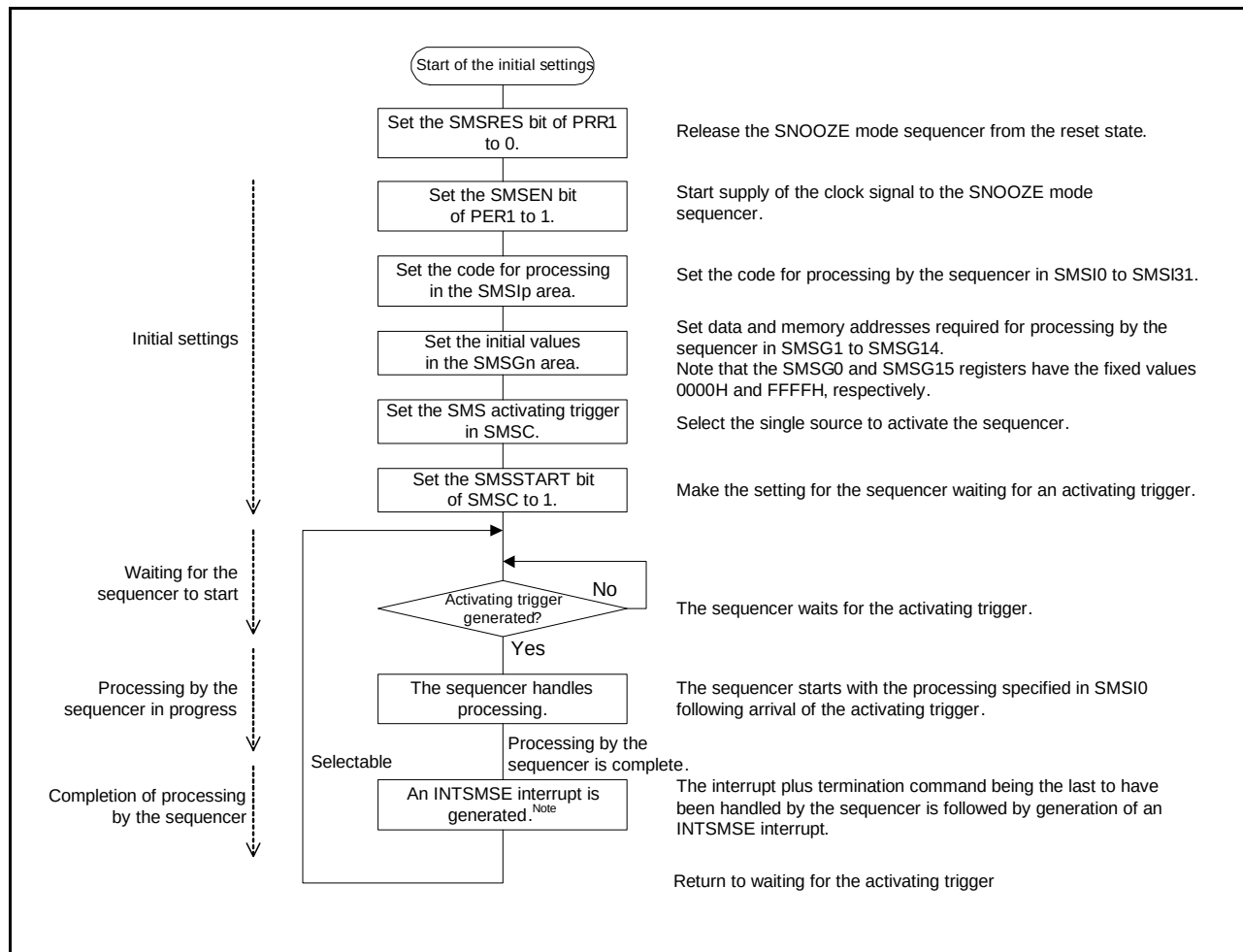
The values of the SZ and SCY flags of the sequencer can be read from the corresponding bits of the SMSS register. See **26.3.6 Sequencer status register (SMSS)**.

26.4.4 Procedures for running the SNOOZE mode sequencer

- (1) Example of the initial settings, activation, and operation of the SNOOZE mode sequencer

Figure 26 - 11 shows an example of a flow of processing from the initial settings until activation and the completion of processing by the SNOOZE mode sequencer.

Figure 26 - 11 Flow of Activating and Running the SNOOZE Mode Sequencer

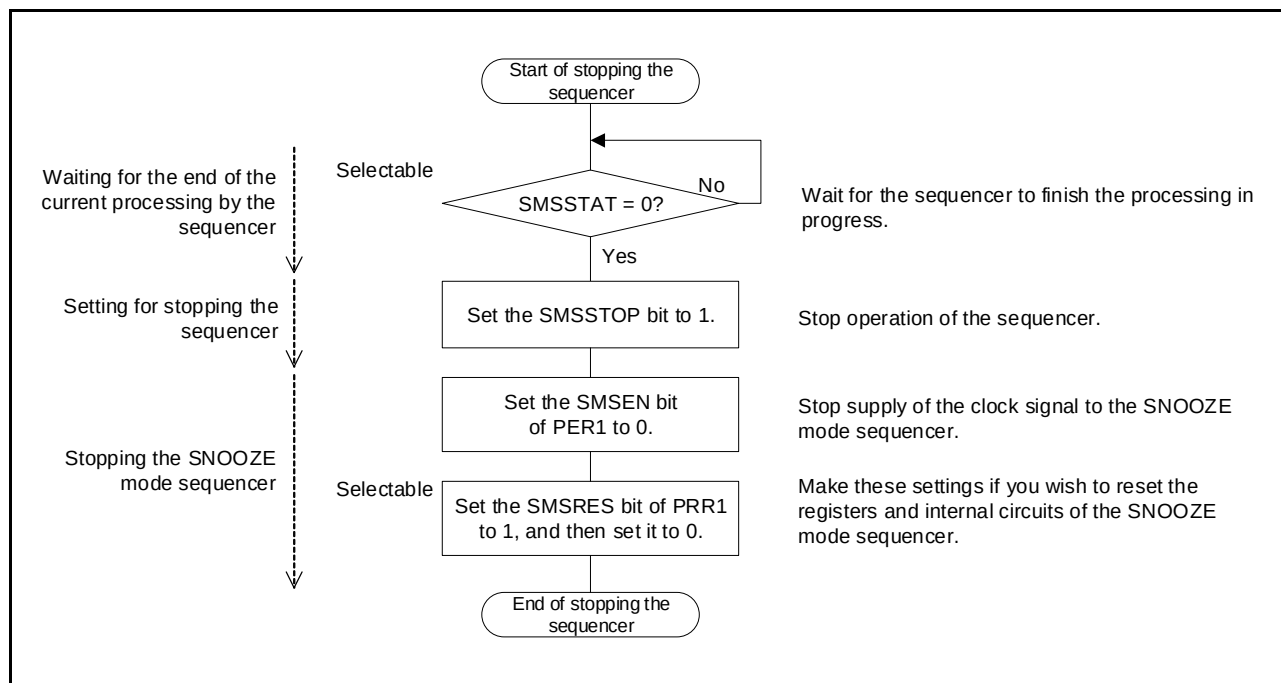


Note If processing by the sequencer ends following the processing for termination command or setting of the trigger bit for forcible termination (the SMSSTOP bit in the SMSC register), an INTSMSE interrupt will not be generated. If processing by the sequencer ends for the latter reason (setting of the trigger bit for forcible termination), the SMSC register itself will be initialized. Therefore, to restart processing by the sequencer, make the initial settings of the SMSC register again (the SMSIp and SMSGn registers are not reset).

(2) Example of a flow of stopping the SNOOZE mode sequencer

Figure 26 - 12 shows an example of a flow of stopping the SNOOZE mode sequencer while it is operating.

Figure 26 - 12 Flow of Stopping the SNOOZE Mode Sequencer



26.4.5 States of the SNOOZE mode sequencer

The SNOOZE mode sequencer has four states: the sequencer stopped state, the activating trigger waiting state, the activation pending state, and the sequencer operating state. Transitions between these states proceed according to settings of the registers and input of the activating trigger.

- Sequencer stopped state

The sequencer enters the stopped state immediately after it is released from the reset state or its operation is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1. Setting the SMSSTART bit of the SMSC register to 1 places the sequencer in the activating trigger waiting state.

- Activating trigger waiting state

The activating trigger waiting state is that in which the detection of the activating trigger for the sequencer is enabled by setting the SMSSTART bit of the SMSC register to 1 but the trigger has not yet arrived. Detection of the activating trigger specified by the SMSTRGSEL3 to SMSTRGSEL0 bits of the SMSC register in this state starts the sequencer operating by placing it in the operating state. Note that if the setting of the SMSTRGWAIT bit of the SMSC register is also 1, the sequencer will not start operating and enters the activation pending state. In addition, if operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

- Activation pending state

The activation pending state is entered following detection of the activating trigger while the value of the SMSTRGWAIT bit of the SMSC register is 1. Clearing the SMSTRGWAIT bit to 0 in this state starts the sequencer operating by placing the sequencer in the sequencer operating state. If operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

- Sequencer operating state

The sequencer operating state is that in which the sequencer is operating and is handling processing specified by the SMSIp registers.

Executing the termination or interrupt plus termination command places the sequencer in the activating trigger waiting state. If operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

The SMSSTART and SMSTRGWAIT bits of the SMSC register and the SMSSTAT bit of the SMSS register can be used to confirm the current state of the SNOOZE mode sequencer.

Table 26 - 4 Correspondence between the Bit Settings and States of the SNOOZE Mode Sequencer

SMSTRGWAIT	SMSSTAT	SMSSTART	State of the SNOOZE Mode Sequencer
0	0	0	Sequencer stopped state
0	0	1	Activating trigger waiting state
0	1	1	Sequencer operating state
1	0	0	Sequencer stopped state
1	0	1	Activating trigger waiting state
1	1	1	Setting prohibited (prohibited state) ^{Note}

Note Enabling holding a trigger pending while the sequencer is in the operating state is prohibited.

26.5 Commands for Use in Processing by the Sequencer

The sequencer can handle 21 types of processing. For details, see sections **26.5.1** to **26.5.21**.

Terms used in the descriptions of the commands (**26.5.1** to **26.5.21**) are defined below.

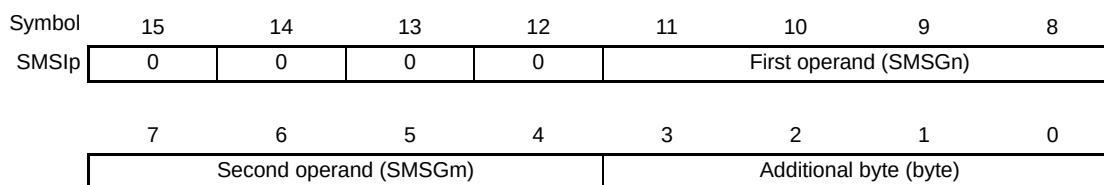
- Equivalent CPU command: Assembler language instruction that leads to the same processing as the given command.
- Equivalent CPU operation: Processing expressed in symbolic form
- \$addr5: 5-bit relative address in 5 of 8 bits (-31 to 0 to 31)
- jdisp8: 8-bit signed displacement value (0000 0000B to 0001 1111B, 1111 1111B to 1110 0001B)
- Number of clock cycles for processing: Number of cycles of the fCLK clock from the time the sequencer starts the processing until the result of the processing is actually reflected
- SMSIp: Sequencer instruction registers (p = 0 to 31)
- SMSGn: Sequencer internal data address (n = 0 to 15) for sequencer general-purpose registers, which is stored as the first operand of most commands
- SMSGm: Sequencer internal data address (m = 0 to 15) for sequencer general-purpose registers, which is stored as the second operand of most commands
- Example of a statement: Example of a binary value to be entered in the SMSIp register
- Sequencer code: Code that is to be entered in bits 15 to 12 of the SMSIp register and indicates the type of processing
- First operand: Code to be entered in bits 11 to 8 of the SMSIp register
- Second operand: Code to be entered in bits 7 to 4 of the SMSIp register
- Additional byte: Code to be entered in bits 3 to 0 of the SMSIp register

26.5.1 8-bit data transfer 1

8-bit data transfer 1 is for transferring 8 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the destination for transfer in memory. The sequencer transfers the 8 bits of data specified as the second operand in the SMSGm register to the above destination in memory. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte.

To specify an SFR as the destination for transfer, set the address of the SFR that can be manipulated in 8-bit units in the first operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU. Note that the sequencer general-purpose registers n (n = 0 to 15) can be specified as destinations for transfer. Executing 8-bit data transfer 1 results in a value being stored in the 8 lower-order bits of the specified sequencer general-purpose register n (n = 0 to 15).

Sequencer code: 0000B
Number of clock cycles for processing: 3 cycles of fCLK
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOV [SMSGn + Byte], SMSGm
Equivalent CPU operation: [SMSGn + Byte] ← SMSGm



Example of a statement: 0000 0001 0111 0010 B

The equivalent CPU command in this case is MOV [SMSG1+2H], SMSG7.

In the case where SMSG1 = FE00H and SMSG7 = xx12H, the value 12H stored in the SMSG7 register will be stored at the address FFE02H.

26.5.2 8-bit data transfer 2

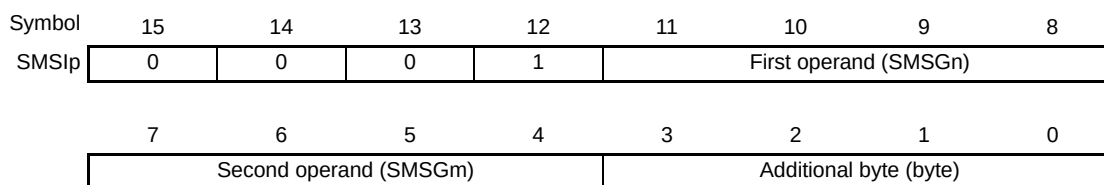
8-bit data transfer 2 is for transferring 8 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The sequencer stores the 8 bits of data at the source address for transfer in the SMSGm register specified as the second operand.

Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte.

To specify an SFR as the source for transfer, set the address of the SFR that can be manipulated in 8-bit units in the second operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU. Note that the sequencer general-purpose registers n (n = 0 to 15) can be specified as sources for transfer.

Executing 8-bit data transfer 2 results in reading of the value of the 8 lower-order bits of the specified sequencer general-purpose register n (n = 0 to 15).

Sequencer code: 0001B
 Number of clock cycles for processing: 3 cycles of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: MOV SMSGm, [SMSGn + Byte]
 Equivalent CPU operation: SMSGm ← [SMSGn + Byte]



Example of a statement: 0001 0001 0111 0010 B

The equivalent CPU command in this case is MOV SMSG7, [SMSG1+2H].

In the case where SMSG1 = FE00H and FFE02H = xx12H, the value 12H stored at the address FFE02H will be stored in the SMSG7 register.

26.5.3 16-bit data transfer 1

16-bit data transfer 1 is for transferring 16 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the destination for transfer in memory. The sequencer transfers the 16 bits of data specified as the second operand in the SMSGm register to the above destination in memory. Specify an even address as the destination for transfer in memory. If an odd address is specified, the LSB of the address is automatically changed to 0 before the data are transferred.

Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte.

To specify an SFR as the destination for transfer, set the address of the SFR that can be manipulated in 16-bit units in the first operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0010B
Number of clock cycles for processing: 3 cycles of fCLK
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOVW [SMSGn + Byte], SMSGm
Equivalent CPU operation: [SMSGn + Byte] ← SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	0	1	0	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0010 0001 0111 0010 B

The equivalent CPU command in this case is MOVW [SMSG1+2H], SMSG7.

In the case where SMSG1 = FE00H and SMSG7 = 1234H, the value 1234H stored in the SMSG7 register will be stored at the addresses FFE02H and FFE03H.

26.5.4 16-bit data transfer 2

16-bit data transfer 2 is for transferring 16 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The sequencer stores the 16 bits of data at the source address for transfer in the SMSGm register specified as the second operand. Specify an even address as the source for transfer in memory. If an odd address is specified, the LSB of the address is automatically changed to 0 before the data are transferred.

Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte.

To specify an SFR as the source for transfer, set the address of the SFR that can be manipulated in 16-bit units in the second operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0011B
 Number of clock cycles for processing: 3 cycles of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: MOV SMSGm, [SMSGn + Byte]
 Equivalent CPU operation: SMSGm ← [SMSGn + Byte]

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	0	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0011 0001 0111 0010 B

The equivalent CPU command in this case is MOVW SMSG7, [SMSG1+2H].

In the case where SMSG1 = FE00H, and FFE02H and FFE03H = 1234H, the value 1234H stored at the addresses FFE02H and FFE03H will be stored in the SMSG7 register.

26.5.5 1-bit data setting

1-bit data setting is for setting a specified bit among eight bits of data at a specified address to 1. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the number of the bit to be set to 1.

Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte.

When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0100B
Number of clock cycles for processing: 4 cycles of fCLK
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: SET1 [SMSGn + Byte].bit
Equivalent CPU operation: [SMSGn + Byte].bit ← 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	0	0	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (bit number)				Additional byte (byte)			

Example of a statement: 0100 0001 0011 0010 B

The equivalent CPU command in this case is SET1 [SMSG1+2H].3.

In the case where SMSG1 = FE00H and FFE02H = 0000 0000B, bit 3 at the address FFE02H will be set to 1, making the value at the address FFE02H 0000 1000B.

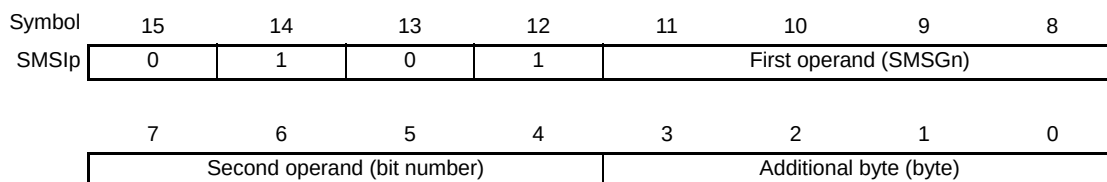
26.5.6 1-bit data clearing

1-bit data clearing is for clearing a specified bit among eight bits of data at a specified address to 0. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the number of the bit to be cleared to 0.

Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte.

When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0101B
Number of clock cycles for processing: 4 cycles of fCLK
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: CLR1 [SMSGn + Byte].bit
Equivalent CPU operation: [SMSGn + Byte].bit ← 0



Example of a statement: 0101 0001 0011 0010 B

The equivalent CPU command in this case is CLR1 [SMSG1+2H].3.

In the case where SMSG1 = FE00H and FFE02H = 0000 1000B, bit 3 at the address FFE02H will be cleared to 0, making the value at the address FFE02H 0000 0000B.

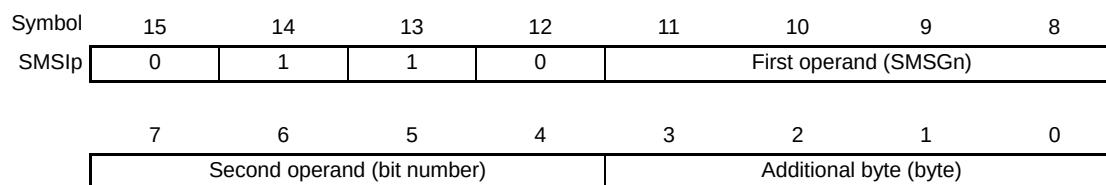
26.5.7 1-bit data transfer

1-bit data transfer is for transferring the value of a specified bit among eight bits of data at a specified address to the SCY flag. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The second operand is used to specify the bit whose value is to be transferred.

Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte.

When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0110B
 Number of clock cycles for processing: 3 cycles of fCLK
 Flags: The state of the SZ flag is retained. The SCY flag reflects the result of the transfer.
 Equivalent CPU command: MOV1 SCY, [SMSGn + Byte].bit
 Equivalent CPU operation: $SCY \leftarrow [SMSGn + Byte].bit$



Example of a statement: 0110 0001 0011 0010 B

The equivalent CPU command in this case is MOV1 SCY, [SMSG1+2H].3.

In the case where SMSG1 = FE00H and FFE02H = 0000 1000B, the value of bit 3 at the address FFE02H will be transferred to the SCY flag, so the SCY flag will be set to 1.

26.5.8 Word addition

Word addition is for adding two 16-bit values. The values of the SMSGn and SMSGm registers that are specified as the first and second operands are added, and the result is stored in the SMSGn register specified as the first operand. If the result of addition that is stored in the SMSGn register is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic carry has been generated from bit 15, the SCY flag is set to 1. Otherwise, the flag is cleared to 0. Set the additional byte to 0000B.

Sequencer code: 0111B (additional byte: 0000B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The SZ and SCY flags reflect the result of the calculation.
 Equivalent CPU command: ADDW SMSGn, SMSGm
 Equivalent CPU operation: SMSGn, SCY ← SMSGn + SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (SMSGm)				0	0	0	0

Example of a statement: 0111 0001 0011 0000 B

The equivalent CPU command in this case is ADDW SMSG1, SMSG3.

In the case where SMSG1 = FFF0H and SMSG3 = 0010H, the settings of the register and flags as the results of the addition will be as follows: SMSG1 = 0000H, SCY = 1, and SZ = 1.

26.5.9 Word subtraction

Word subtraction is for subtracting two 16-bit values. The value of the SMSG_m register specified as the second operand is subtracted from the value of the SMSG_n register specified as the first operand, and then the resulting value is stored in the SMSG_n register specified in the first operand. If the result of subtraction that is stored in the SMSG_n register is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic borrow has been generated from bit 0, the SCY flag is set to 1. Otherwise, the flag is cleared to 0.

Set the additional byte to 0001B.

Sequencer code: 0111B (additional byte: 0001B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The SZ and SCY flags reflect the result of the calculation.
 Equivalent CPU command: SUBW SMSG_n, SMSG_m
 Equivalent CPU operation: SMSG_n, SCY ← SMSG_n - SMSG_m

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSG _n)			
	7	6	5	4	3	2	1	0
	Second operand (SMSG _m)				0	0	0	1

Example of a statement: 0111 0001 0011 0001 B

The equivalent CPU command in this case is SUBW SMSG1, SMSG3.

In the case where SMSG1 = 1234H and SMSG3 = 1200H, the settings of the register and flags as the results of the subtraction will be as follows: SMSG1 = 0034H, SCY = 0, and SZ = 0.

26.5.10 Word comparison

Word comparison is for comparing two 16-bit values. The value of the SMSGm register specified as the second operand is subtracted from the value of the SMSGn register specified as the first operand. If the setting of the SMSGn register as the result of the subtraction is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic borrow has been generated from bit 0, the SCY flag is set to 1. Otherwise, the flag is cleared to 0.

Set the additional byte to 0010B.

Sequencer code: 0111B (additional byte: 0010B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The SZ and SCY flags reflect the result of the calculation.
 Equivalent CPU command: CMPW SMSGn, SMSGm
 Equivalent CPU operation: SMSGn - SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (SMSGm)				0	0	1	0

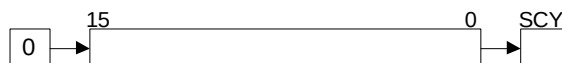
Example of a statement: 0111 0001 0011 0010 B

The equivalent CPU command in this case is CMPW SMSG1, SMSG3.

In the case where SMSG1 = 1234H and SMSG3 = 1200H, the settings of the flags as the results of the comparison will be as follows: SCY = 0 and SZ = 0.

26.5.11 Logical shift right

Logical shift right is for shifting a 16-bit value by 1 bit to the right. The value of the SMSGn register specified as the first operand is shifted by 1 bit to the right. At this time, the MSB (bit 15) of the SMSGn register is set to 0 and the value of the LSB (bit 0) that is shifted out is stored in the SCY flag.



Set the additional byte to 0011B. Since the second operand is not used, set it to 0000B.

Sequencer code: 0111B (additional byte: 0011B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The state of the SZ flag is retained. The SCY flag reflects the result of the calculation.
 Equivalent CPU command: SHRW SMSGn
 Equivalent CPU operation: $SCY \leftarrow SMSGn.0$, $SMSGn.m-1 \leftarrow SMSGn.m$, $SMSGn.15 \leftarrow 0$

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1	1

Example of a statement: 0111 0001 0000 0011 B

The equivalent CPU command in this case is SHRW SMSG1.

In the case where SMSG1 = AAF5H, the settings of the register and flag as the results of the logical shift will be as follows: SMSG1 = 557AH and SCY = 1.

26.5.12 Branch 1 (SCY = 1)

Branch 1 is for processing to branch if SCY = 1. With SCY = 1, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SCY = 0, processing proceeds to that set in the next SMSIp register.

Set the additional byte to 0000B. Set values in the range from -31 to 31 as the first and second operands.

Sequencer code: 1000B (additional byte: 0000B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: BC \$addr5
 Equivalent CPU operation: SMSCV[4:0] ← SMSCV[4:0] + jdisp8 if SCY = 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands (\$addr5)			
	7	6	5	4	3	2	1	0
	First and second operands (\$addr5)				0	0	0	0

Example of a statement: 1000 0000 0101 0000 B

The equivalent CPU command in this case is BC \$05H.

In the case where the value of the SMSCV[4:0] bits of the SMSS register is 03H, and SCY = 1, the value of the SMSCV[4:0] bits will be 08H, so processing will proceed to that stored in the SMSI8 register.

26.5.13 Branch 2 (SCY = 0)

Branch 2 is for processing to branch if SCY = 0. With SCY = 0, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SCY = 1, processing proceeds to that set in the next SMSIp register.

Set the additional byte to 0001B. Set values in the range from -31 to 31 as the first and second operands.

Sequencer code: 1000B (additional byte: 0001B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: BNC \$addr5
 Equivalent CPU operation: SMSCV[4:0] ← SMSCV[4:0] + jdisp8 if SCY = 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands (\$addr5)			
	7	6	5	4	3	2	1	0
	First and second operands (\$addr5)				0	0	0	1

Example of a statement: 1000 1111 1110 0001 B

The equivalent CPU command in this case is BNC \$0FEH

In the case where the value of the SMSCV[4:0] bits of the SMSS register is 0EH, and SCY = 0, the value of the SMSCV[4:0] bits will be 0CH, so processing will proceed to that stored in the SMSI12 register.

26.5.14 Branch 3 (SZ = 1)

Branch 3 is for processing to branch if SZ = 1. With SZ = 1, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SZ = 0, processing proceeds to that set in the next SMSIp register.

Set the additional byte to 0010B. Set values in the range from -31 to 31 as the first and second operands.

Sequencer code: 1000B (additional byte: 0010B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: BZ \$addr5
 Equivalent CPU operation: SMSCV[4:0] ← SMSCV[4:0] + jdisp8 if SZ = 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands (\$addr5)			
	7	6	5	4	3	2	1	0
	First and second operands (\$addr5)				0	0	1	0

Example of a statement: 1000 0000 0101 0010 B

The equivalent CPU command in this case is BZ \$05H.

In the case where the value of the SMSCV[4:0] bits of the SMSS register is 03H, and SZ = 1, the value of the SMSCV[4:0] bits will be 08H, so processing will proceed to that stored in the SMSI8 register.

26.5.15 Branch 4 (SZ = 0)

Branch 4 is for processing to branch if SZ = 0. With SZ = 0, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SZ = 1, processing proceeds to that set in the next SMSIp register.

Set the additional byte to 0011B. Set values in the range from -31 to 31 as the first and second operands.

Sequencer code: 1000B (additional byte: 0011B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: BNZ \$addr5
 Equivalent CPU operation: SMSCV[4:0] ← SMSCV[4:0] + jdisp8 if SZ = 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands (\$addr5)			
	7	6	5	4	3	2	1	0
	First and second operands (\$addr5)				0	0	1	1

Example of a statement: 1000 1111 1110 0011 B

The equivalent CPU command in this case is BNZ \$0FEH.

In the case where the value of the SMSCV[4:0] bits of the SMSS register is 0EH, and SZ = 0, the value of the SMSCV[4:0] bits will be 0CH, so processing will proceed to that stored in the SMSI12 register.

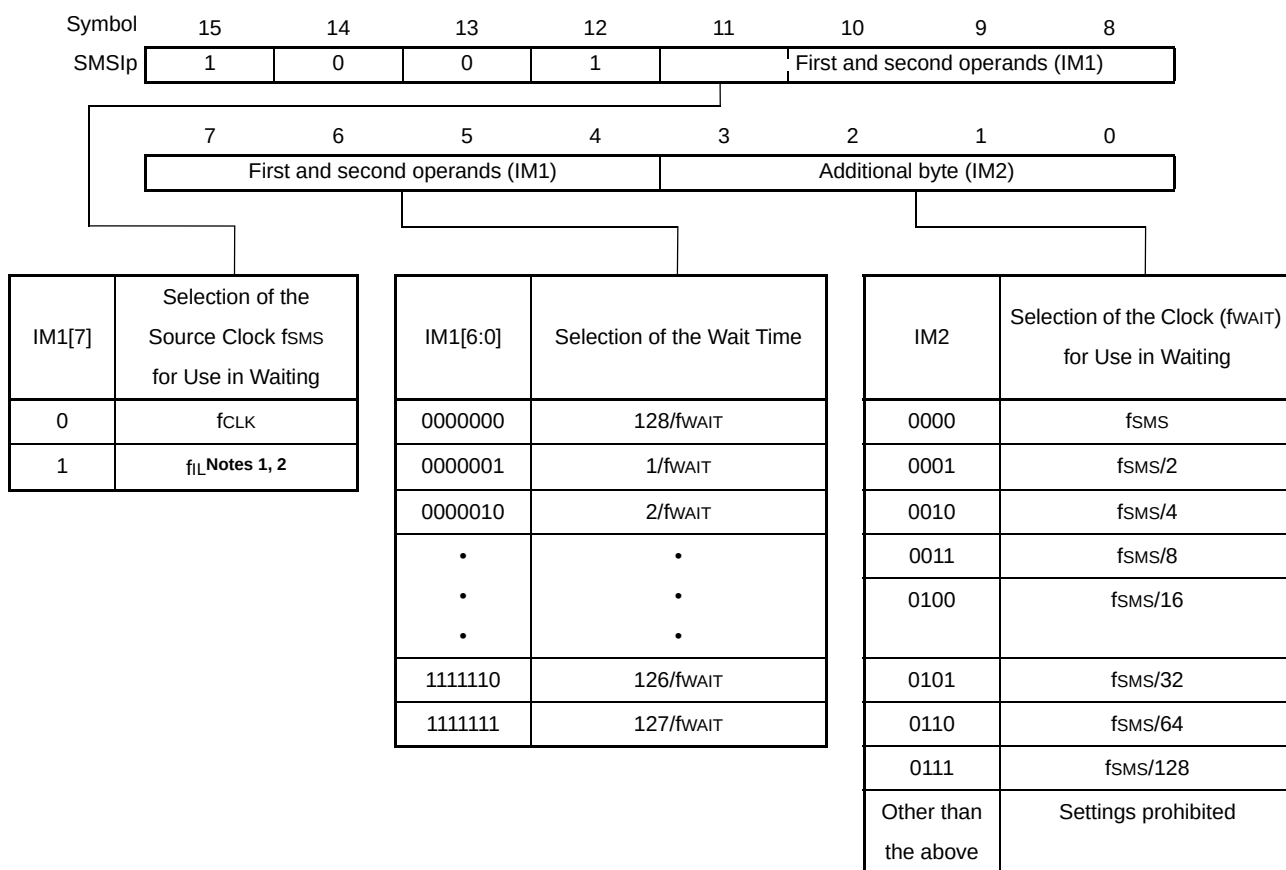
26.5.16 Wait

The wait command holds processing pending for a certain period. Execution of processing is held pending for the number of cycles of f_{WAIT} specified in the second operand by using the clock and division ratio specified as the first operand and additional byte.

Use the first and second operands to select the source clock for counting of the wait time and the value to be counted as the wait period, and use the additional byte to select the clock for use in waiting, specifically, select the divisor for the frequency of the selected source clock.

If the low-speed on-chip oscillator clock is to be used as the source clock for counting of the wait time, set the LONGWAIT bit of the SMSC register to 1 in advance.

Sequencer code: 1001B
Number of clock cycles for processing: See the description titled “**(1) Number of clock cycles for use in waiting**”.
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WAIT IM1, IM2
Equivalent CPU operation: Holding processing pending for a certain period



Example of a statement:

Example of the Setting of the SMSIp Register	Source Clock for Use in Waiting	Clock for Use in Waiting	Number to Count for Waiting	Wait Time in μ s (typ.)
1001 0 0000001 0000	fCLK	fCLK	1	0.03125
1001 0 0000000 0111	fCLK	fCLK/128	128	512
1001 0 1000000 0110	fCLK	fCLK/64	64	128
1001 1 0000001 0000	fiL	fiL	1	30.51757813
1001 1 0000000 0111	fiL	fiL/128	128	500000
1001 1 1100000 0101	fiL	fiL/32	96	93750

Note 1. When the low-speed on-chip oscillator clock (fiL) is in use for the CPU/peripheral hardware clock frequency (fCLK), select fCLK (IM1[7] = 0) as the source clock fsms for counting the wait time.

Note 2. When the low-speed on-chip oscillator clock (fiL) is to be used as the source clock for the timing of waiting by the SNOOZE mode sequencer, set peripheral enable register 0 (PER0) and peripheral enable register 1 (PER1) to 00H and 40H, respectively.

Remark fCLK: CPU/peripheral hardware clock frequency
fiL: Low-speed on-chip oscillator clock frequency (32.768 kHz)

(1) Number of clock cycles for use in waiting

The number of clock cycles for use in waiting depends on the setting of the wait command. Refer to values set for IM1 and IM2 to calculate the required processing time.

Note that the wait processing requires time for synchronization before the counter for use in timing the wait starts operating and after the counter finishes operating. Refer to an example described below to calculate the time for synchronization.

Before execution of the wait command:

(1 cycle of fCLK) + (2^{Note} cycles of the selected clock for use in waiting)

After execution of the wait command:

2^{Note} cycles of fCLK

Note This is the number of clock cycles for synchronization and will be no greater than 2.
This time may also be extremely close to 1 cycle with some types of timing.

26.5.17 Conditional wait 1 (bit = 1)

In conditional wait 1, processing is repeatedly held pending as long as the value of a specified bit among eight bits of data at a specified address is 1. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the target bit for the condition. The sequencer reads the specified bit. If the setting is 1, the sequencer does not handle the succeeding processing but goes into wait processing. The sequencer repeats this as long as the value of the specified bit is 1. When reading from the specified bit returns 0, the sequencer handles the succeeding processing.

Set a value in the range from 0 to 7 (from 0000B to 0111B) as the second operand.

When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 1010B
Number of clock cycles for processing: See the description titled “(1) Number of clock cycles required for conditional wait 1”.
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WHILE1 [SMSGn + Byte].bit
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] if [SMSGn + Byte].bit = 1

Symbol	15	14	13	12	11	10	9	8
SMSlp	1	0	1	0	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (bit number)				Additional byte (byte)			

Example of a statement: 1010 0011 0101 0010 B

The equivalent CPU command in this case is WHILE1 [SMSG3+2].5.

In the case where SMSG3 = 02E8H, the sequencer holds processing pending as long as the value of bit 5 at address F02EAH is 1. When the value of the bit becomes 0, the sequencer handles the succeeding processing.

(1) Number of clock cycles required for conditional wait 1

In conditional wait 1 processing, access to the specified bit is repeated until its value becomes 0.

Starting the processing requires one cycle of fCLK and each round of reference to the register bit and judgement requires two cycles of fCLK. For example, the time set up in the case where the fifth access to the bit for reference of the destination register returns 0 is calculated from the following formula.

$$fCLK \times 1 + (fCLK \times 2) \times 5 = fCLK \times 11$$

Note that if attempted access to memory by the SNOOZE mode sequencer is in contention with access by the CPU or DTC, the SNOOZE mode sequencer may be placed in the pending state. Such cases also increase the number of clock cycles.

26.5.18 Conditional wait 2 (bit = 0)

In conditional wait 2, processing is repeatedly held pending as long as the value of a specified bit among eight bits of data at a specified address is 0. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the target bit for the condition. The sequencer reads the specified bit. If the setting is 0, the sequencer does not handle the succeeding processing but goes into wait processing. The sequencer repeats this as long as the value of the specified bit is 0. When reading from the specified bit returns 1, the sequencer handles the succeeding processing.

Set a value in the range from 0 to 7 (from 0000B to 0111B) as the second operand.

When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 1011B
Number of clock cycles for processing: See the description titled “**(1) Number of clock cycles required for conditional wait 2**”.
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WHILE0 [SMSGn + Byte].bit
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] if [SMSGn + Byte].bit = 0

Symbol	15	14	13	12	11	10	9	8
SMSlp	1	0	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (bit number)				Additional byte (byte)			

Example of a statement: 1011 0011 0101 0010 B

The equivalent CPU command in this case is WHILE0 [SMSG3+2].5.

In the case where SMSG3 = FFE0H, the sequencer holds processing pending as long as the value of bit 5 at address FFFE2H is 0. When the value of the bit becomes 1, the sequencer handles the succeeding processing.

(1) Number of clock cycles required for conditional wait 2

Since the number of clock cycles of waiting set up by conditional wait 2 is counted in the same way as described for conditional wait 1, refer to **26.5.17 (1) Number of clock cycles required for conditional wait 1**.

26.5.19 Termination

The termination command stops the SNOOZE mode sequencer. Specifically, execution of the command stops the SNOOZE mode sequencer, clears the SMSSTAT and SMSCV[4:0] bits in the SMSS register to 0, and places the sequencer in the activating trigger waiting state.

Set the additional byte to 0000B. Set all bits of the first and second operands to 0.

Sequencer code: 1111B (additional byte: 0000B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: FINISH
 Equivalent CPU operation: SMSCV[4:0] ← 0, stopping the sequencer

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Example of a statement: 1111 0000 0000 0000 B

The equivalent CPU command in this case is FINISH.

The termination command stops the sequencer, clears the SMSSTAT and SMSCV[4:0] bits of the SMSS register to 0, and places the sequencer in the activating trigger waiting state.

26.5.20 Interrupt plus termination

The interrupt plus termination command issues an interrupt signal and then stops the SNOOZE mode sequencer. Issuing the interrupt signal enables starting the CPU when it has been placed on standby. Specifically, execution of the command issues the interrupt signal, stops the SNOOZE mode sequencer, clears the SMSSTAT and SMSCV[4:0] bits in the SMSS register to 0, and places the sequencer in the activating trigger waiting state.

Set the additional byte to 0001B. Set all bits of the first and second operands to 0.

Sequencer code: 1111B (additional byte: 0001B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: WAKEUP
 Equivalent CPU operation: SMSS[4:0] ← 0, stopping the sequencer after issuing an interrupt

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	1

Example of a statement: 1111 0000 0000 0001 B

The equivalent CPU command in this case is WAKEUP.

The interrupt plus termination command stops the sequencer after issuing an INTSMSE interrupt, clears the SMSSTAT and SMSCV[4:0] bits of the SMSS register to 0, and places the sequencer in the activating trigger waiting state.

26.5.21 DTC activation

The DTC activation command issues a DTC activating source signal so that DTC transfer proceeds. That is, execution of the command issues an activating source signal for the DTC. If the corresponding control settings for the DTC have been made in advance, the specified DTC transfer will proceed.

Set the additional byte to 0010B. Set all bits of the first and second operands to 0.

Sequencer code: 1111B (additional byte: 0010B)
 Number of clock cycles for processing: 1 cycle of fCLK
 Flags: The states of the SZ and SCY flags are retained.
 Equivalent CPU command: DTCTRG
 Equivalent CPU operation: Output of a DTC activating source signal

Symbol	15	14	13	12	11	10	9	8
SMSlp	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1	0

Example of a statement: 1111 0000 0000 0010 B

The equivalent CPU command in this case is DTCTRG.

The DTC activation command issues a DTC activating source signal. DTC transfer proceeds according to the DTC control settings.

To hold processing by the sequencer pending until completion of DTC transfer, use the wait command to hold processing pending until the DTC has been started, and then use conditional wait 1 to hold the processing pending until the corresponding bit of DTC activation enable register i (DTCENi) (i = 0 to 4) has been set to 0.

26.6 Operation in Standby Modes

State	Operation of the SNOOZE Mode Sequencer
HALT mode	Operation continues. Note 1
STOP mode	The activating trigger for the SNOOZE mode sequencer can be accepted. Note 3
SNOOZE mode	Operation continues. Notes 2, 4, 5, 6

Note 1. When the subsystem clock is selected as fCLK, operation is disabled if the RTCLPC bit of the OSMC register is 1.

Note 2. The SNOOZE mode can only be set when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected as fCLK.

Note 3. Detection of an SMS activating trigger in STOP mode places the chip in SNOOZE mode, making the SNOOZE mode sequencer capable of operation. The state of the chip returns to the STOP mode after the operations of the SMS are completed. Note that the sequencer does not have access to certain memory areas in SNOOZE mode. For details, see **26.4.2 Memory space allocated to the sequencer.**

Note 4. When a transfer end interrupt from the CSIp in SNOOZE mode is being used as the activating trigger for the SNOOZE mode sequencer, use the interrupt plus termination command to release the chip from the SNOOZE mode and start processing by the CPU, or make the settings for reception by the CSIp (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting the SSCm register, and writing 1 to the SSm0 bit) again before the processing for termination.

Note 5. When a transfer end interrupt from the UARTq in SNOOZE mode is being used as the activating trigger for the SNOOZE mode sequencer, use the interrupt plus termination command to release the chip from the SNOOZE mode and start processing by the CPU, or make the settings for reception by the UARTq (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting the SSCm register, and writing 1 to the SSm1 bit) again before the processing for termination.

Note 6. When an A/D conversion end interrupt from the A/D converter in SNOOZE mode is being used as the activating trigger for the SNOOZE mode sequencer, use the interrupt plus termination command to release the chip from the SNOOZE mode and start processing by the CPU, or make the settings for the SNOOZE mode function of the A/D converter (writing 1 to the AWC bit after having written 0 to it) again before the processing for termination.

Remark p = 00; m = 0

Section 27 Capacitive Sensing Unit (CTS2La)

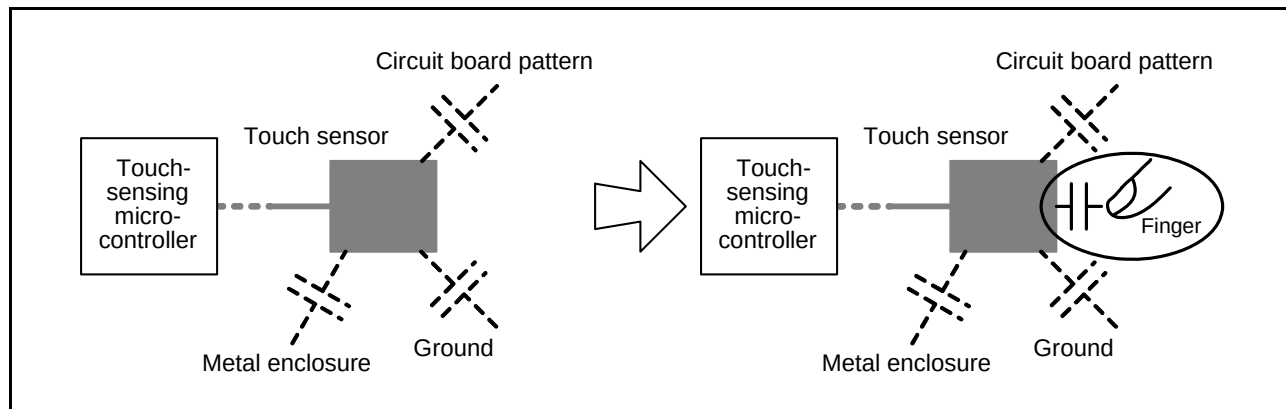
The number of output channels of the capacitive sensing unit depends on the product.

ROM size	32 or 64 Kbytes									
Pin count	16	20	24	25	30	32	36	40	44	48
Number of the CTS2La output channels	5 TS11 to TS13, TS18, TS20	9 TS10 to TS13, TS17, TS18, TS20, TS26, TS27	11 TS00, TS01, TS10 to TS13, TS17, TS18, TS20, TS26, TS27	12 TS00, TS01, TS10 to TS13, TS17 to TS20, TS26, TS27	16 TS00, TS01, TS10 to TS18, TS20, TS21, TS26 to TS28	17 TS00 to TS02, TS10 to TS18, TS20, TS21, TS26 to TS28	21 TS00 to TS04, TS10 to TS18, TS20 to TS23, TS26 to TS28	23 TS00 to TS05, TS10 to TS18, TS20 to TS24, TS26 to TS28	25 TS00 to TS05, TS09 to TS18, TS20 to TS28	29 TS00 to TS28

The capacitive sensing unit (CTS2La) measures the electrostatic capacitance of the capacitive sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTS2La to detect whether a finger is in contact with the capacitive sensor. The electrode surface of the capacitive sensor is usually enclosed with an electrical conductor so that a finger does not come into direct contact with the electrode.

As shown in **Figure 27 - 1**, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the electrostatic capacitance value increases.

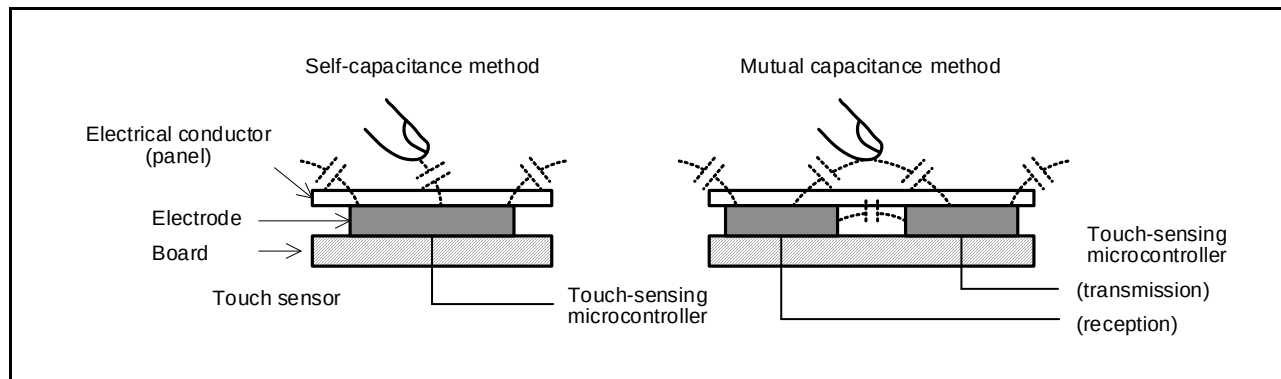
Figure 27 - 1 Increased Electrostatic Capacitance Because of the Presence of a Finger



Electrostatic capacitance is detected by the self-capacitance and mutual capacitance methods.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used; one electrode is used as a transmit electrode and the other electrode is used as a receive electrode. The CTSU detects a change in the electrostatic capacitance generated between these electrodes when a finger is placed close to them.

Figure 27 - 2 Self-capacitance Method and Mutual Capacitance Method



Electrostatic capacitance is measured by counting clock signal cycles whose frequency changes according to the amount of charged or discharged current for a specified period.

For details of the principles of measurement operation, see the RL78 family application note, Capacitive Touch Sensing Unit (CTS2L) Operation Explanation (R01AN5744). For details on the development of touch applications, see Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512).

27.1 Overview

Table 27 - 1 lists the CTSU functions and **Figure 27 - 3** shows a block diagram of the CTSU. **Figure 27 - 4** shows the sensor drive pulse output clock configuration.

Table 27 - 1 CTSU Functions

Item		Configuration
CTS2 operating voltage condition		VDD = 1.8 to 5.5 V
Operating clock		fCLK, fCLK/2, fCLK/4, or fCLK/8
Pins	Electrostatic capacitance measurement	TSm (m = 00 to 28) up to 29 channels
	Connection pin to capacitor for measurement secondary power	TSCAP (10 nF) We recommend connecting a 10 nF capacitor.
Measurement mode	Self-capacitance measurement mode	Electrostatic capacitance is measured from the charged current that flows toward the electrode used in the self-capacitance method.
	Mutual capacitance measurement mode	Electrostatic capacitance is measured from the charged current that flows toward the capacitance generated between the transmit and receive electrodes used in the mutual capacitance method.
	Current measurement mode	Current from a measurement pin is measured.
Calibration mode		Characteristic correction of the current control oscillator for measurement
Noise prevention		Synchronous noise prevention, high-pass noise prevention Majority decision by multi-frequency measurement
Adjustment for each pin		Offset current adjustment function Sensor drive pulse frequency specification Measurement time specification
Measurement start conditions		Software trigger External trigger (ELC)
Low-power function		SNOOZE function supported
Interrupt requests	DTC activation source/ DTC interrupt source	Request to write to a configuration register of an individual CTS2 channel Request to transfer data measured by the CTS2
	Interrupt source	Measurement end interrupt
Transmission power switching of the mutual capacitance method		The power for transmission with the mutual capacitance method is switchable to any from among VDD for I/O ports, VDD for the internal logic power supply, and a dedicated VDD.

As shown in **Figure 27 - 3**, the CTS2 consists of a status control block, a trigger control block, a clock control block, a channel control block, a port control block, a sensor drive pulse generator, a measurement block, an interrupt block, an IO block, a SNOOZE control block, and SFRs.

Figure 27 - 3 CTSU Block Diagram

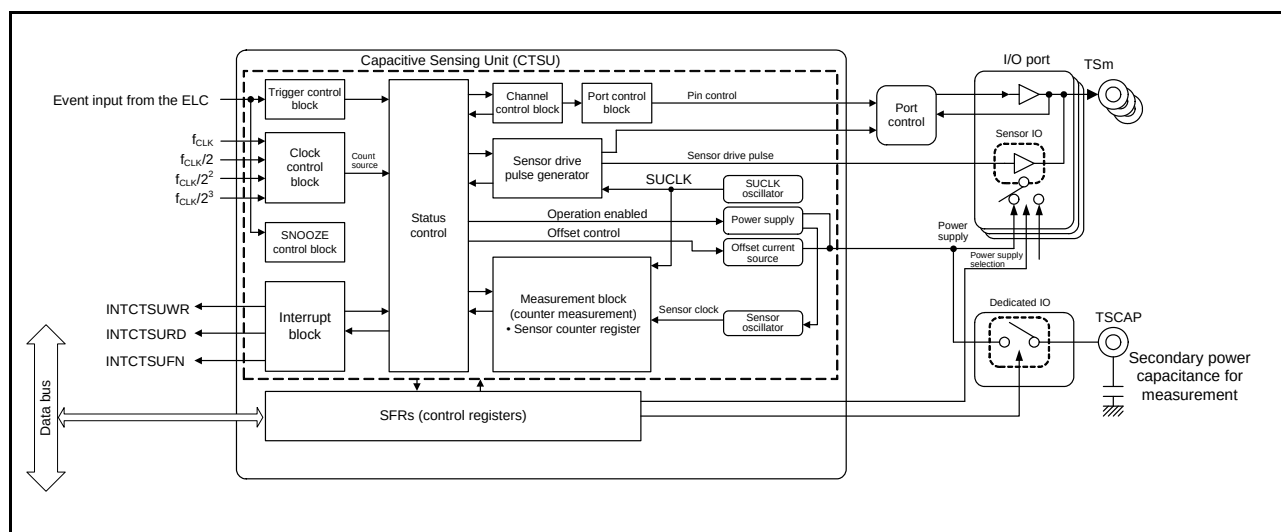


Figure 27 - 4 Sensor Drive Pulse Output Clock Configuration

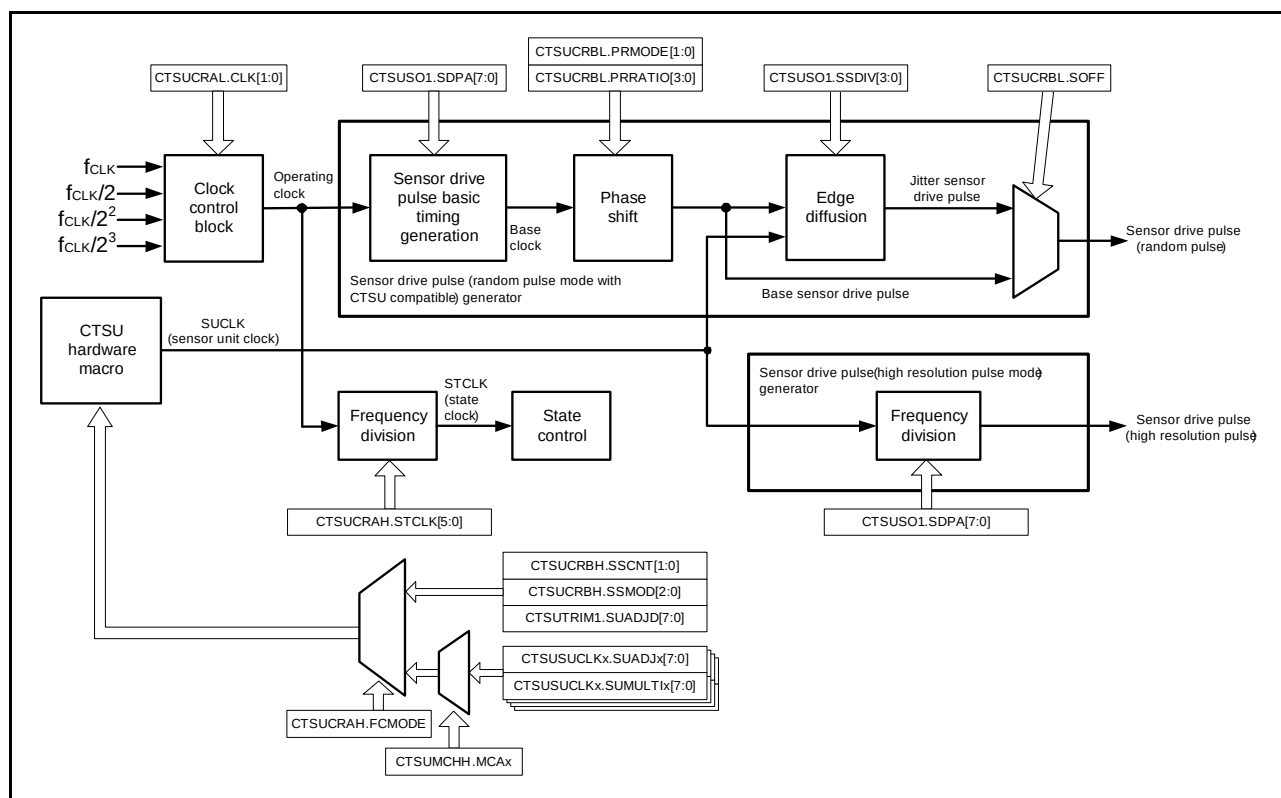


Table 27 - 2 External Pins Used in CTSU

Pin name	Input/output	Function
TSm (m = 00 to 28)	Input/output	Electrostatic capacitance measurement pin, transmit pin in the mutual capacitance method, active shield control pin, or current measurement pin
TSCAP	Input/output	Connection pin to capacitor for measurement secondary power

27.2 Registers for Controlling the Capacitive Sensing Unit

The following registers are used to control the capacitive sensing unit.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH)
- CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH)
- CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH)
- CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)
- CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)
- CTSU status register L (CTSUSRL)
- CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)
- CTSU sensor counter registers L and H (CTSUSC, CTSUUC)
- CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1)
- CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)
- CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1)
- CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3)

27.2.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the CTSU is to be used, be sure to set bit 0 (CTS2EN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 27 - 5 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	<0>						
PER1	0	SMS2EN	0	TML32EN	DTC2EN	UTA2EN	0	CTS2EN						
<table><tr><td>CTS2EN</td><td>Control of supply of an input clock to the CTSU</td></tr><tr><td>0</td><td>Stops supply of an input clock. • The SFRs used by the CTSU cannot be written. • When an SFR used by the CTSU is read, the value returned is 00H or 0000H. </td></tr><tr><td>1</td><td>Enables supply of an input clock. • The SFRs used by the CTSU can be read and written. </td></tr></table>									CTS2EN	Control of supply of an input clock to the CTSU	0	Stops supply of an input clock. • The SFRs used by the CTSU cannot be written. • When an SFR used by the CTSU is read, the value returned is 00H or 0000H.	1	Enables supply of an input clock. • The SFRs used by the CTSU can be read and written.
CTS2EN	Control of supply of an input clock to the CTSU													
0	Stops supply of an input clock. • The SFRs used by the CTSU cannot be written. • When an SFR used by the CTSU is read, the value returned is 00H or 0000H.													
1	Enables supply of an input clock. • The SFRs used by the CTSU can be read and written.													

Caution The functions that are mounted depend on the product. For details on the PER1 register, see the description in Section 6 Clock Generator.

27.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 27 - 6 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	<0>						
PRR1	0	SMSRES	0	TML32RES	0	0	0	CTSURES						
<table><tr><td>CTSURES</td><td>Control resetting of the CTSU</td></tr><tr><td>0</td><td>The CTSU is released from the reset state.</td></tr><tr><td>1</td><td>The CTSU is in the reset state. • The SFRs for use with the CTSU are initialized. </td></tr></table>									CTSURES	Control resetting of the CTSU	0	The CTSU is released from the reset state.	1	The CTSU is in the reset state. • The SFRs for use with the CTSU are initialized.
CTSURES	Control resetting of the CTSU													
0	The CTSU is released from the reset state.													
1	The CTSU is in the reset state. • The SFRs for use with the CTSU are initialized.													

Caution Be sure to clear bits 7, 5, 3, 2, and 1 to 0.

27.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH)

The CTSUCRAL register can be accessed in 8-bit units (CTSUCR1 as upper part and CTSUCR0 as lower part).
The CTSUCRAH register can be accessed in 8-bit units (CTSUCR3 as upper part and CTSUCR2 as lower part).

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (1/7)

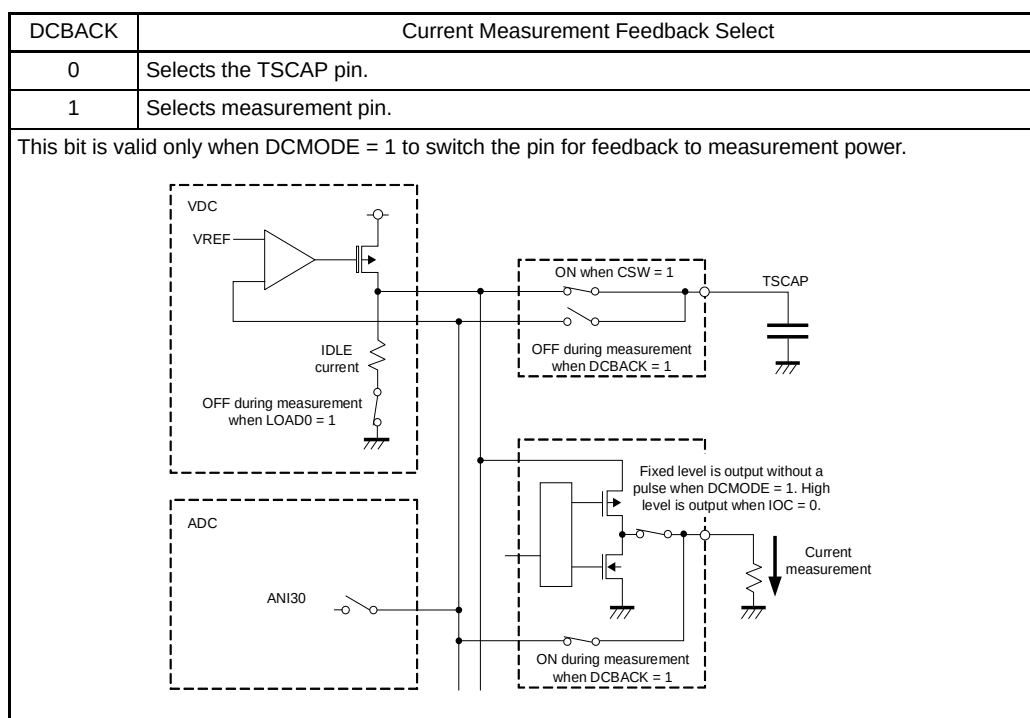
Address: F0500H, F0501H (CTSUCRAL), F0502H, F0503H (CTSUCRAH)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRAH	DCBACK	DCMODE	STCLK[5:0]					
	7	6	5	4	3	2	1	0
	FCMODE	SDPSEL	POSEL[1:0]		LOAD1	LOAD0	ATUNE2	0

Symbol	15	14	13	12	11	10	9	8
CTSUCRAL	MD1	MD0	CLK[1:0]		ATUNE1	ATUNE0	CSW	PON
	7	6	5	4	3	2	1	0
	TXVSEL	TXVSEL2	PUMPON	INIT	0	SNZ	CAP	STRT



DCMODE	Current Measurement Mode Select
0	Normal mode
1	Current measurement mode

The current measurement mode is used to measure current without switched capacitor operation.
Outputs a fixed value to the measurement pin according to the IOC bit and stops the drive pulse.
IOC = 0: High-level output IOC = 1: Low-level output

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (2/7)

STCLK[5:0]						STCLK Select
0	0	0	0	0	0	Operating clock divided by 2
0	0	0	0	0	1	Operating clock divided by 4
0	0	0	0	1	0	Operating clock divided by 6
0	0	0	0	1	1	Operating clock divided by 8
:	:	:	:	:	:	:
1	1	1	1	1	0	Operating clock divided by 126
1	1	1	1	1	1	Operating clock divided by 128
These bits set the division value for the operating clock to obtain STCLK (state clock). STCLK is related to measurement time and clock recovery cycle. The division value is determined by the following equation. Division value = (STCLK[5:0] + 1) × 2 (divided by 2 to 128) Setting STCLK to 0.5 MHz (2 μs) is recommended.						

FCMODE	Sensor unit clock (SUCLK) Select
0	SUCLK is used as frequency diffusion clock.
1	SUCLK is used as recovery clock for multi-clock measurement. To use SUCLK as recovery clock, set the CTSUDBGR1.CCOCLK bit to 1.

- When the FCMODE bit is 0 (SUCLK used as frequency diffusion clock)

As specified in the CTSUTRIM1.SUADJD[7:0], CTSUCRBH.SSCNT[1:0], and CTSUCRBH.SSMOD[2:0], SUCLK is generated when the digital oscillator oscillates and the frequency is spectrum-diffused.
- When the FCMODE is 1 (SUCLK used as recovery clock for multi-clock measurement)

As specified in the CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, or CTSUSUCLK3 register, SUCLK is generated when the clock recovery control is performed.

Before setting the FCMODE bit to 1, set the CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, or CTSUSUCLK3 register.

The recovery is performed in status 0 (non-measurement state) for all selected clocks during measurement. The SUADJx bits in the CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, or CTSUSUCLK3 register are updated by recovery (x = 0 to 3).

When the FCMODE bit is 1, do not change the CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, or CTSUSUCLK3 register setting.

<Relationship between SDPSEL and FCMODE>

SDPSEL	FCMODE	Operation
0	0	Random pulse mode (CTS2 compatible setting)
1	1	Sensor unit clock (SUCLK) mode Used for multi-clock measurement
Other than above		Setting prohibited

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (3/7)

SDPSEL	Sensor Drive Pulse Select
0	Random pulse mode (CTSUCRAL compatible setting) The operating clock divided by the settings of the CTSUCRAL1.SDPA[7:0] bits is used as the base clock, and the sensor drive pulse is obtained by phase-shifting the base clock using the random number generated according to the settings of the CTSUCRALBL.PRMODE[1:0] and CTSUCRALBL.PRRATIO[3:0] bits. It is also possible to apply jitter by the frequency diffusion clock.
1	Sensor unit clock (SUCLK) mode The sensor drive pulse is obtained by applying frequency recovery based on fCLK to generate SUCLK and dividing it by the settings of the CTSUCRAL1.SDPA[7:0] bits.
The SDPSEL bit selects the sensor drive pulse.	

POSEL[1:0]		Non-measured Pin Output Select
0	0	Low-level output (from the I/O port regardless of the settings of the TXVSEL and TXVSEL2 bits)
0	1	Hi-Z
1	0	Low-level output (by the settings of the TXVSEL and TXVSEL2 bits)
1	1	Same phase (transmission) pulse output (by the settings of the TXVSEL and TXVSEL2 bits)

The POSEL[1:0] bits select the non-measured pin output.

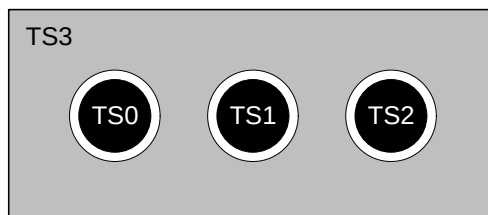
[Example] When measurement is performed by the self-capacitance method using the electrode shown below to enable the active shield function:

- Shield pin: TS3

Set the transmit/receive control register to 1 to output a transmit pulse to use TS3 as a shield pin.

- Measurement pins: TS0 to TS2

Set the POSEL[1:0] bits to 11B. The same pulse as the shield pin is output from the non-measured pin.



LOAD[1:0]		Measurement Load Control
0	0	2.5-μA constant current load mode
0	1	No-load mode
1	0	20-μA constant current load mode
1	1	Resistive load mode (for calibration)
To place the measurement load in the resistive load mode, set these bits to 01B and then set them to 11B.		

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (4/7)

ATUNE2	Analog Adjustment 2
0	In accord with the setting of the ATUNE1 bit
1	When ATUNE1 is 0: 20 μ A (1/1) When ATUNE1 is 1: 160 μ A (1/8) current measurement mode
The ATUNE2 bit sets the current mirror ratio of the measurement power-supply current to the current control oscillator input current.	

MD1	Measurement Mode Select 1
0	Self-capacitance method (single measurement) When CHTRCx is set to 1 (transmission), the same phase pulse is output from the TSm pin for measurement. When multiple CHTRCx bits are set to 1, measurement is scanned.
1	Mutual capacitance method (double-measurement) Set the CHTRCx bit to 1 (transmission) to handle measurement. The same phase pulse is output from the TSm pin in the first measurement. The reverse phase pulse is output from the TSm pin in the second measurement.
Set the MD1 bit to 0 to select the self-capacitance method. Set the MD1 bit to 1 to select the mutual capacitance method.	

MD0	Measurement Mode Select 0
0	Single scan mode
1	Multi-scan mode
The MD0 bit selects single scan mode or multi-scan mode.	

CLK[1:0]		Operating Clock Select
0	0	fCLK
0	1	fCLK/2
1	0	fCLK/4
1	1	fCLK/8
The CLK[1:0] bits are used to divide the CPU/peripheral hardware clock (fCLK) using the prescaler.		

ATUNE1	Analog Adjustment 1
0	80 μ A (1/4)
1	40 μ A (1/2)
The ATUNE1 bit sets the current mirror ratio of the measurement power-supply current to the current control oscillator input current.	

ATUNE0	Analog Adjustment 0
0	Measurement power-supply voltage = 1.5 V This setting cannot be used if VDD is less than 2.4 V.
1	Measurement power-supply voltage = 1.2 V
This bit is used to change voltage to suit the system's power supply specifications.	

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (5/7)

CSW	Secondary Power Supply Capacitor for Measurement
0	Turns off the external capacitance connection switch.
1	Turns on the external capacitance connection switch.
This bit controls charging of the secondary power supply capacitance for measurement connected to the TSCAP pin by turning the capacitance switch on or off. After the capacitance switch is turned on, wait for the specified time until the capacitance connected to the TSCAP pin is charged, and then set STRT to 1 to start measurement. Before starting measurement, use the I/O port to output a low level to the TSCAP pin, and discharge the secondary power-supply capacitance for measurement. Set the CSW bit to 1 before setting the PON bit to 1. When $V_{DD} < 4.5$ V, set the PUMPON bit to 1 and then set the CSW bit to 1.	

PON	Measurement Power On
0	Power off
1	Power on
The PON bit turns on the VDC for measurement to supply the power (internal voltage) for measurement. 1.2- or 1.5-V power is supplied depending on the setting of the ATUNE0 bit. Do not set this bit to 1 when the setting of CSW is 0.	

TXVSEL	TXVSEL2	Transmission Power Supply Select
0	0	V_{DD} for I/O ports The I/O buffer power supply for port pins is selected.
0	1	Dedicated V_{DD} This setting is recommended when the transmission power is for use with the mutual capacitance method. Note
1	0	V_{DD} for the internal logic power supply Use this setting when the active shield function is in use.
1	1	Dedicated V_{DD} This setting is recommended when the transmission power is for use with the mutual capacitance method. Note

Note The same transmission power supply is selected when the setting of TXVSEL2 is 1.

PUMPON	Boost Circuit Control
0	Boost power off $V_{CCX3} = V_{DD}$
1	Boost power on $V_{CCX3} \approx 4.5$ V
The PUMPON bit controls the boost power voltage (V_{CCX3}). When V_{DD} is 4.5 V or lower, turn on the boost power.	

INIT	Control Block Initialization
0	—
1	Initializes the internal control registers.
To forcibly terminate operation, be sure to set the STRT bit to 0 and the INIT bit to 1 simultaneously. In this case, operation stops and the internal control registers are initialized. Do not set the STRT bit to 1 (CTS2 operation start) and the INIT bit to 1 simultaneously. The TXVSEL2 bit is read as 0.	

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (6/7)

SNZ	SNOOZE Enable				
0	Disables the SNOOZE function.				
1	Enables the SNOOZE function.				

The SNZ bit enables or disables the SNOOZE function when an external trigger is selected (CAP = 1). Setting this bit to 1 drives the CTSU hardware macro into the suspended state to enable low-power operation in the standby state.

<CTSU hardware state control>

PON	SNZ	CAP	STRT	Trigger	State of the CTSU
0	0	0	0	—	Stopped
1	0	0	0	—	State until measurement starts (VDC = ON)
1	0	0	1	—	Measurement in the normal mode (VDC = ON)
1	1	1	0	—	Prepared for measurement start by an external trigger (VDC = OFF)
1	1	1	1	Not present	Suspended (waiting for a trigger) (VDC = OFF)
1	1	1	1	Present	Measurement in the SNOOZE mode (VDC = ON) ^{Note}
1	1	0	0	—	Software suspended (software setting) (VDC = OFF)
Other than above					Setting prohibited

Note When a trigger is generated in the STOP mode, measurement is handled in the SNOOZE mode.

The SNZ bit enables SNOOZE operation. In the external trigger waiting state enabled by setting the STRT bit to 1, the CPU can enter STOP mode. When a falling edge of the external trigger is detected during STOP mode, the CTSU sends a clock request to the clock generating block and enters the SNOOZE state to start measurement. After the measurement end interrupt, clear this bit to 0 by software.

The software suspended state (software setting) in this table is used when the software of a system without SNOOZE function suspends the CTSU hardware macro to enable low-power operation. In this case, set the SNZ bit to 0 after the CPU returns to the previous state by an external interrupt, and then set the STRT bit to 1 to start measurement upon a software trigger.

CAP	Measurement Start Trigger Select
0	Software trigger
1	External trigger (ELC)

The CAP bit specifies the measurement start condition.

For details, see the STRT bit explanation.

Figure 27 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (7/7)

STRT	Measurement Operation Start	
0	Measurement stopped	
1	Measurement operating	

The STRT bit specifies whether to start or stop CTSU operation.

When CAP is 0, writing 1 to the STRT bit starts measurement. At the end of measurement, this bit is cleared to 0 by hardware.

When CAP is 1, writing 1 to this bit drives the CTSU into the external trigger waiting state. Measurement starts at the falling edge of the external trigger. After the measurement is complete, the CTSU waits for the next external trigger and continues operation.

The following table shows the CTSU states.

<CTS state>

STRT	CAP	CTS state
0	0	Stopped
0	1	Stopped
1	0	Measurement in progress
1	1	Measurement in progress or waiting for an external trigger ^{Note}

Note The state can be read from the CTSUSRL.STC[2:0] bits.
Measurement in progress: CTSUSRL.STC[2:0] ≠ 000B
While waiting for an external trigger: CTSUSRL.STC[2:0] = 000B

If 1 is written to the STRT bit by software when the STRT bit is already set to 1, this writing is ignored and operation continues.

To forcibly terminate operation by software when the STRT bit is set to 1, be sure to set the STRT bit to 0 and the INIT bit to 1 simultaneously.

27.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH)

The CTSUCRBL register can be accessed in 8-bit units (CTSUSST as upper part and CTSUSDPRS as lower part).
The CTSUCRBH register can be accessed in 8-bit units (CTSUDCLKC as upper part).

Figure 27 - 8 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (1/3)

Address: F0504H, F0505H (CTSUCRBL), F0506H, F0507H (CTSUCRBH)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRBH	0	0	SSCNT[1:0]		0	SSMOD[2:0]		
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
Symbol	15	14	13	12	11	10	9	8
CTSUCRBL	SST[7:0]							
	7	6	5	4	3	2	1	0
	PROFF	SOFF	PRMODE[1:0]		PRRATIO[3:0]			
SSCNT[1:0]			SUCLK Diffusion Control					
0	0	SSADJ + 0						
0	1	SSADJ + 1						
1	0	SSADJ + 2						
1	1	SSADJ + 3						
The SSCNT[1:0] bits adjust the diffusion clock frequency.								
The setting of 11B is for random-pulse mode (CTS2 compatible).								
SSMOD[2:0]			SUCLK Diffusion Mode Select					
0	0	0	256 cycles					
0	0	1	384 cycles					
0	1	0	512 cycles					
0	1	1	1024 cycles					
1	1	1	No diffusion					
Other than above			Setting prohibited					
The SUCLK clock is used for the SSCG modulation of the CTS2.								
The SSMOD[2:0] and SSCNT[1:0] bits are used to determine the FM modulation frequency.								
The SSMOD[2:0] bits set the FM modulation cycle.								

Figure 27 - 8 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (2/3)

SST[7:0]					Sensor Stabilization Wait Time Control			
The SST[7:0] bits set the period from the start of sensor drive pulse supply until the TSCAP pin voltage becomes stable.								
<When CTSUCRAH.SDPSEL = 0>								
The table below shows the stabilization time in units of base sensor drive pulse cycles.								
0	0	0	0	0	0	0	0	2 cycles
0	0	0	0	0	0	0	1	4 cycles
0	0	0	0	0	0	1	0	6 cycles
0	0	0	0	0	0	1	1	8 cycles
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	0	510 cycles
1	1	1	1	1	1	1	1	512 cycles
<When CTSUCRAH.SDPSEL = 1>								
The table below shows the stabilization time in units of STCLK cycles.								
0	0	0	0	0	0	0	0	1 cycle
0	0	0	0	0	0	0	1	2 cycles
0	0	0	0	0	0	1	0	3 cycles
0	0	0	0	0	0	1	1	4 cycles
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	0	255 cycles
1	1	1	1	1	1	1	1	256 cycles
The TSCAP pin voltage is stabilized by supplying sensor drive pulses. The SST[7:0] value is related to the number of cycles as shown below.								
• When CTSUCRAH.SDPSEL = 0								
The stabilization wait time is specified by the number of cycles of the base sensor drive pulse. Number of cycles = 2 × (value of these bits + 1) Set the stabilization wait time within the following range. Number of cycles set by CTSUCRBL.SST[7:0] ≥ (CTSUCRBL.PRRATIO[3:0] + 1)								
• When CTSUCRAH.SDPSEL = 1								
The stabilization wait time is specified by the number of STCLK cycles. Number of cycles = 1 × (value of these bits + 1)								

PROFF	Pseudo-Random Number Disable Control
0	Pseudo-random number control is enabled.
1	Pseudo-random number control is disabled.

The PROFF bit disables the pseudo-random number control.

A value of 1 or 0 is output per cycle to generate pseudo-random numbers (1-bit random number generation)

When PROFF is 1, one cycle is added to the cycles specified by the PRMODE[1:0] bits.

Figure 27 - 8 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (3/3)

SOFF	Jitter Disable Control	
0	Applies jitter.	
1	Does not apply jitter.	
The SOFF bit sets whether to apply jitter to the sensor drive pulse to prevent synchronous noise. The output of the sensor drive pulse is selected from the base sensor drive pulse or the jittered sensor drive pulse.		

PRMODE[1:0]		Pseudo-Random Number Generation Cycle
0	0	255 cycles (When PROFF = 1: 256 cycles)
0	1	63 cycles (When PROFF = 1: 64 cycles)
1	0	31 cycles (When PROFF = 1: 32 cycles)
1	1	3 cycles (When PROFF = 1: 4 cycles)

PRRATIO[3:0]	Phase Shift Frequency
The PRRATIO[3:0] bits specify the phase shift frequency of the base clock using a pseudo-random number. These bits become a factor to determine the measurement period.	

27.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH)

The CTSUMCHL register can be accessed in 8-bit units (CTSUMCH1 as upper part and CTSUMCH0 as lower part).
The CTSUMCHH register can be accessed in 8-bit units (CTSUMFAF as lower part).

Figure 27 - 9 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH) (1/2)

Address: F0508H, F0509H (CTSUMCHL), F050AH, F050BH (CTSUMCHH)

After reset: 0000H, 3F3FH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8						
CTSUMCHH	0	0	0	0	0	0	0	0						
	7	6	5	4	3	2	1	0						
	0	0	0	0	MCA3	MCA2	MCA1	MCA0						
Symbol	15	14	13	12	11	10	9	8						
CTSUMCHL	0	0	MCH1[5:0]											
	7	6	5	4	3	2	1	0						
	0	0	MCH0[5:0]											
	<table><tr><td>MCAx</td><td>Multi-clock x Enable</td></tr><tr><td>0</td><td>Disabled</td></tr><tr><td>1</td><td>Enabled</td></tr></table>								MCAx	Multi-clock x Enable	0	Disabled	1	Enabled
MCAx	Multi-clock x Enable													
0	Disabled													
1	Enabled													
	The MCAx bit sets valid clock for measurement using SUCLK with two or more frequencies. For measurement using three clocks, set three bits out of the four bits (MCA3 to MCA0) to 1. When making measurement using SUCLK with two or more frequencies, this bit enables or disables the target clock. When measurement starts with two or more MCAx bits enabled, valid channels are scanned in ascending order from the MCA0 bit. After a pin has been measured, if the next valid clock is present, the same pin is measured. If there is no valid clock, measurement shifts to the next pin. The SUCLK frequency is set by the CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, and CTSUSUCLK3 registers. The following shows the relationship between the MCAx bits and the CTSUSUCLK0 to CTSUSUCLK3 registers. • MCA0: Enables SUCLK set by the CTSUSUCLK0.SUMULTI0[7:0] and CTSUSUCLK0.SUADJ0[7:0] bits • MCA1: Enables SUCLK set by the CTSUSUCLK1.SUMULTI1[7:0] and CTSUSUCLK1.SUADJ1[7:0] bits • MCA2: Enables SUCLK set by the CTSUSUCLK2.SUMULTI2[7:0] and CTSUSUCLK2.SUADJ2[7:0] bits • MCA3: Enables SUCLK set by the CTSUSUCLK3.SUMULTI3[7:0] and CTSUSUCLK3.SUADJ3[7:0] bits													

Figure 27 - 9 Format of CTSU Measurement Channel Registers L and H (CTS2MCHL, CTS2MCHH) (2/2)

MCH1[5:0]						Measurement Channel 1
- In single scan mode (CTS2CRAL.MD0 = 0), the MCH1[5:0] bits set the transmit channel to be measured. Do not set channels that are not to be measured using the CTS2CHACAH, CTS2CHACAL, CTS2CHACBH, or CTS2CHACBL register. If such channels are set, measurement is completed immediately after it starts. - In multi-scan mode (CTS2CRAL.MD0 = 1), the MCH0[5:0] bits indicate the value of the transmit channel that is being measured, and writing to these bits has no effect (cleared at the beginning of measurement).						
0	0	0	0	0	0	TS0
0	0	0	0	0	1	TS1
0	0	0	0	1	0	TS2
0	0	0	0	1	1	TS3
:	:	:	:	:	:	:
1	1	1	1	1	0	TS62
1	1	1	1	1	1	TS63
Do not modify these bits during measurement (CTS2CRAL.STRT = 1). Otherwise, operation is not guaranteed. When measurement is stopped, these bits become 111111B.						

MCH0[5:0]						Measurement Channel 0
- In single scan mode (CTS2CRAL.MD0 = 0), the MCH0[5:0] bits set the receive channel to be measured. Do not set channels that are not to be measured using the CTS2CHACAH, CTS2CHACAL, CTS2CHACBH, or CTS2CHACBL register. If such channels are set, measurement is completed immediately after it starts. - In multi-scan mode (CTS2CRAL.MD0 = 1), the MCH1[5:0] bits indicate the value of the receive channel that is being measured, and writing to these bits has no effect (cleared at the beginning of measurement).						
0	0	0	0	0	0	TS0
0	0	0	0	0	1	TS1
0	0	0	0	1	0	TS2
0	0	0	0	1	1	TS3
:	:	:	:	:	:	:
1	1	1	1	1	0	TS62
1	1	1	1	1	1	TS63
Do not modify these bits during measurement (CTS2CRAL.STRT = 1). Otherwise, operation is not guaranteed. When measurement is stopped, these bits become 111111B.						

27.2.6 CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)

The CTSUCHACAL register can be accessed in 8-bit units (CTSUCHAC1 as upper part and CTSUCHAC0 as lower part).

The CTSUCHACAH register can be accessed in 8-bit units (CTSUCHAC3 as upper part and CTSUCHAC2 as lower part).

The CTSUCHACBL register can be accessed in 8-bit units (CTSUCHAC5 as upper part and CTSUCHAC4 as lower part).

The CTSUCHACBH register can be accessed in 8-bit units (CTSUCHAC7 as upper part and CTSUCHAC6 as lower part).

Figure 27 - 10 Format of CTSU Channel Enable Control Registers AL, AH, BL, and BH
(CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)

Address: F050CH, F050DH (CTSUCHACAL), F050EH, F050FH (CTSUCHACAH),
F0510H, F0511H (CTSUCHACBL), F0512H, F0513H (CTSUCHACBH)

After reset: 0000H, 0000H, 0000H, 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHACAH	CHAC31	CHAC30	CHAC29	CHAC28	CHAC27	CHAC26	CHAC25	CHAC24

7	6	5	4	3	2	1	0
CHAC23	CHAC22	CHAC21	CHAC20	CHAC19	CHAC18	CHAC17	CHAC16

Symbol	15	14	13	12	11	10	9	8
CTSUCHACAL	CHAC15	CHAC14	CHAC13	CHAC12	CHAC11	CHAC10	CHAC09	CHAC08

7	6	5	4	3	2	1	0
CHAC07	CHAC06	CHAC05	CHAC04	CHAC03	CHAC02	CHAC01	CHAC00

Symbol	15	14	13	12	11	10	9	8
CTSUCHACBH	CHAC63	CHAC62	CHAC61	CHAC60	CHAC59	CHAC58	CHAC57	CHAC56

7	6	5	4	3	2	1	0
CHAC55	CHAC54	CHAC53	CHAC52	CHAC51	CHAC50	CHAC49	CHAC48

Symbol	15	14	13	12	11	10	9	8
CTSUCHACBL	CHAC47	CHAC46	CHAC45	CHAC44	CHAC43	CHAC42	CHAC41	CHAC40

7	6	5	4	3	2	1	0
CHAC39	CHAC38	CHAC37	CHAC36	CHAC35	CHAC34	CHAC33	CHAC32

CHACx	Channel Enable Control
0	Do not measure pins.
1	Measure pins.
The CHACx bits set whether TSm pin measurement is required.	

(Cautions are listed on the next page.)

Caution 1. CHAC63 to CHAC0 are used for setting measurement of the TS63 to TS0 pins.

Bits for pins that are not present in the device in use are read as 0. When writing, write 0 to these bits.

Caution 2. Set the CHACx bits to 1 for transmit and receive pins whose electrostatic capacitance is to be measured.

Caution 3. Set the CTSUCHACAH, CTCSUCHACAL, CTSUCHACBH, and CTSUCHACBL registers when CTSUCRAL.STRT is 0.

27.2.7 CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)

The CTSUCHTRCAL register can be accessed in 8-bit units (CTSUCHTRC1 as upper part and CTSUCHTRC0 as lower part).

The CTSUCHTRCAH register can be accessed in 8-bit units (CTSUCHTRC3 as upper part and CTSUCHTRC2 as lower part).

The CTSUCHTRCBL register can be accessed in 8-bit units (CTSUCHTRC5 as upper part and CTSUCHTRC4 as lower part).

The CTSUCHTRCBH register can be accessed in 8-bit units (CTSUCHTRC7 as upper part and CTSUCHTRC6 as lower part).

Figure 27 - 11 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, BL, and BH
(CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)

Address: F0514H, F0515H (CTSUCHTRCAL), F0516H, F0517H (CTSUCHTRCAH),
F0518H, F0519H (CTSUCHTRCBL), F051AH, F051BH (CTSUCHTRCBH)

After reset: 0000H, 0000H, 0000H, 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCAH	CHTRC31	CHTRC30	CHTRC29	CHTRC28	CHTRC27	CHTRC26	CHTRC25	CHTRC24
	7	6	5	4	3	2	1	0
	CHTRC23	CHTRC22	CHTRC21	CHTRC20	CHTRC19	CHTRC18	CHTRC17	CHTRC16

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCAL	CHTRC15	CHTRC14	CHTRC13	CHTRC12	CHTRC11	CHTRC10	CHTRC09	CHTRC08
	7	6	5	4	3	2	1	0
	CHTRC07	CHTRC06	CHTRC05	CHTRC04	CHTRC03	CHTRC02	CHTRC01	CHTRC00

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCBH	CHTRC63	CHTRC62	CHTRC61	CHTRC60	CHTRC59	CHTRC58	CHTRC57	CHTRC56
	7	6	5	4	3	2	1	0
	CHTRC55	CHTRC54	CHTRC53	CHTRC52	CHTRC51	CHTRC50	CHTRC49	CHTRC48

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCBL	CHTRC47	CHTRC46	CHTRC45	CHTRC44	CHTRC43	CHTRC42	CHTRC41	CHTRC40
	7	6	5	4	3	2	1	0
	CHTRC39	CHTRC38	CHTRC37	CHTRC36	CHTRC35	CHTRC34	CHTRC33	CHTRC32

CHTRCx	Channel Transmit/Receive Control
0	Reception
1	Transmission
The CHTRCx bits assign the TSm pins to reception or transmission.	

(Cautions are listed on the next page.)

- Caution 1.** CHTRC63 to CHTRC0 are used for setting transmit/receive control of the TS63 to TS0 pins.
Bits for pins that are not present in the device in use are read as 0. When writing, write 0 to these bits.
- Caution 2.** The CHTRCx bits assign the TSm pins to reception or transmission.
When MD1 is set to 0, if one of these bits is set to 1 (transmission), the corresponding pin can be used for shield signal output.
However, when setting as a shield output, do not set two or more bits to 1.
- Caution 3.** Set the CHTRCx bits to 1 for the TSm pin whose electrostatic capacitance is to be measured. In the mutual capacitance method, set the CHTRCx bits to 1 for transmit and receive pins whose electrostatic capacitance is to be measured.
- Caution 4.** Set the CTSUCHTRCAH, CTSUCHTRCAL, CTSUCHTRCBH, and CTSUCHTRCBL registers when CTSUCRAL.STRT is 0.

27.2.8 CTSU status register L (CTSUSRL)

The CTSUSRL register can be accessed in 8-bit units (CTSUST as upper part and CTSUST1 as lower part).

Figure 27 - 12 Format of CTSU Status Register L (CTSUSRL) (1/3)

Address: F051CH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSRL	PS	SUCKOVF	SENSOVF	DTSR	0	STC[2:0]		
	7	6	5	4	3	2	1	0
	ICOMP0	ICOMP1	ICOMPRST	0	0	0	MFC[1:0]	

PS	Mutual Capacitance Measurement Status
0	First measurement
1	Second measurement

The PS bit indicates the measurement status in double-measurement (CTSUCRAL.MD1 = 1).

This bit indicates whether the measurement is the first or second of double-measurement per channel.

This bit is 0 when measurement is stopped or during single measurement (CTSUCRAL.MD1 = 0).

SUCKOVF	SUCLK Counter Overflow
0	No overflow occurred.
1	An overflow occurred.

The SUCKOVF bit indicates whether the SUCLK counter has overflowed. When an overflow occurs, the result of measurement is read from the CTSUUC.UC[15:0] bits as FFFFH.

Even if an overflow occurs, measurement processing continues for the set period.

Occurrence of an overflow generates no interrupt. Therefore, read the measurement result of each channel after measurement is complete (after a measurement end interrupt is generated) to check an overflowed channel.

This bit is cleared when 0 is written after 1 is read by software. This bit is also cleared by the CTSUCRAL.INIT bit.

Figure 27 - 12 Format of CTSU Status Register L (CTSUSRL) (2/3)

SENSOVF	Sensor Counter Overflow		
0	No overflow occurred.		
1	An overflow occurred.		
The SENSOVF bit indicates whether the sensor counter has overflowed. When an overflow occurs, the result of measurement is read from the CTSUSC.SC[15:0] bits as FFFFH. Even if an overflow occurs, measurement processing continues for the set period. Occurrence of an overflow generates no interrupt. Therefore, read the measurement result of each channel after measurement is complete (after a measurement end interrupt is generated) to check an overflowed channel. This bit is cleared when 0 is written after 1 is read by software. This bit is also cleared by the CTSUCRAL.INIT bit.			

DTSR	Data Transfer Status		
0	The measurement result has been read.		
1	The measurement result has not been read.		
The DTSR bit indicates whether the measurement result stored in the sensor counter has been read. This bit is set to 1 on completion of measurement and is cleared to 0 when the CTSUSC register is read by software or the DTC. This bit is also cleared by the CTSUCRAL.INIT bit.			

STC[2:0]			Measurement Status Counter
0	0	0	Status 0
0	0	1	Status 1
0	1	0	Status 2
0	1	1	Status 3
1	0	0	Status 4
1	0	1	Status 5
The STC[2:0] bits indicate the current measurement status.			

ICOMP0	Reference Resistance Comparison Result		
0	Normal		
1	Abnormal TSCAP voltage		
TSCAP voltage error flag This bit is cleared to 0 when CTSUCRAL.PON is set to 0 or 1 is written to the ICOMPRST bit. When the TSCAP voltage becomes abnormal, the sensor counters at that time are all cleared to 0.			

ICOMP1	Current Error		
0	Normal		
1	Abnormal current		
Current error flag This bit is cleared to 0 when CTSUCRAL.PON is set to 0 or 1 is written to the ICOMPRST bit.			

ICOMPRST	ICOMP0 and ICOMP1 Reset		
Writing 1 to this bit clears the ICOMP1 and ICOMP0 bits to 0.			

Figure 27 - 12 Format of CTSU Status Register L (CTSUSRL) (3/3)

MFC[1:0]		Multi-clock Counter
0	0	Multi-clock 0
0	1	Multi-clock 1
1	0	Multi-clock 2
1	1	Multi-clock 3
The MFC[1:0] bits indicate the clock that is being measured during multi-clock measurement (CTSUCRAH.FCMODE = 1).		

27.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)

Figure 27 - 13 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (1/2)

Address: F0520H, F0521H (CTSUSO0), F0522H, F0523H (CTSUSO1)

After reset: 0000H, 0000H

R/W: R/W

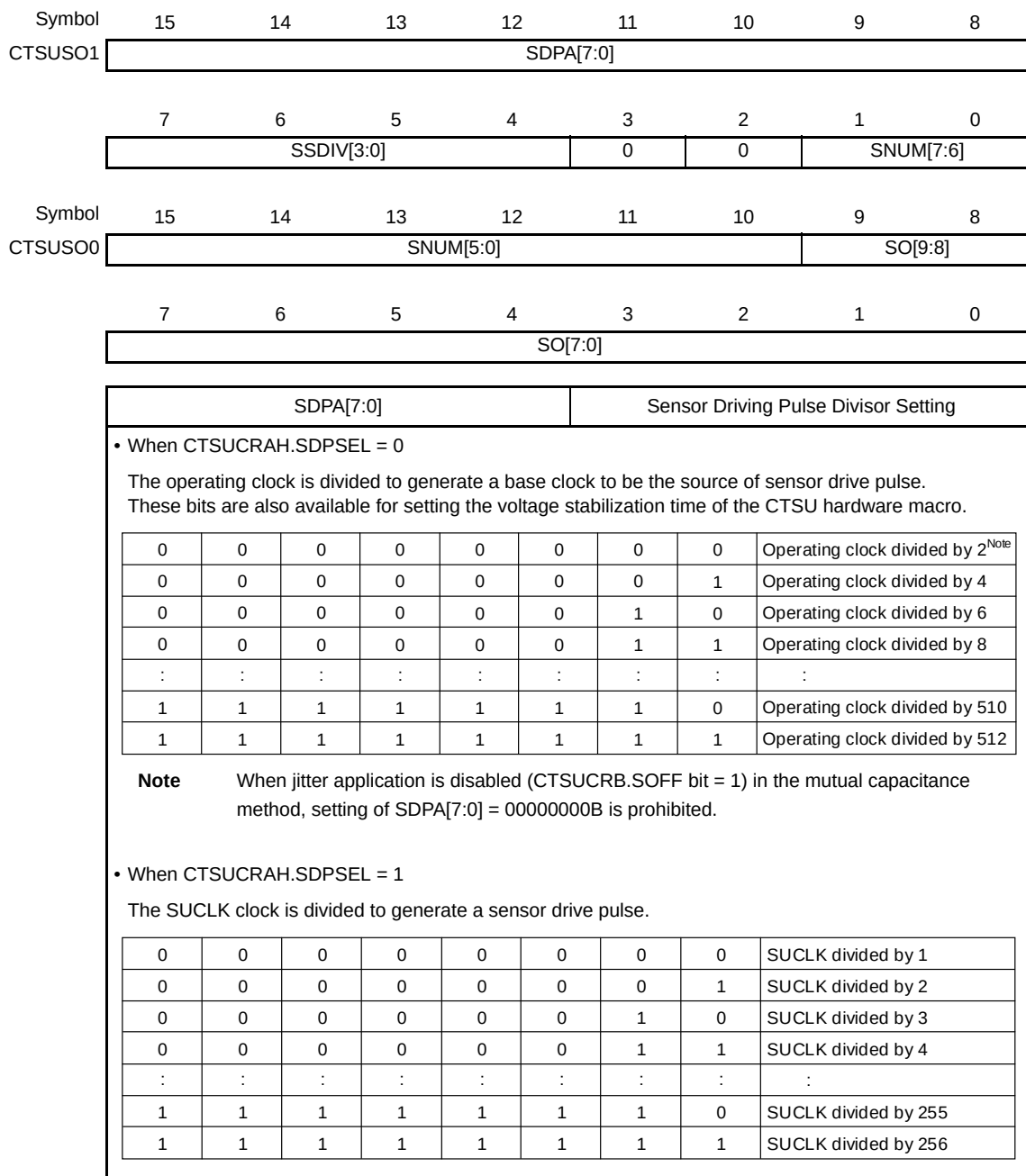


Figure 27 - 13 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (2/2)

SSDIV[3:0]				Spectrum Diffusion Sampling Cycle Control
The SSDIV[3:0] bits are valid only when CTSUCRAH.SDPSEL is 0. Sampling cycle (divided by 1 to 16) can be set for the jitter application function.				
0	0	0	0	Divided by 1
0	0	0	1	Divided by 2
:	:	:	:	:
1	1	1	0	Divided by 15
1	1	1	1	Divided by 16
The SSDIV[3:0] bits set the sampling cycle of the jitter application function. Set the sampling cycle to less than 1/4 of the sensor drive pulse cycle. Set these bits for the pin to be measured after a request to write to a configuration register of an individual CTSU channel (INTCTSUWR) is generated.				

SNUM[7:0]				Measurement Count Setting
- When CTSUCRAH.SDPSEL = 0 The measurement count is set. The SNUM[7:0] bits specify how many times the base unit determined by the CTSUCRBL.PRRATIO[3:0] and CTSUCRBL.PRMODE[1:0] bits is repeated during the measurement period. Measurement count = (SNUM[5:0] + 1) × 2 Set the SNUM[7: 6] bits to 00B. - When CTSUCRAH.SDPSEL = 1 The measurement time is set. Measurement time = (STCLK cycles × 8) × (value of SNUM[7:0] bits + 1) Set these bits for the pin to be measured after a request to write to a configuration register of an individual CTSU channel (INTCTSUWR) is generated.				

SO[9:0]				Sensor Offset Adjustment
The SO[9:0] bits adjust the input current offset of the sensor ICO. * The adjustment value varies with the value of the CTSUCRAL.ATUNE[0] bit. Set these bits for the pin to be measured after a request to write to a configuration register of an individual CTSU channel (INTCTSUWR) is generated.				

After a request to write to a configuration register of an individual CTSU channel (INTCTSUWR) is generated, write to the CTSUSO register, which causes a transition to Status 3. Therefore, when writing a value to the CTSUSO register, set all bits at a time.

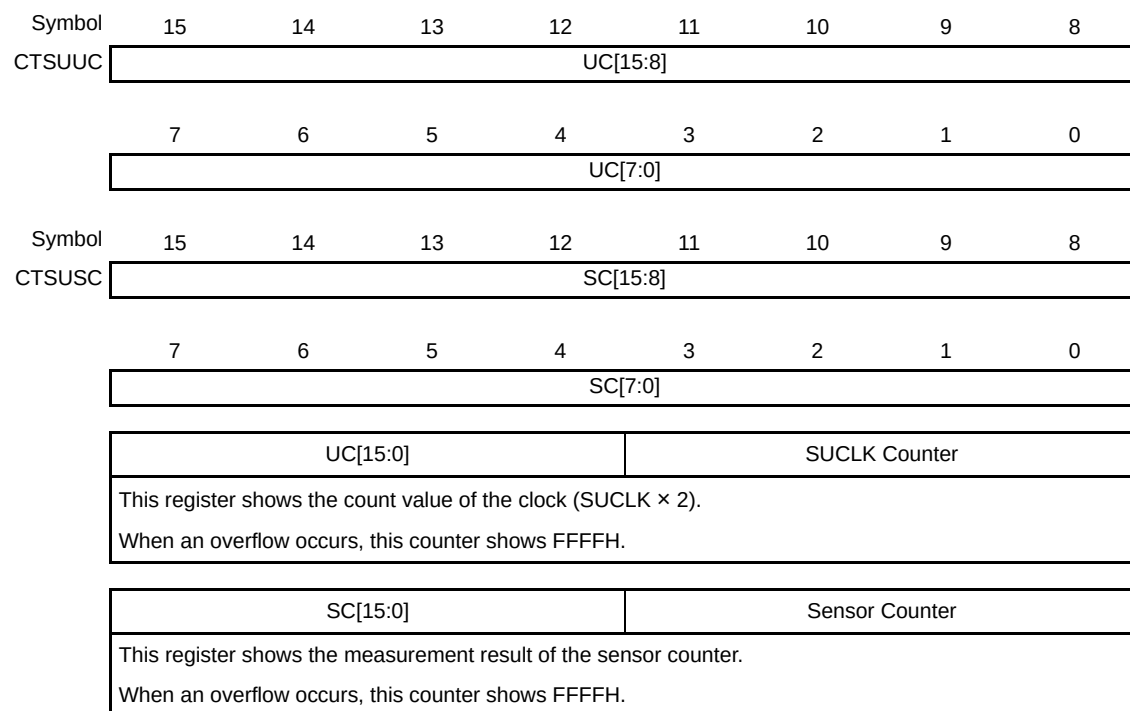
27.2.10 CTSU sensor counter registers L and H (CTSUSC, CTSUUC)

Figure 27 - 14 Format of CTSU Sensor Counter Registers L and H (CTSUSC, CTSUUC)

Address: F0524H, F0525H (CTSUSC), F0526H, F0527H (CTSUUC)

After reset: 0000H, 0000H

R/W: R



27.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1)

Figure 27 - 15 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (1/3)

Address: F0528H, F0529H (CTSUDBGR0), F052AH, F052BH (CTSUDBGR1)

After reset: 0000H, 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUDBGR1	TXREV	CCOCALIB	CCOCLK	DACCLK	SUCARRY	SUMSEL	DACCARRY	DACMSEL
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Symbol	15	14	13	12	11	10	9	8
CTSUDBGR0	0	0	0	0	DCOFF	0	IOC	CNTRDSEL
	7	6	5	4	3	2	1	0
	TSOC	SUCLKEN	CLKSEL0[1:0]	DRV	TSOD	0	0	0

TXREV	Transmit Pin Inverted Output
0	The pulse output from the transmit pin is not inverted.
1	The pulse output from the transmit pin is inverted.

<Single measurement (CTSUCRAL.MD1 = 0)>
0: Same phase of receive pin
1: Reverse phase of receive pin
<Double-measurement (CTSUCRAL.MD1 = 1)>
0: First: Same phase of receive pin; Second: Reverse phase of receive pin
1: First: Reverse phase of receive pin; Second: Same phase of receive pin
Set this bit to 0 for normal measurement (setting to 1 is assumed for debugging).

CCOCALIB	CCO Calibration Mode Select
MODE circuit state 0: Normal mode 1: Oscillator calibration mode The oscillator calibration mode is used when the external current is compared with the current DAC to correct the oscillator characteristics by software.	

CCOCLK	CCO Modulation Circuit Clock Select
0	Operating clock
1	SUCLK

Figure 27 - 15 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (2/3)

DACCLK	DAC Modulation Circuit Clock Select
0	Operating clock
1	SUCLK

SUCARRY	CCO Carry Input (normally set this bit to 0)
0	Normal operation
1	Use this setting in the current oscillator input current test mode.

SUMSEL	Current Oscillator Input Current Matrix Test
0	Normal operation
1	Current oscillator input current test mode
This bit controls the input current for the current oscillator.	

DACCARRY	DAC Upper Current Source Carry Input (normally set this bit to 0)
0	Normal operation
1	Use this setting in the current offset DAC current test mode.

DACMSEL	Current Offset DAC Current Matrix Test
0	Normal operation
1	Current offset DAC current test mode
This bit controls the current for the current offset DAC.	

DCOFF	Down-Conversion Disable
0	Normal operation mode
1	Down-conversion disabled
The DCOFF bit disables conversion of the VDDSENS voltage (disables the P-ch driver). The amplifier circuit of the converter operates as a comparator.	

IOC	Transmit Pin Control
The IOC bit selects the output level from the TSm pin when the CTSUDBGR1.TSOD bit or the CTSUCRAL.DCMODE bit is set to 1. <When CTSUDBGR1.TSOD = 1> 0: Low level is output from the TSm pin. 1: High level is output from the TSm pin. <When CTSUCRAL.DCMODE = 1> 0: High level is output from the TSm pin. 1: Low level is output from the TSm pin. When the CTSUDBGR1.TSOD and CTSUCRAL.DCMODE bits are 0, this bit is ignored.	

CNTRDSEL	Sensor Counter Register Read Count Select
0	Transitions to the next state after single reading.
1	Transitions to the next state after reading twice.

Figure 27 - 15 Format of CTSU Calibration Registers L and H (CTS2BGR0, CTS2BGR1) (3/3)

TSOC	Calibration Setting 2	
0	Electrostatic capacitance measurement mode	
1	Calibration setting 2	
The TSOC bit is used to set calibration of the CTSU. When measuring electrostatic capacitance, set this bit to 0.		

SUCLKEN	SUCLK Enable	
0	SUCLK is disabled.	
1	SUCLK is enabled.	

CLKSEL0[1:0]		Observation Clock Select 0
0	0	Low level is always output.
0	1	Sensor ICO clock (divided by 8)
1	0	Setting prohibited
1	1	SUCLK (divided by 8)
The CLKSEL0[1:0] bits select the clock to be observed from the three clocks that the CTSU hardware macro outputs.		

DRV	Calibration Setting 1	
0	Electrostatic capacitance measurement mode	
1	Calibration setting 1	
The DRV bit is used to set calibration of the CTSU. When measuring electrostatic capacitance, set this bit to 0.		

TSOD	TSM Pin Fixed Output	
0	Electrostatic capacitance measurement mode	
1	TSM pin fixed output (high- or low-level output)	
The TSOD bit is used for calibration of the CTSU. When this bit is set to 1, the level specified by the CTSUDBG0.IOC bit is output from the TSM pin.		

27.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)

Figure 27 - 16 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH
(CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) (1/2)

Address: F052CH, F052DH (CTSUSUCLK0), F052EH, F052FH (CTSUSUCLK1),
F0530H, F0531H (CTSUSUCLK2), F0532H, F0533H (CTSUSUCLK3)

After reset: 0000H, 0000H, 0000H, 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK1	SUMULT1[7:0]							
	7	6	5	4	3	2	1	0
	SUADJ1[7:0]							
Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK0	SUMULT0[7:0]							
	7	6	5	4	3	2	1	0
	SUADJ0[7:0]							
Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK3	SUMULT3[7:0]							
	7	6	5	4	3	2	1	0
	SUADJ3[7:0]							
Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK2	SUMULT2[7:0]							
	7	6	5	4	3	2	1	0
	SUADJ2[7:0]							
SUMULTx[7:0]		SUCLK Multiplication Rate Setting						
The SUMULTx[7:0] bits set the multiplication rate of STCLK (assuming 0.5 MHz (divided fCLK)) to generate SUCLK.								
STCLK is compared with SUCLK divided by this setting. Based on the comparison result, the SUADJx[7:0] bits are updated. The target clock frequency is 32 MHz to 80 MHz.								
0	0	0	0	0	0	0	0	× 1
:	:	:	:	:	:	:	:	:
0	0	1	1	1	1	1	1	× 64
:	:	:	:	:	:	:	:	:
0	1	1	1	1	1	1	1	× 128
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	× 256

Figure 27 - 16 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH
(CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) (2/2)

SUADJx[7:0]	SUCLK Frequency Adjustment
The SUADJx[7:0] bits set the initial value of the SUCLK frequency. The drift is adjusted and the SUADJx[7:0] value is updated by the clock recovery function. The output frequency varies from the set value in each MCU. The SUCLK frequency is adjusted based on the register set value as an initial value and the register value is updated by the clock recovery control.	

27.2.13 CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1)

The CTSUTRIM0 register can be accessed in 8-bit units (DACTRIM as upper part and RTRIM as lower part).

The CTSUTRIM1 register can be accessed in 8-bit units (TRESULT4 as upper part and SUADJD as lower part).

Figure 27 - 17 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1) (1/2)

Address: F0600H, F0601H (CTSUTRIM0), F0602H, F0603H (CTSUTRIM1)

After reset: 0000H, 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM1	TRESULT4[7:0]							
	7	6	5	4	3	2	1	0
	SUADJD[7:0]							
Symbol	15	14	13	12	11	10	9	8
CTSUTRIM0	DACTRIM[7:0]							
	7	6	5	4	3	2	1	0
	RTRIM[7:0]							
TRESULT4[7:0]		Test Result 4						
The TRESULT4[7:0] bits store the coefficient of variation for the 120-kΩ reference load resistance. These bits are the initial value set at the factory. Do not modify these bits.								
SUADJD[7:0]		SUCLK Frequency Adjustment						
The SUADJD[7:0] bits hold the initial value for generating approx. 64 MHz, which is set at the factory. When FCMODE is 0, this set value is input to the digital oscillator. Do not modify this initial value set at the factory.								
DACTRIM[7:0]		Offset Current DAC Upper/Lower Matching Variation Adjustment						
The DACTRIM[7:0] bits adjust the upper/lower matching variation (coefficient of the lower current source) of the offset current DAC. These bits hold the initial value set at the factory. Do not modify this value.								
0	0	0	0	0	0	0	0	× 0.0
:	:	:	:	:	:	:	:	:
0	1	1	0	0	0	0	0	× 0.875
:	:	:	:	:	:	:	:	:
1	0	0	0	0	0	0	0	× 1.0
:	:	:	:	:	:	:	:	:
1	0	0	1	0	0	0	0	× 1.125
:	:	:	:	:	:	:	:	:
1	0	1	0	0	0	1	1	× 1.273
Other than above: Setting prohibited								

Figure 27 - 17 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1) (2/2)

RTRIM[7:0]								Reference Resistance Adjustment
The RTRIM[7:0] bits adjust the reference resistance value. These bits hold the initial value set at the factory. Do not modify this value.								
RTRIM[7:0]								Resistance value
0	0	0	0	0	0	0	0	Low
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	1	High

27.2.14 CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3)

The CTSUTRIM2 register can be accessed in 8-bit units (TRESULT1 as upper part and TRESULT0 as lower part).
The CTSUTRIM3 register can be accessed in 8-bit units (TRESULT3 as upper part and TRESULT2 as lower part).

Figure 27 - 18 Format of CTSU Trimming Registers BL and BH (CTSUTRIM2, CTSUTRIM3)

Address: F0604H, F0605H (CTSUTRIM2), F0606H, F0607H (CTSUTRIM3)
After reset: 0000H, 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM3	TRESULT3[7:0]							
	7	6	5	4	3	2	1	0
	TRESULT2[7:0]							
Symbol	15	14	13	12	11	10	9	8
CTSUTRIM2	TRESULT1[7:0]							
	7	6	5	4	3	2	1	0
	TRESULT0[7:0]							
TRESULT3[7:0]			Test Result 3					
The TRESULT3[7:0] bits store the coefficient of variation for the 60-kΩ reference load resistance. These bits are the initial value set at the factory. Do not modify these bits.								
TRESULT2[7:0]			Test Result 2					
The TRESULT2[7:0] bits store the coefficient of variation for the 30-kΩ reference load resistance. These bits are the initial value set at the factory. Do not modify these bits.								
TRESULT1[7:0]			Test Result 1					
The TRESULT1[7:0] bits store the coefficient of variation for the 15-kΩ reference load resistance. These bits are the initial value set at the factory. Do not modify these bits.								
TRESULT0[7:0]			Test Result 0					
The TRESULT0[7:0] bits store the coefficient of variation for the 7.5-kΩ reference load resistance. These bits are the initial value set at the factory. Do not modify these bits.								

27.3 Usage Notes of the Capacitive Sensing Unit

(1) Evaluation of detection by the capacitive sensing unit (CTS2La)

In the final stage of development, the user must judge whether or not detection by the touch sensor is reliable in the systems for customers. To do so, the user must run the systems in nearly completed products and use the QE for Capacitive Touch tool (development assistance tool for capacitive touch sensors) to thoroughly evaluate operation by monitoring the measurement of electrostatic capacitance.

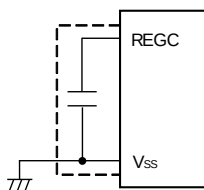
If obtaining the desired results of detection is not possible, adjust the CapTouch parameters (mainly the touch threshold) through QE for Capacitive Touch and re-evaluate the system.

Section 28 Regulator

28.1 Overview

The RL78/G22 incorporates a circuit for constant voltage operation. To stabilize the output voltage from the regulator, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μ F). Use a capacitor with good characteristics, since it is for stabilizing the internal voltage.

The REGC pin can be used as a reference voltage for an external circuit. An external circuit for connection to the REGC pin for this purpose must have an input impedance of at least 1.5 M Ω . The voltage on the REGC pin is in the range from 1.38 to 1.60 V, and the typical value is 1.5 V.



Caution The length of the wiring within the broken lines in the above figure should be as short as possible.

The regulator outputs a 1.5-V voltage.

Section 29 Option Bytes

29.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G22 form an option byte area.

Option bytes consist of user option bytes (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.

For the bits to which no function is allocated, do not change their initial values.

To use the boot swap operation during self-programming, 000C0H to 000C3H are replaced by 020C0H to 020C3H.

Therefore, set the same values as 000C0H to 000C3H to 020C0H to 020C3H.

Caution The option bytes should always be set regardless of whether each function is used.

29.1.1 User option bytes (000C0H to 000C2H or 020C0H to 020C2H)

(1) 000C0H or 020C0H

- Setting of watchdog timer operation
 - Enabling or disabling of counter operation
 - Enabling or stopping of counter operation in the HALT or STOP mode
- Setting of overflow time of watchdog timer
- Setting of window open period of watchdog timer
- Setting of interval interrupt of watchdog timer
 - Interval interrupt is used or not used

Caution When boot swapping is to be used, 000C0H is replaced with 020C0H. Therefore, set the same value as the setting in 000C0H in 020C0H.

(2) 000C1H or 020C1H

- Setting of LVD0 operation mode
 - Reset mode
 - Interrupt mode
 - LVD0 off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
- Setting of LVD0 detection level (VLVD0)

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 020C2H).

Caution 2. When boot swapping is to be used, 000C1H is replaced with 020C1H. Therefore, set the same value as the setting in 000C1H in 020C1H.

(3) 000C2H or 020C2H

- Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.

- LS (low-speed main) mode
- HS (high-speed main) mode
- LP (low-power main) mode

- Setting of the frequency of the high-speed on-chip oscillator
 - Select from 1 MHz to 32 MHz.

Caution When boot swapping is to be used, 000C2H is replaced with 020C2H. Therefore, set the same value as the setting in 000C2H in 020C2H.

29.1.2 On-chip debug option byte (000C3H or 020C3H)

- Control of on-chip debug operation
 - On-chip debug operation is disabled or enabled.
- Handling of data of flash memory in case of failure in on-chip debug security ID authentication
 - Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution When boot swapping is to be used, 000C3H is replaced with 020C3H. Therefore, set the same value as the setting in 000C3H in 020C3H.

29.2 Format of User Option Bytes

Figure 29 - 1 Format of User Option Byte (000C0H or 020C0H)

Address: 000C0H or 020C0H^{Note 1}

Symbol	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
	WDTINT	Use of interval interrupt of watchdog timer						
	0	Interval interrupt is not used.						
	1	Interval interrupt is generated when 75% of the overflow time + 1/2 f _{IL} is reached.						
	WINDOW1	WINDOW0	Watchdog timer window open period ^{Note 2}					
	0	0	Setting prohibited					
	0	1	50%					
	1	0	Setting prohibited					
	1	1	100%					
	WDTON	Operation control of watchdog timer counter						
	0	Counter operation disabled (counting stopped after reset)						
	1	Counter operation enabled (counting started after reset)						
	WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f _{IL} = 37.683 kHz (max.))				
	0	0	0	2 ⁷ /f _{IL} (3.39 ms)				
	0	0	1	2 ⁸ /f _{IL} (6.79 ms)				
	0	1	0	2 ⁹ /f _{IL} (13.58 ms)				
	0	1	1	2 ¹⁰ /f _{IL} (27.17 ms)				
	1	0	0	2 ¹² /f _{IL} (108.69 ms)				
	1	0	1	2 ¹⁴ /f _{IL} (434.78 ms)				
	1	1	0	2 ¹⁵ /f _{IL} (869.56 ms)				
	1	1	1	2 ¹⁷ /f _{IL} (3478.26 ms)				
	WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)						
	0	Counter operation stopped in HALT/STOP mode ^{Note 2}						
	1	Counter operation enabled in HALT/STOP mode						

Note 1. When boot swapping is to be used, 000C0H is replaced with 020C0H. Therefore, set the same value as the setting in 000C0H in 020C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless of the value of the WINDOW1 and WINDOW0 bits.

Figure 29 - 2 Format of User Option Byte (000C1H or 020C1H) (1/3)

Address: 000C1H or 020C1H **Note**

Symbol	7	6	5	4	3	2	1	0
	LVD0EN	LVD0SEL	1	1	1	LVD0V2	LVD0V1	LVD0V0

• LVD0 setting (reset mode)

Detection Voltage		Option Byte Setting Value				
VLVD0		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	1	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

Note When boot swapping is to be used, 000C1H is replaced with 020C1H. Therefore, set the same value as the setting in 000C1H in 020C1H.

Caution Be sure to set bits 5 to 3 to 1.

Remark 1. For details on the LVD0 circuit, see **Section 23 Voltage Detector (LVD)**.

Remark 2. The detection voltage is a typical value. For details, see **34.6.4 Characteristics of the LVD circuit**.

Figure 29 - 2 Format of User Option Byte (000C1H or 020C1H) (2/3)

• LVD0 setting (interrupt mode)

Detection Voltage		Option Byte Setting Value				
VLVD0		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	0	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

Note When boot swapping is to be used, 000C1H is replaced with 020C1H. Therefore, set the same value as the setting in 000C1H in 020C1H.

Caution Be sure to set bits 5 to 3 to 1.

Remark 1. For details on the LVD0 circuit, see **Section 23 Voltage Detector (LVD)**.

Remark 2. The detection voltage is a typical value. For details, see **34.6.4 Characteristics of the LVD circuit**.

Figure 29 - 2 Format of User Option Byte (000C1H or 020C1H) (3/3)

- LVD0 off setting (external reset input from the RESET pin is used)

Detection Voltage		Option Byte Setting Value				
VLVD0		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
—	—	0	×	0	1	0
—		Settings other than the above are prohibited.				

Note When boot swapping is to be used, 000C1H is replaced with 020C1H. Therefore, set the same value as the setting in 000C1H in 020C1H.

Caution 1. Be sure to set bits 5 to 3 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 020C2H).

Remark 1. ×: Don't care.

Remark 2. For details on the LVD0 circuit, see Section 23 Voltage Detector (LVD).

Remark 3. The detection voltage is a typical value. For details, see 34.6.4 Characteristics of the LVD circuit.

Figure 29 - 3 Format of User Option Byte (000C2H or 020C2H)

Address: 000C2H or 020C2H^{Note}

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

Value of the Option Byte (000C2H)		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	1	LP (low-power main) mode	1 MHz to 2 MHz (Rewriting of flash memory is not possible.)	1.6 V to 5.5 V
1	0	LS (low-speed main) mode	1 MHz to 4 MHz (Rewriting of flash memory is not possible.)	1.6 V to 5.5 V
			1 MHz to 24 MHz	1.8 V to 5.5 V
1	1	HS (high-speed main) mode	1 MHz to 4 MHz (Rewriting of flash memory is not possible.)	1.6 V to 5.5 V
			1 MHz to 32 MHz	1.8 V to 5.5 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the High-speed On-chip Oscillator Clock
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Note When boot swapping is to be used, 000C2H is replaced with 020C2H. Therefore, set the same value as the setting in 000C2H in 020C2H.

Caution 1. Be sure to set bits 5 and 4 to 10B.

Caution 2. The operating frequency range and operating voltage range vary depending on the flash operation mode. For details, see 34.4 AC Characteristics.

29.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 29 - 4 Format of On-chip Debug Option Byte (000C3H or 020C3H)

Address: 000C3H or 020C3H ^{Note}

Symbol	7	6	5	4	3	2	1	0
	OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debugging.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Note When boot swapping is to be used, 000C3H is replaced with 020C3H. Therefore, set the same value as the setting in 000C3H in 020C3H.

Caution Only bits 7 and 0 (OCDENSET and OCDERSD) are specifiable.
Be sure to set bits 6 to 1 to 000010B.

Remark The use of on-chip debugging changes the values of bits 3 to 1. Accordingly, the values after such settings are made become undefined. However, note that when setting the option byte, be sure to set bits 3 to 1 to their default values (0, 1, and 0).

29.4 Setting of Option Bytes

The user option bytes and on-chip debug option byte can also be set by using linker options instead of statements in the source code. In such cases, the settings made by using linker options are given priority over the statements in the source code as shown below.

An example of the statements in relation to the option byte settings in software is shown below.

.CSEG	OPT_BYTE	
.DB	0x36	; Does not use the interval interrupt of the watchdog timer. ; Enables watchdog timer operation. ; The window open period of the watchdog timer is 50%. ; The overflow time of the watchdog timer is $2^{10}/f_{IL}$. ; Stops watchdog timer operation during HALT/STOP mode.
.DB	0xBF	; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges. ; Select the interrupt mode as the LVD0 operation mode.
.DB	0x6D	; Select the LP (low-power main) mode as the flash operation mode and 1 MHz ; as the frequency of the high-speed on-chip oscillator clock.
.DB	0x85	; Enables on-chip debug operation, does not erase flash memory data when ; security ID authentication fails.

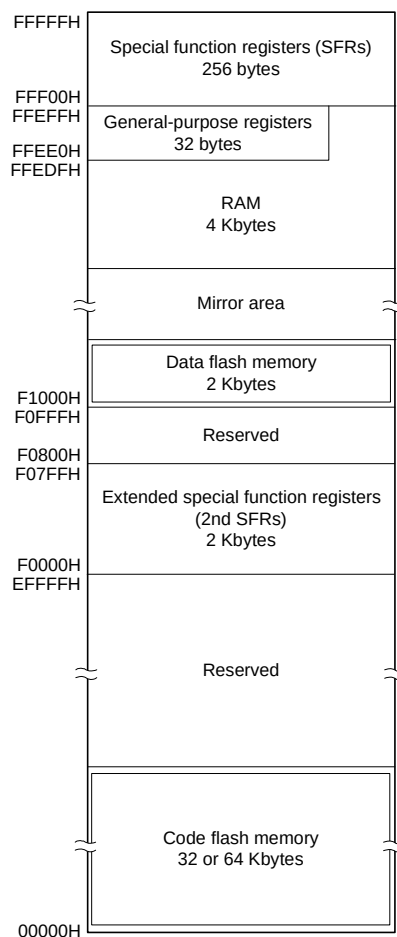
When boot swapping is to be used during self-programming, 000C0H to 000C3H are replaced with 020C0H to 020C3H. Therefore, set the same values as the settings in 000C0H to 000C3H in 020C0H to 020C3H as shown below.

OPT2	.CSEG	AT	0x020C0
	.DB		0x36 ; Does not use the interval interrupt of the watchdog timer. ; Enables watchdog timer operation. ; The window open period of the watchdog timer is 50%. ; The overflow time of the watchdog timer is $2^{10}/f_{IL}$. ; Stops watchdog timer operation during HALT/STOP mode.
	.DB		0xBF ; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges. ; Select the interrupt mode as the LVD0 operation mode.
	.DB		0x6D ; Select the LP (low-power main) mode as the flash operation mode and 1 MHz ; as the frequency of the high-speed on-chip oscillator clock.
	.DB		0x85 ; Enables on-chip debug operation, does not erase flash memory data when ; security ID authentication fails.

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify 020C0H to 020C3H for use as the option bytes in the case of boot swapping, use the relocation attribute AT to specify absolute addresses.

Section 30 Flash Memory

The RL78 microcontroller incorporates flash memory that allows the writing, erasure, and overwriting of programs. The flash memory consists of a code flash memory area, from which programs can be executed, a data flash memory area, which is suitable for storing data, and an extra area, which is for storing flash memory operation and security settings.



The following methods for programming the flash memory are available.

The contents of code flash memory can be rewritten through serial programming by using a flash memory programmer or other external device (via UART communications), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **30.1.**)
Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (that Incorporates UART) (see **30.2.**)
Data can be written to the flash memory on-board through UART communications with an external device (a microcontroller or ASIC).
- Self-Programming (see **30.6.**)
The user application can execute self-programming of the code flash memory.

Caution When rewriting the flash memory, stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (foco). Do not change the flash operating mode select register (FLMODE).

The data flash memory can be rewritten to by using self-programming during user program execution (background operation). For access and writing to the data flash memory, see **30.6 Self-Programming** and **30.10 Data Flash Memory**.

The code flash memory and the data flash memory have a function to protect them from rewriting. For details, see **24.3.3 Flash memory guard function**.

30.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmers can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP6
- E2 or E2 Lite on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. Mount a connector for the dedicated flash memory programmer on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter before the RL78 microcontroller is mounted on the target system.

Table 30 - 1 Wiring between RL78/G22 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.						
					16-Pin	20-Pin	24-Pin	25-Pin	30-Pin	32-Pin	36-Pin
					QFN (3.0 × 3.0)	LSSOP (4.4 × 6.5)	HWQFN (4 × 4)	WFLGA (3 × 3)	LSSOP (300 mil)	HWQFN (5 × 5) LGFP (7 × 7)	WFLGA (4 × 4)
Signal Name		I/O	Pin Function								
PG-FP6	E2 or E2 Lite On-Chip Debugging Emulator										
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	16	3	23	A5	5	1	F6
SI/RxD	—	I/O									
—	RESET	Output	Reset signal	RESET	7	4	24	B5	6	2	E5
/RESET	—	Output									
VCC	VDD	I/O	VDD voltage generation/ power monitoring	VDD	5	10	6	B3	12	8	B6
GND		—	Ground	VSS	4	9	5	B2	11	7	C5
				REGC Note	3	8	4	A2	10	6	D5
FLMD1	EMVDD	—	Driving power for TOOL0 pin	VDD	5	10	6	B3	12	8	B6

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.		
					40-Pin	44-Pin	48-Pin
Signal Name		I/O	Pin Function		HWQFN (6 × 6)	LQFP (10 × 10)	LFQFP (7 × 7) HWQFN (7 × 7)
PG-FP6	E2 or E2 Lite On-Chip Debugging Emulator			TOOL0	—	1	2
—	—	I/O	Transmit/ receive signal	TOOL0/ P40	—	—	—
SI/RxD	—	I/O		—	—	—	—
—	RESET	Output	Reset signal	RESET	2	3	40
/RESET	—	Output		—	—	—	—
Vcc	VDD	I/O	VDD voltage generation/ power monitoring	VDD	10	11	48
GND		—	Ground	Vss	9	10	47
				REGC Note	8	9	46
FLMD1	EMVDD	—	Driving power for the TOOL0 pin	VDD	10	11	48

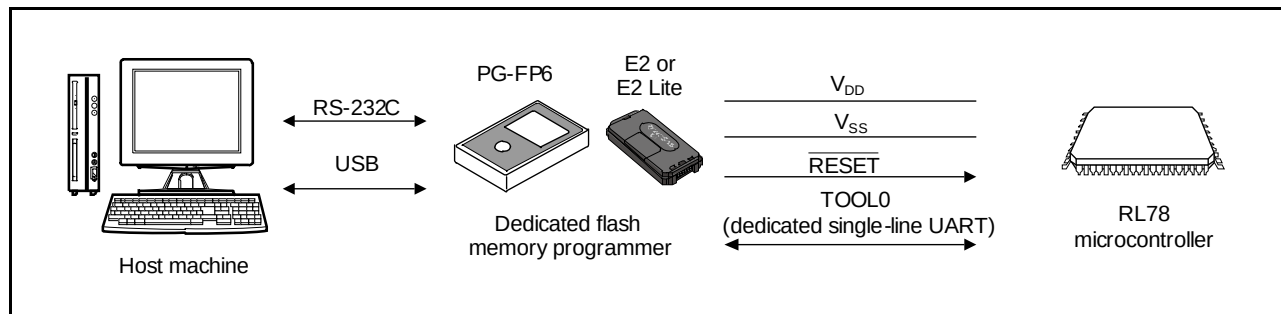
Note Connect the REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open-circuit when using the flash memory programmer for flash memory programming.

30.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 1 Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

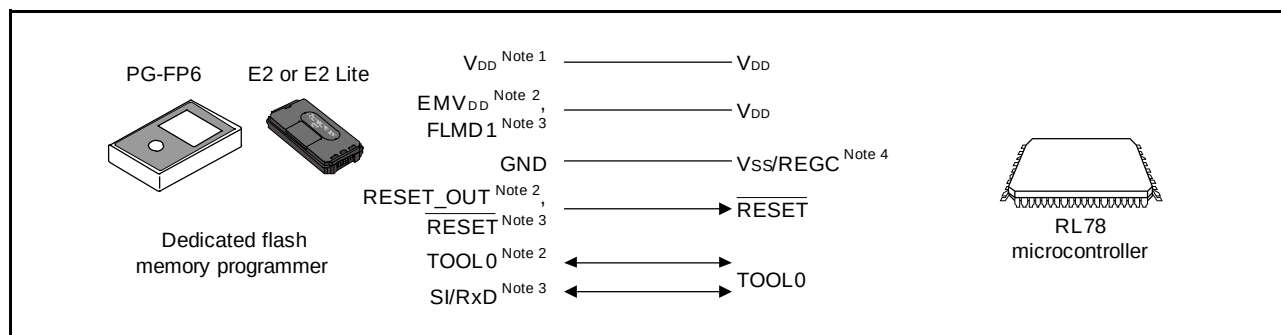
For the interface between the dedicated flash memory programmer and RL78 microcontroller, the TOOL0 pin of the RL78 microcontroller is used for writing and erasure via the dedicated single-line UART.

30.1.2 Communications mode

The TOOL0 pin of the RL78 microcontroller is used for communications between the dedicated flash memory programmer and RL78 microcontroller through serial communications by using the dedicated single-line UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 30 - 2 Communications with Dedicated Flash Memory Programmer



- Note 1.** The signal name for the PG-FP6 is Vcc.
- Note 2.** When using the E2 or E2 Lite on-chip debugging emulator.
- Note 3.** When using the PG-FP6.
- Note 4.** Connect the REGC pin to ground via a capacitor (0.47 to 1 μ F).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual for the PG-FP6 or, E2 or E2 Lite on-chip debugging emulator for details.

Table 30 - 2 Pin Connection

Dedicated Flash Memory Programmer			RL78 Microcontroller	
Signal Name		I/O	Pin Function	Pin Name Note 1
PG-FP6	E2 or E2 Lite On-Chip Debugging Emulator			
VCC	VDD	I/O	VDD voltage generation/power monitoring	VDD
GND		—	Ground	Vss, REGC Note 2
FLMD1	EMVDD	—	Driving power for the TOOL0 pin	VDD
$\overline{\text{RESET}}$	—	Output	Reset signal	$\overline{\text{RESET}}$
—	RESET_OUT	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Pins to be connected differ with the product. For details, see **Table 30 - 1**.

Note 2. Connect the REGC pin to ground via a capacitor (0.47 to 1 μF).

30.2 Serial Programming Using External Device (that Incorporates UART)

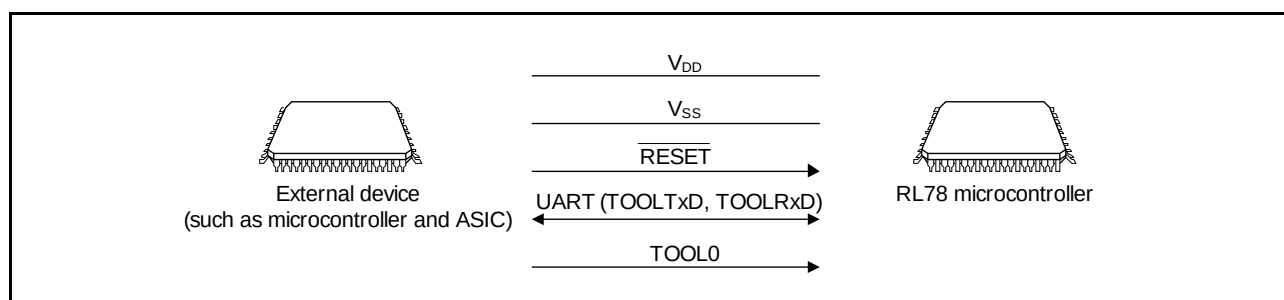
On-board data writing to the flash memory of the RL78 microcontroller is possible by using an external device (a microcontroller or ASIC) that is connected to the microcontroller through a UART.

On the development of flash memory programmer by user, refer to the application note, RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756).

30.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 3 Environment for Writing Program to Flash Memory

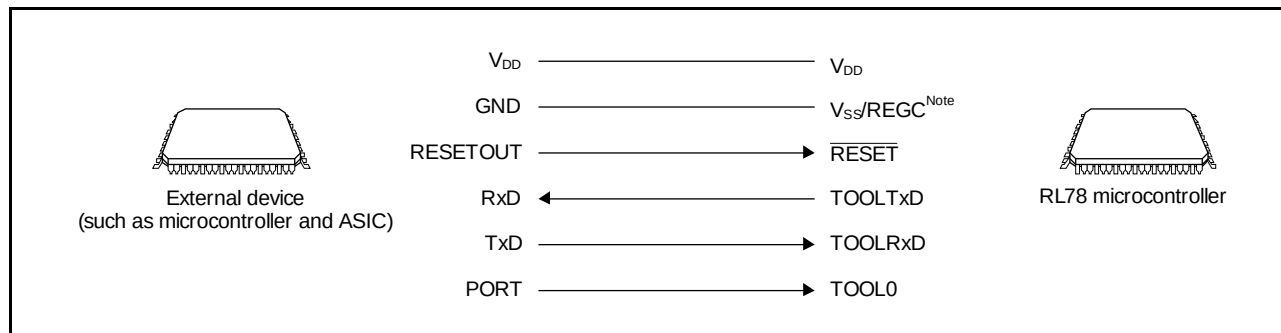


30.2.2 Communications mode

The TOOLTxD and TOOLRxD pins of the RL78 microcontroller are used for communications between the external device and RL78 microcontroller through serial communications by using the dedicated UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 30 - 4 Communications with External Device



Note Connect the REGC pin to ground via a capacitor (0.47 to 1 μ F).

The external device generates the following signals for the RL78 microcontroller.

Table 30 - 3 Pin Connection

External Device			RL78 Microcontroller
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD
GND	—	Ground	Vss, REGC ^{Note}
RESETOUT	Output	Reset signal output	RESET
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect the REGC pin to ground via a capacitor (0.47 to 1 μ F).

30.3 Handling of Pins on the Board

To write to the flash memory on board by using a flash memory programmer, mount a connector for the dedicated flash memory programmer on the target system. In addition, include a function for mode switching from normal operation to flash memory programming in the board design.

Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states do not suit the operation of external devices, apply appropriate handling to the pins.

Remark For details on flash memory programming mode, refer to **30.4.2 Flash memory programming mode**.

30.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1-k Ω pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin: Input of the low level is prohibited for t_{HD} period after external reset release. However, when this pin is used via a pull-down resistor, use a resistor with a value of 500 k Ω or more.

When used as an output pin: When this pin is used via a pull-down resistor, use a resistor with a value of 500 k Ω or more.

Remark 1. t_{HD} : This is the time over which the TOOL0 pin must be kept at the low level following the end of the external and internal resets for setting of the flash memory programming mode. See **34.10 Timing of Entry to Flash Memory Programming Modes**.

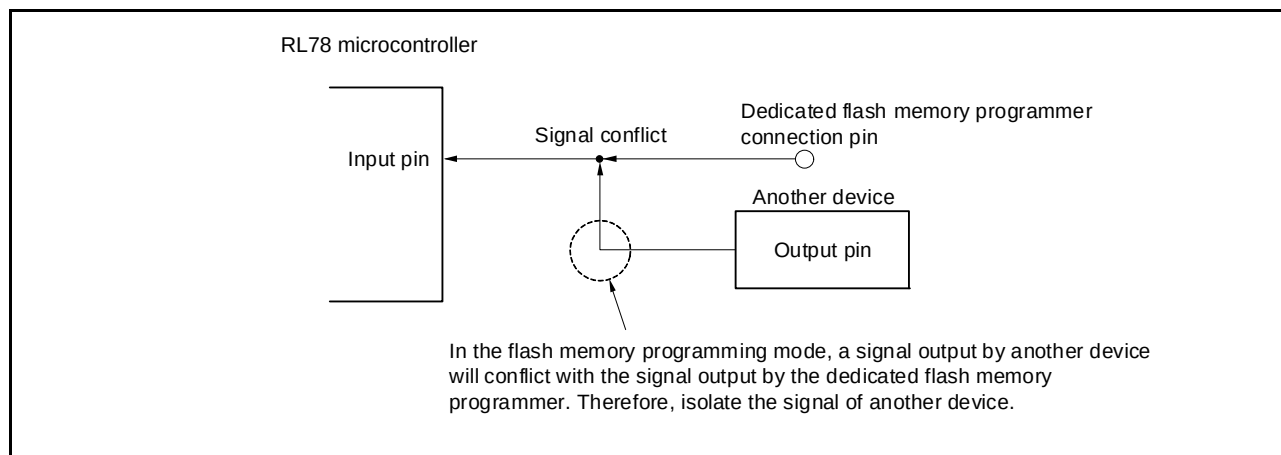
Remark 2. The SAU and IICA pins are not used for communications between the RL78 microcontroller and dedicated flash memory programmer, because the single-line UART (TOOL0 pin) is used.

30.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 30 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



30.3.3 Port pins

Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states of the port pins do not suit the operation of external devices that are connected to individual port pins, apply appropriate handling to the pins such as connection to VDD or VSS via resistors.

30.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μ F) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

30.3.5 Power supply

To use the supply voltage output of the flash memory programmer, connect the VDD pin to VDD^{Note} of the flash memory programmer, and the VSS pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect the power supply in the same manner as during normal operation.

However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the VDD and VSS pins to VDD^{Note} and GND of the flash memory programmer to use the power monitor function by the flash memory programmer.

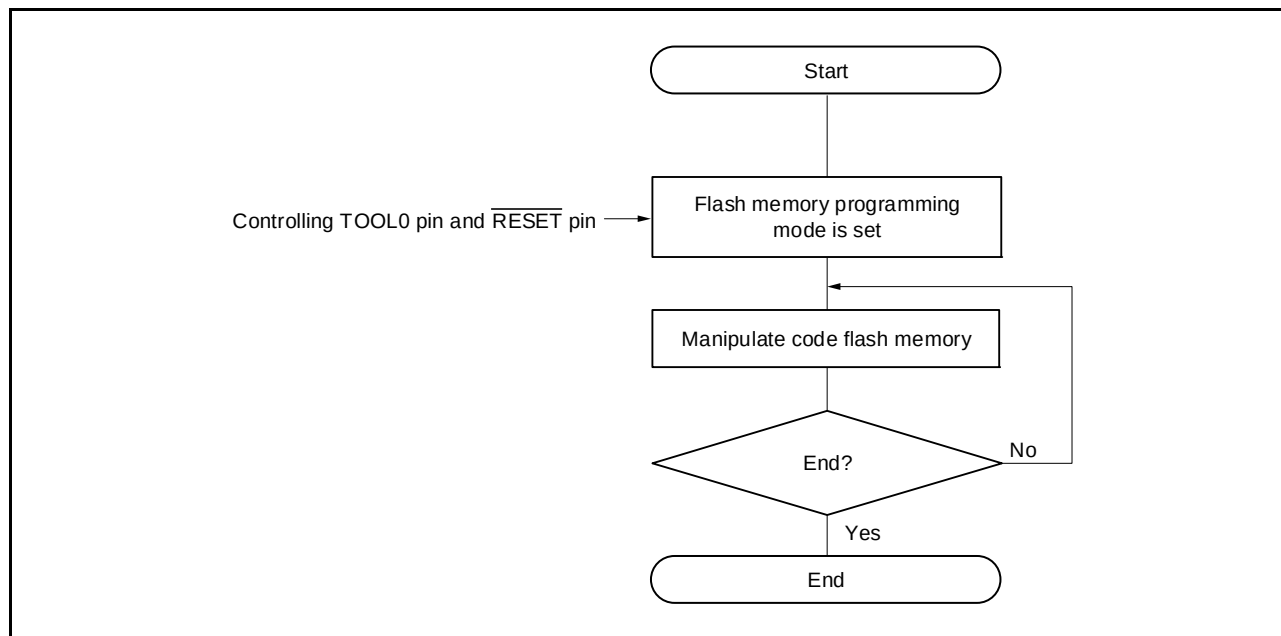
Note The signal name for the PG-FP6 is VCC.

30.4 Programming Method

30.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 30 - 6 Code Flash Memory Manipulation Procedure



30.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. The following describes how to enter the flash memory programming mode.

<For serial programming by using the dedicated flash memory programmer>

Connect the RL78 microcontroller to a dedicated flash memory programmer. Starting communications with the dedicated flash memory programmer automatically places the RL78 LSI chip in the flash memory programming mode.

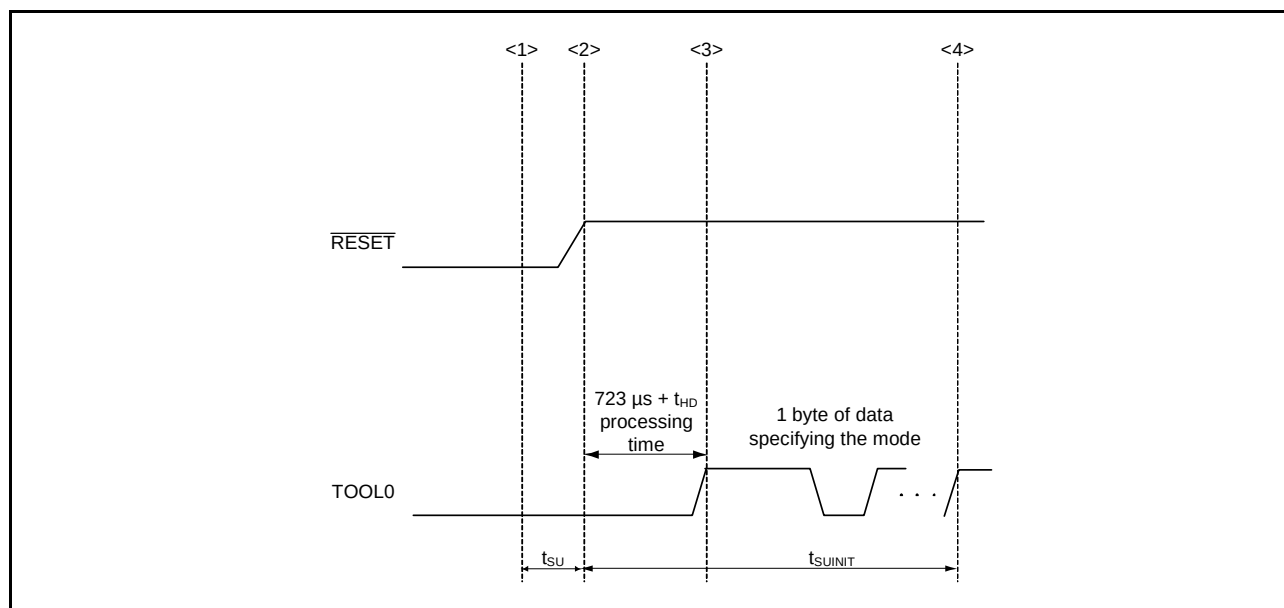
<For serial programming by using an external device (UART communications)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 30 - 4**). After that, enter flash memory programming mode according to steps <1> to <4> shown in **Figure 30 - 7**. For details, refer to the application note, RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756).

Table 30 - 4 Relationship between the TOOL0 Pin and Operating Mode after Release from the Reset State

TOOL0	Operating Mode
V _{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 30 - 7 Entry to Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.

<2> The external reset is de-asserted (release from a POR or LVD reset must precede this).

<3> The TOOL0 pin is set to the high level.

<4> The bit rate setting is based on the UART reception.

Remark t_{SUINIT}: The section is the up to 100 ms from the end of the external reset within which specifying the initial communications settings must be finished.

t_{SU}: This is the time from the TOOL0 pin being placed at the low level until the end of the external reset.

t_{HD}: This is the time over which the TOOL0 pin must be kept at the low level following the end of the external and internal resets (excluding the processing time of the firmware to control the flash memory).

For details, see **34.10 Timing of Entry to Flash Memory Programming Modes**.

30.4.3 Selecting communications mode

The communications modes of the RL78 microcontroller are shown in the table below.

Table 30 - 5 Communications Modes

Communications Mode	Standard Setting ^{Note 1}				Pins Used
	Port	Speed ^{Note 2}	Frequency	Multiply Rate	
Single-line UART (when a flash memory programmer is used, or when an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for standard settings on GUI of the flash memory programmer.

Note 2. As factors other than bit rate errors, such as dullness of the signal waveform, may also affect UART communications, conduct extensive evaluation with a selected bit rate before attempting to use it.

30.4.4 Communications commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 30 - 6**.

The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the application note, RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756).

Table 30 - 6 Flash Memory Control Commands

Classification	Command Name	Function
Verification	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erasure	Block Erase	Erases a specified area in the flash memory.
Blank checking	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Writing	Programming	Writes data to a specified area in the flash memory ^{Note} .
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the device name, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Releases the setting to prohibit writing.
Others	Reset	Used to detect the state of synchronization during communications.
	Baud Rate Set	Sets the bit rate when UART communications mode is selected.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

The product information (such as device name and firmware version) can be obtained by executing the Silicon Signature command.

Tables 30 - 7 and **30 - 8** show the signature data list and an example of signature data, respectively.

Table 30 - 7 Signature Data List

Field Name	Description	Number of Bytes of Transmit Data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area end address	End address of the code flash memory area (Sent from the lower-order byte of the address. Example: 00000H to 0FFFFH (64 Kbytes) → FFH, FFH, 00H)	3 bytes
Data flash memory area end address	End address of the data flash memory area (Sent from the lower-order byte of the address. Example: F1000H to F17FFH (2 Kbytes) → FFH, 17H, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from the byte equivalent to the highest-order digit of the version number. Example: Ver. 1.02 → 01H, 00H, 02H)	3 bytes

Table 30 - 8 Example of Signature Data

Field Name	Description	Number of Bytes of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol C	3 bytes	10 00 0A
Device name	R7F102GGE	10 bytes	52 = "R" 37 = "7" 46 = "F" 31 = "1" 30 = "0" 32 = "2" 47 = "G" 47 = "G" 45 = "E" 20 = " "
Code flash memory area end address	Code flash memory area 00000H to 0FFFFH (64 Kbytes)	3 bytes	FF FF 00
Data flash memory area end address	Data flash memory area F1000H to F17FFH (2 Kbytes)	3 bytes	FF 17 0F
Firmware version	Ver. 1.02	3 bytes	01 00 02

30.5 Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values)

The following shows the processing times for each command (reference value) when the PG-FP6 is used as a dedicated flash memory programmer.

Table 30 - 9 Processing Times for Commands When the PG-FP6 Is in Use (Reference Values)

PG-FP6 Command	Code Flash Memory	
	32 Kbytes	64 Kbytes
Erasure	T.B.D.	T.B.D.
Writing	T.B.D.	T.B.D.
Verification	T.B.D.	T.B.D.
Writing after erasure	T.B.D.	T.B.D.

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.
Port: TOOL0 (single-line UART)
Speed: 1,000,000 bps

30.6 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory, it can be used to update the program in the field.

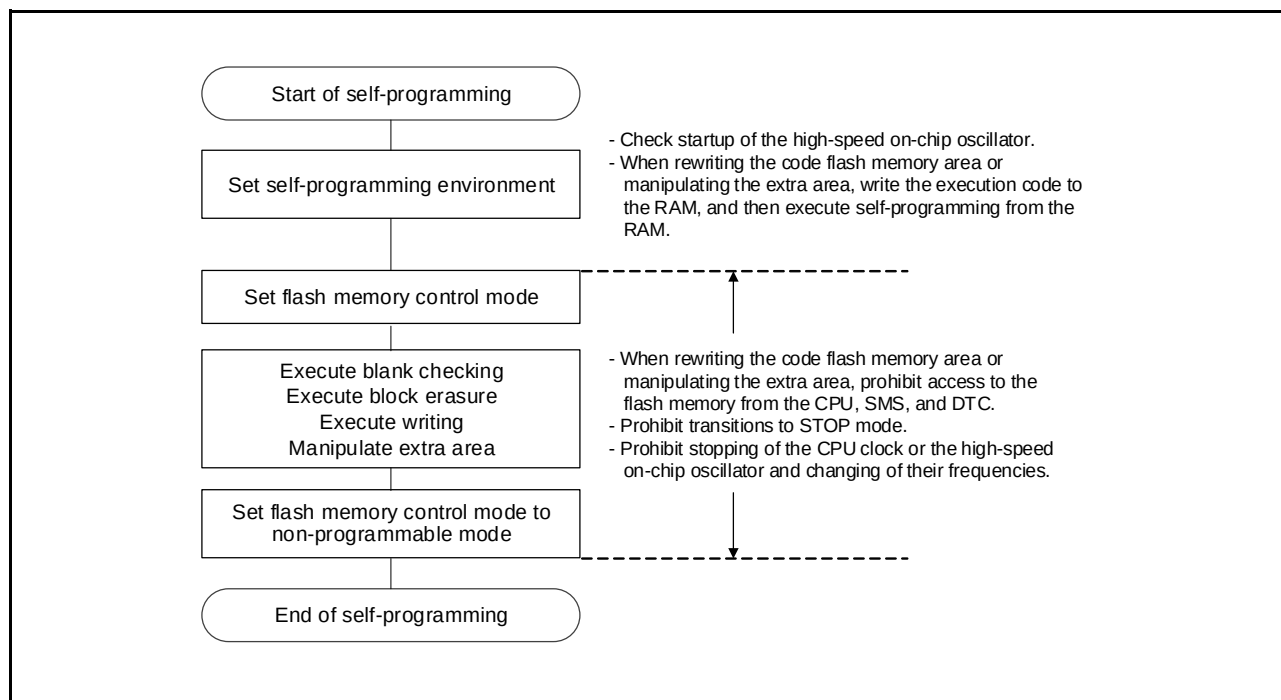
- Caution 1.** The self-programming function cannot be used when the CPU operates with the subsystem clock (f_{SUB}).
- Caution 2.** To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute self-programming in the state where the IE flag is cleared to 0 by the DI instruction. To enable an interrupt, clear the interrupt mask flag for an interrupt to be accepted to 0 in the state where the IE flag is set to 1 by the EI instruction, and then execute self-programming.
- Caution 3.** The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be re-executed after 5 μs have elapsed. Stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (foco).
- Caution 4.** Do not change the flash operating mode select register (FLMODE) while the flash memory is being rewritten.

30.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the flash memory by using self-programming.

For details on registers for use in self-programming, see **30.6.2 Registers for controlling the flash memory**.

Figure 30 - 8 Flow of Self-Programming (Rewriting the Flash Memory)



30.6.2 Registers for controlling the flash memory

The following registers are used to control the flash memory.

- Flash address pointer registers H and L (FLAPH, FLAPL)
- Flash end address pointer registers H and L (FLSEDH, FLSEDL)
- Flash write buffer registers H and L (FLWH, FLWL)
- Flash protect command register (PFCMD)
- Flash status register (PFS)
- Flash programming mode control register (FLPMC)
- Flash area selection register (FLARS)
- Flash memory sequencer initial setting register (FSSET)
- Flash memory sequencer control register (FSSQ)
- Flash extra area sequencer control register (FSSE)
- Flash registers initialization register (FLRST)
- Flash memory sequencer status registers H and L (FSASTH, FSASTL)
- Flash security flag monitoring register (FLSEC)
- Flash FSW monitoring register E (FLFSWE)
- Flash FSW monitoring register S (FLFSWS)
- Data flash control register (DFLCTL)
- Interrupt vector jump enable register (VECTCTRL)
- Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

30.6.2.1 Flash address pointer registers H and L (FLAPH, FLAPL)

The FLAPH and FLAPL registers specify the address where programming the flash memory is to start.
The FLAPH and FLAPL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively.
The values of the FLAPH and FLAPL registers are 00H and 0000H, respectively, under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 30 - 9 Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL)

Address: F02C4H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLAPH	0	0	0	0	FLAP19	FLAP18	FLAP17	FLAP16

Address: F02C2H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLAPL	FLAP15	FLAP14	FLAP13	FLAP12	FLAP11	FLAP10	FLAP9	FLAP8

	7	6	5	4	3	2	1	0
	FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0

Caution 1. The FLAPH and FLAPL registers can be rewritten under either of the following conditions.

- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
- The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (the SQEND and ESQEND bits in FSASTH are 0).

30.6.2.2 Flash end address pointer registers H and L (FLSEDH, FLSEDL)

The FLSEDH and FLSEDL registers specify the address where programming of the flash memory is to end. The FLSEDH and FLSEDL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively. The values of the FLSEDH and FLSEDL registers are 00H and 0000H, respectively, under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 30 - 10 Format of Flash End Address Pointer Registers H and L (FLSEDH, FLSEDL)

Address: F02C8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLSEDH	0	0	0	0	EWA19	EWA18	EWA17	EWA16

Address: F02C6H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSEDL	EWA15	EWA14	EWA13	EWA12	EWA11	EWA10	EWA9	EWA8
	7	6	5	4	3	2	1	0
	EWA7	EWA6	EWA5	EWA4	EWA3	EWA2	EWA1	EWA0

Caution 1. The FLSEDH and FLSEDL registers can be rewritten under either of the following conditions.

- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
- The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (the SQEND and ESQEND bits in FSASTH are 0).

Caution 3. The settings of the EWA1 and EWA0 bits are meaningless during programming of the code flash memory.

Table 30 - 10 Method of Setting the FLAPH, FLAPL, FLSEDH, and FLSEDL Registers

Commands Exclusively for Use with the Code/Data Flash Memory Area Sequencer		Settings of the FLAP and FLSED Registers	
FSSQ	Writing	FLAPH, FLAPL: Address from which writing is to proceed FLSEDH, FLSEDL: All 0s	
	Blank checking	For one word: Settings of FLAPH and FLAPL = settings of FLSEDH and FLSEDL For two or more words: Settings of FLAPH and FLAPL < settings of FLSEDH and FLSEDL	
	Block erasure ^{Note}	Code flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP[19:11] bits and the FLAP[10:2] bits, respectively. FLSEDH, FLSEDL: Set the end address and 1s in the EWA[19:11] bits and the EWA[10:2] bits, respectively.
		Data flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP[19:8] bits and the FLAP[7:0] bits, respectively. FLSEDH, FLSEDL: Set the end address and 1s in the EWA[19:8] bits and the EWA[7:0] bits, respectively.
FSSE	All commands	Settings of FLAPH and FLAPL: All 0s Settings of FLSEDH and FLSEDL: All 0s	

Note Set the FLAPH, FLAPL, FLSEDH, and FLSEDL registers so that the following condition is met.
Combined settings of FLAPH and FLAPL ≤ combined settings of FLSEDH and FLSEDL

30.6.2.3 Flash write buffer registers H and L (FLWH, FLWL)

The FLWH and FLWL registers hold data to be written during programming of the flash memory.

The FLWH and FLWL registers can be set by a 16-bit memory manipulation instruction.

The value of each of the FLWH and FLWL registers is 0000H under any of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.
- The flash memory sequencer has finished operating.

Set data to be written to the data flash memory in the 8 lower-order bits of the FLWL register.

Figure 30 - 11 Format of Flash Write Buffer Registers H and L (FLWH, FLWL)

Address: F02CEH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWH	FLW31	FLW30	FLW29	FLW28	FLW27	FLW26	FLW25	FLW24
	7	6	5	4	3	2	1	0
	FLW23	FLW22	FLW21	FLW20	FLW19	FLW18	FLW17	FLW16

Address: F02CCH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWL	FLW15	FLW14	FLW13	FLW12	FLW11	FLW10	FLW9	FLW8
	7	6	5	4	3	2	1	0
	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

Caution 1. The FLWH and FLWL registers can be rewritten under either of the following conditions.

- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
- The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (the SQEND and ESQEND bits in FSASTH are 0).

Caution 3. When writing to the data flash memory, set write data in the 8 lower-order bits of the FLWL register. Set other bits to 0.

30.6.2.4 Flash protect command register (PFCMD)

The PFCMD register protects the flash programming mode control register (FLPMC) against write access.

To enable write access to the FLPMC register, write A5H according to the specific sequence. For details on the procedure for handling the specific sequence, see **30.6.3 Setting the flash memory control mode**.

The PFCMD register can be set by an 8-bit memory manipulation instruction.

Figure 30 - 12 Format of Flash Protect Command Register (PFCMD)

Address: F00C0H
After reset: Undefined
R/W: W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

30.6.2.5 Flash status register (PFS)

The PFS register indicates whether or not a protection error has occurred during write access to the flash programming mode control register (FLPMC).

For details on the conditions for setting and clearing the FPRERR bit, see **30.6.3.1 Procedure for executing the specific sequence.**

The PFS register can be read by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 30 - 13 Format of Flash Status Register (PFS)

Address: F00C1H

After reset: 00H

R/W: R

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR
FPRERR	Protection error flag							
0	No error has occurred.							
1	An error has occurred.							

30.6.2.6 Flash programming mode control register (FLPMC)

The FLPMC register disables or enables writing to the flash memory and selects the programming mode.

Enabling write access to the FLPMC register requires the execution of a specific sequence. For details on the procedure for handling the specific sequence, see **30.6.3 Setting the flash memory control mode**.

The FLPMC register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 08H.

Figure 30 - 14 Format of Flash Programming Mode Control Register (FLPMC)

Address: F02C0H
After reset: 08H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	EEEMD	FWEDIS	0	FLSPM	0

EEEMD	Selection of the programming mode for the data flash memory
0	Non-programmable mode
1	Programming mode

FWEDIS	Software control over enabling or disabling erasure and programming of the code flash memory ^{Note}
0	Enables erasure and programming.
1	Disables erasure and programming.

FLSPM	Selection of the programming mode for the code flash memory
0	Non-programmable mode
1	Programming mode

Note Be sure to keep the value of this bit at 0 until erasure or programming of the code flash memory is completed.

Caution When the extra area sequencer and the code/data flash memory area sequencer are stopped (the SQEND and ESQEND bits in FSASTH are 0), rewriting to the FLPMC register is enabled.

30.6.2.7 Flash area selection register (FLARS)

The FLARS register selects the area of the flash memory for self-programming.

The FLARS register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the FLARS register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 30 - 15 Format of Flash Area Selection Register (FLARS)

Address: F02C1H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLARS	0	0	0	0	0	0	0	EXA

EXA	Selection of the area of the flash memory for self-programming
0	Code/data flash memory areas
1	Extra area

Caution The FLARS register can be rewritten under either of the following conditions.

- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
- The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

30.6.2.8 Flash memory sequencer initial setting register (FSSET)

The FSSET register sets the operating frequency of the flash memory sequencer and makes the initial settings for boot swapping.

The FSSET register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 30 - 16 Format of Flash Memory Sequencer Initial Setting Register (FSSET)

Address: F00B6H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FSSET	TMSPMD	TMBTSEL	0	FSET4	FSET3	FSET2	FSET1	FSET0

TMSPMD	Selection of boot area setting Note
0	Specifies the boot area according to the setting of EX bit 8 (BTFLG) in the security flag and boot swap function setting area of the extra area. BTFLG = 0: Boot cluster 1 as the boot area BTFLG = 1: Boot cluster 0 as the boot area (default)
1	Specifies the boot area according to the setting of the TMBTSEL bit.

TMBTSEL	Specification of the boot area when TMSPMD = 1
0	Specifies boot cluster 0 as the boot area.
1	Specifies boot cluster 1 as the boot area.

FSET4 to FSET0	Setting of the operating frequency of the flash memory sequencer
—	Sets the operating frequency of the flash memory sequencer. For the correspondence between the operating frequency of the flash memory sequencer and the setting of the FSET4 to FSET0 bits, see Table 30 - 11 .

Note Setting the TMSPMD and TMBTSEL bits is not possible while the BTPR bit in FLSEC is 0 (rewriting of the boot area is disabled).

Caution 1. The FSSET register can be rewritten under either of the following conditions.

- The FLSPM bit in the FLPNC register is 1 (code flash memory area: programming mode).
- The EEEMD bit in the FLPNC register is 1 (data flash memory area: programming mode).

Caution 2. The set values of the boot area are immediately reflected. To change the boot area after a reset is released, read the MBTSEL bit in FSASTL while TMSPMD = 0 and set the same value to the TMBTSEL bit. After that, set the TMSPMD bit to 1, and then specify the boot cluster to be activated as the boot area after release from the reset state in the BTFLG bit using the extra area sequencer. The boot cluster set by the BTFLG bit is activated as the boot area at the next reset release.

Table 30 - 11 Correspondence between the Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET4 to FSET0 Bits

Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits	Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits	Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits
32	11111b	31	11110b	30	11101b
29	11100b	28	11011b	27	11010b
26	11001b	25	11000b	24	10111b
23	10110b	22	10101b	21	10100b
20	10011b	19	10010b	18	10001b
17	10000b	16	01111b	15	01110b
14	01101b	13	01100b	12	01011b
11	01010b	10	01001b	9	01000b
8	00111b	7	00110b	6	00101b
5	00100b	4	00011b	3	00010b
2	00001b	1	00000b	—	—

Caution Set the value corresponding to that obtained by rounding the CPU operating frequency up to the nearest whole number in the FSET[4:0] bits.
(For example, when the CPU operating frequency is 4.5 MHz, set the bits for 5 MHz.)
Note that frequencies that are not whole numbers, such as 1.5 MHz, are not available as CPU operating frequencies below 4 MHz.

30.6.2.9 Flash memory sequencer control register (FSSQ)

The FSSQ register specifies operation control and commands for use with the code/data flash memory area sequencer. When the SQST bit in this register is set to 1, the code/data flash memory area sequencer executes the command set in the MDCH, SQMD2, SQMD1, and SQMD0 bits.

The FSSQ register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the FSSQ register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 30 - 17 Format of Flash Memory Sequencer Control Register (FSSQ) (1/2)

Address: F02C5H

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FSSQ	SQST	FSSTP	0	0	MDCH	SQMD2	SQMD1	SQMD0
	SQST	Operation control of the code/data flash memory area sequencer						
	0	The code/data flash memory area sequencer is stopped. Note 1						
	1	The code/data flash memory area sequencer is started.						
	FSSTP	Forcible termination control of the code/data flash memory area sequencer						
	0	The code/data flash memory area sequencer is not forcibly terminated.						
	1	The code/data flash memory area sequencer is forcibly terminated.						

Figure 30 - 17 Format of Flash Memory Sequencer Control Register (FSSQ) (2/2)

MDCH	SQMD2	SQMD1	SQMD0	Commands for Use with the Code/Data Flash Memory Area Sequencer
0	0	0	1	• Writing Writes data stored in the FLWH and FLWL registers to the address specified in the FLAPH and FLAPL registers. Note 2 Writes 4-byte data when the code flash memory area address is specified. Writes the 1-byte data stored in the eight lower-order bits (FLW7 to FLW0) in FLWL to the specified address when the data flash memory area address is specified.
0	0	1	1	• Blank checking of the code flash memory area Checks whether the value of the code flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1. Note 3
1	0	1	1	• Blank checking of the data flash memory area Checks whether the value of the data flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1.
0	1	0	0	• Block erasure Erases blocks in the range from the block start address specified in the FLAPH and FLAPL registers to the block end address specified in the FLSEDH and FLSEDL registers. Note 4
Other than above				Setting prohibited

Note 1. Check that the SQEND bit in the FSASTH register is 1 (the code/data flash memory area sequencer being stopped), and then set the SQST bit to 0 to stop the code/data flash memory area sequencer.

Note 2. Four-byte data can be written to the code flash memory area. Set the two lower-order bits of the FLSEDL register to 00B to be a multiple of 4. For details, see **30.6.6.4 Operations for rewriting the code flash memory area**.

Note 3. Specify a start address (at intervals of four bytes) for blank checking of the code flash memory area. Set the two lower-order bits of the FLSEDL register to 00B to be a multiple of 4. For details, see **30.6.6.4 Operations for rewriting the code flash memory area**.

Note 4. The code flash memory area blocks can be erased in units of 2 Kbytes. The data flash memory blocks can be erased in units of 256 bytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For details, see **30.6.6.4 Operations for rewriting the code flash memory area** and **30.6.6.5 Operations for rewriting the data flash memory area**. For the relationship between the address and block number, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution The FSSQ register can be rewritten under either of the following conditions.
- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode) and the FWEDIS bit is 0 (enabling erasure and programming of the code flash memory).
- The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

30.6.2.10 Flash extra area sequencer control register (FSSE)

The FSSE register specifies operation control and commands for use with the extra area sequencer.

When the ESQST bit of the FSSE register is set to 1, the extra area sequencer executes the command set by the ESQMD3 to ESQMD0 bits.

The FSSE register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the FSSE register is 00H under either of the following conditions.

- Following a reset
- The value of the FLRST bit of the FLRST register is 1.

Figure 30 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE) (1/2)

Address: F00B7H

After reset: 00H

R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
FSSE	ESQST	0	0	0	ESQMD3	ESQMD2	ESQMD1	ESQMD0
	ESQST	Operation control of the extra area sequencer						
	0	The extra area sequencer is stopped. ^{Note}						
	1	The extra area sequencer is started.						

Figure 30 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE) (2/2)

ESQMD3	ESQMD2	ESQMD1	ESQMD0	Commands for Use with the extra area sequencer
0	0	0	1	Write to the flash shield window setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash shield window setting area of the extra area to control flash shield window mode and set the start block and end block. Also, if the setting of EX bit 15 (FSPR) in the flash shield window setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQR) to 1.
0	1	1	0	Write to the flash read protection setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash read protection setting area of the extra area to disable changing of the flash read protection setting and set the start block and end block. Also, if the setting of EX bit 31 (SWPR) in the flash read protection setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQR) to 1.
0	1	1	1	Write to the security flag and boot swap function setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash memory security flag and boot swap function setting area of the extra area to disable block erasure, writing, and rewriting of the boot area and set selection of the boot area. Also, if the setting of EX bit 9 (BTPR) in the security flag and boot swap function setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQR) to 1.
Other than above				Setting prohibited

Note Check that the ESQEND bit in the FSASTH register is 1 (the extra area sequencer being stopped), and then set the ESQST bit to 0 to stop the extra area sequencer.

Caution 1. The FSSE register can be rewritten when the following condition is met.

- The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode) and the FWEDIS bit is 0 (enabling erasure and programming of the code flash memory).

Caution 2. To write to the extra area, set the EXA bit of the FLARS register to 1 and set the data to be written in the FLWH and FLWL registers before activating the extra area sequencer.

Caution 3. Rewrite the ESQMD3 to ESQMD0 bits while the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in FSASTH).

30.6.2.11 Flash registers initialization register (FLRST)

The FLRST register initializes all registers related to the sequencer whether it is to be used with the extra area or code/data flash memory areas.

The FLRST register can be set by a 1-bit or 8-bit memory manipulation instruction.

For details about how to manipulate the FLRST register, see **30.6.4 Initializing the registers for use with the flash memory sequencer**.

The value of this register following a reset is 00H.

Figure 30 - 19 Format of Flash Registers Initialization Register (FLRST)

Address: F02C9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLRST	0	0	0	0	0	0	0	FLRST

FLRST	Control of initializing the registers
0	The registers are not reset.
1	The FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers are reset.

Caution 1. Registers can be initialized by setting the FLRST bit to 1 only when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in FSASTH).

Caution 2. When using the sequencer, be sure to set the FLRST bit to 0 before setting the FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers. Do not set the FLRST bit to 1 during operation of the sequencer.

30.6.2.12 Flash memory sequencer status registers H and L (FSASTH, FSASTL)

The FSASTH and FSASTL registers indicate the results of the respective operations of the flash memory sequencer when it has been used with the extra area or code/data flash memory areas.

The FSASTH and FSASTL registers can be read by a 1-bit or 8-bit memory manipulation instruction.

Figure 30 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (1/2)

Address: F02CBH
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTH	ESQEND	SQEND	0	0	0	0	0	0

Address: F02CAH
After reset: Undefined **Note**
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTL	MBTSEL	MOPEN	ESEQER	SEQER	BLER	0	WRER	ERER

Note The initial value of the MBTSEL bit is undefined because it depends on the value of the BTFLG bit (boot area switching flag) stored in the extra area.

ESQEND	Extra area sequencer operation end status flag
0	The extra area sequencer is operating or stopped by setting the ESQST bit to 0.
1	The extra area sequencer is stopped.

SQEND	Code/data flash memory area sequencer operation end status flag
0	The code/data flash memory area sequencer is operating or stopped by setting the SQST bit to 0.
1	The code/data flash memory area sequencer is stopped.

MBTSEL	Boot flag monitoring bit ^{Note}
0	BTFLG = 1 (boot area: boot cluster 0)
1	BTFLG = 0 (boot area: boot cluster 1)

MOPEN	Code/data flash memory area sequencer operation status flag
0	The code/data flash memory area sequencer is stopped.
1	The code/data flash memory area sequencer is operating.

Figure 30 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (2/2)

ESEQR	Error flag of the extra area sequencer
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer is activated.	

SEQR	Error flag of the flash memory sequencer
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

BLER	Error flag for the blank check command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

WRER	Error flag for the write command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during writing.	

ERER	Error flag for the block erase command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during block erasure.	

Note This bit indicates the inverse of the value of the BTFLG bit (boot area switching flag) stored in the extra area.

30.6.2.13 Flash security flag monitoring register (FLSEC)

The FLSEC register monitors the information on the security flag and boot swap function settings in the extra area.
The FLSEC register can be read by a 16-bit memory manipulation instruction.

Figure 30 - 21 Format of Flash Security Flag Monitoring Register (FLSEC)

Address: F00B0H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLSEC	0	0	0	WRPR	0	SEPR	BTPR	BTFLG
	7	6	5	4	3	2	1	0
	0	0	0	0	SWPR	0	IFPR	IDEN
WRPR	Write-disabled flag							
0	Writing is disabled.							
1	Writing is enabled.							
SEPR	Block erase-disabled flag							
0	Block erasure is disabled.							
1	Block erasure is enabled.							
BTPR	Boot area rewrite-disabled flag							
0	Rewriting of the boot area is disabled.							
1	Rewriting of the boot area is enabled.							
BTFLG	Boot area switching flag							
0	The boot area is boot cluster 1.							
1	The boot area is boot cluster 0.							
SWPR	Changing of the flash memory read protection setting disabled flag							
0	Changing of the flash memory read protection setting is disabled.							
1	Changing of the flash memory read protection setting is enabled.							
IFPR	Connection to the programmer and on-chip debugger disabled flag							
0	Connection to the programmer and on-chip debugger is disabled.							
1	Connection to the programmer and on-chip debugger is enabled.							
IDEN	Programmer connection ID authentication enabled flag							
0	ID authentication is enabled.							
1	ID authentication is disabled.							

30.6.2.14 Flash FSW monitoring register E (FLFSWE)

The FLFSWE register monitors the end block number specified for the flash memory shield area and whether the shield area is inside or outside the window range.

After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWE register.

For details on the flash shield window function, see **30.8 Flash Shield Window Function**.

The FLFSWE register can be read by a 16-bit memory manipulation instruction.

Figure 30 - 22 Format of Flash FSW Monitoring Register E (FLFSWE)

Address: F00B4H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWE	FSWC	0	0	0	0	0	0	FSWE8
	7	6	5	4	3	2	1	0
	FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0
	FSWC	Setting of the flash memory shield area						
	0	Inside shield mode The flash memory shield area is set inside the window range.						
	1	Outside shield mode The flash memory shield area is set outside the window range.						
	FSWE8 to FSWE0	End block number of the flash memory shield area						
	—	End block number + 1 Note						

Note These bits show the value set in the extra area. The actual end block number is (the value of the FSWE8 to FSWE0 bits - 1). Though the end block number is specified for serial programming, (end block number + 1) is set in the extra area. For details, see **Table 30 - 12**.

30.6.2.15 Flash FSW monitoring register S (FLFSWS)

The FLFSWS register monitors the start block number of the flash memory shield area and whether rewriting of the flash shield window setting is disabled or enabled.

After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWS register.

For details on the flash shield window function, see **30.8 Flash Shield Window Function**.

The FLFSWS register can be read by a 16-bit memory manipulation instruction.

Figure 30 - 23 Format of Flash FSW Monitoring Register S (FLFSWS)

Address: F00B2H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWS	FSPR	0	0	0	0	0	0	FSWS8
	7	6	5	4	3	2	1	0
	FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0
	FSPR	Changing of the flash shield window setting disabled flag						
	0	Changing of the flash shield window setting is disabled.						
	1	Changing of the flash shield window setting is enabled.						
	FSWS8 to FSWS0	Start block number of the flash memory shield area						
	—	Start block number						

30.6.2.16 Data flash control register (DFLCTL)

The DFLCTL register enables or disables access to the data flash memory area.

The DFLCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 30 - 24 Format of Data Flash Control Register (DFLCTL)

Address: F0090H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash memory area access control
0	Access to the data flash memory area is disabled.
1	Access to the data flash memory area is enabled.

30.6.2.17 Interrupt vector jump enable register (VECTCTRL)

The VECTCTRL register specifies the destination address of the jump in response to any interrupt which occurs during self-programming.

The VECTCTRL register can be set by an 8-bit memory manipulation instruction.

The value of this register following a reset is 00H.

Figure 30 - 25 Format of Interrupt Vector Jump Enable Register (VECTCTRL)

Address: F00FFH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
VECTCTRL	0	0	0	0	0	0	0	VECTCTRL

VECTCTRL	Setting the interrupt branch destinations
0	Interrupt vector addresses in the ROM
1	Specified addresses in the RAM ^{Note}

Note A destination address in the RAM is specified by the FLSIVC1 and FLSIVC0 registers.
For details, see **30.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

30.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

The FLSIVC0 and FLSIVC1 registers specify the destination address of the jump in response to any interrupt which occurs during self-programming.

For details on how to handle an interrupt during self-programming, see **30.6.7 Interrupts in code flash memory programming mode**.

The FLSIVC0 and FLSIVC1 registers can be set by a 16-bit memory manipulation instruction.

The values of the FLSIVC0 and FLSIVC1 registers following a reset are 0000H and 000FH, respectively.

Figure 30 - 26 Format of Interrupt Vector Change Registers 0 and 1 (FLSIVC0, FLSIVC1)

Address: F0480H, F0481H

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC0	FLSIV15	FLSIV14	FLSIV13	FLSIV12	FLSIV11	FLSIV10	FLSIV9	FLSIV8
	7	6	5	4	3	2	1	0
	FLSIV7	FLSIV6	FLSIV5	FLSIV4	FLSIV3	FLSIV2	FLSIV1	FLSIV0

Address: F0482H, F0483H

After reset: 000FH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	FLSIV19	FLSIV18	FLSIV17	FLSIV16

Caution Set the values of the 4 higher-order bits and 16 lower-order bits of the branch destination address in the FLSIVC1 and FLSIVC0 registers, respectively.

30.6.3 Setting the flash memory control mode

The flash memory has the following flash memory control modes.

- Code flash memory programming mode

The code flash memory area and the extra area can be rewritten.

- Data flash memory programming mode

The data flash memory area and the extra area can be rewritten.

- Non-programmable mode

The flash memory (code flash memory area, data flash memory area, and extra area) cannot be rewritten.

To rewrite the flash memory, set the flash memory control mode to code flash memory programming mode or data flash memory programming mode. Setting each of the flash memory control modes requires executing the specific sequence for setting the flash protect command register (PFCMD) and flash programming mode control register (FLPMC).

Caution For handling of the data flash memory area, follow the procedure while access to the data flash memory is enabled (the value of the DFLEN bit of the DFLCTL register is 1).

30.6.3.1 Procedure for executing the specific sequence

Writing the required values to the flash protect command register (PFCMD) and the flash programming mode control register (FLPMC) by following steps 1 to 4 below enables the transitions to each of the flash memory control modes.

- <1> Write A5H to the PFCMD register.
- <2> Write the value to be set to the FLPMC register.
- <3> Write the inverse of the value to be set to the FLPMC register.
- <4> Write the value to be set to the FLPMC register.

- The specific sequence can only be executed while the value of the FLRST bit of the FLRST register is 0 and the flash memory sequencer is stopped.
- If writing to any other memory area or register is attempted in the intervals between steps 1 to 4 during execution of the specific sequence, a protection error occurs, writing to the specified register does not proceed, and the FPRERR flag of the flash status register (PFS) is set to 1. The FPRERR flag is cleared following a reset or when execution of the specific sequence is re-started.

30.6.3.2 Procedure for entry to the code flash memory programming mode

The procedure for entry to the code flash memory programming mode is given below.

- <1> Write A5H to the PFCMD register.
- <2> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).
- <3> Write FDH to the FLPMC register (inverse of 02H).
- <4> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).

30.6.3.3 Procedure for entry to the data flash memory programming mode

The procedure for entry to the data flash memory programming mode is given below.

- <1> Write A5H to the PFCMD register.
- <2> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).
- <3> Write EFH to the FLPMC register (inverse of 10H).
- <4> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).

30.6.3.4 Procedure for entry to the non-programmable mode

After executing the procedure for entry to the non-programmable mode from the code flash memory programming mode or data flash memory programming mode and waiting^{Note}, reading from the target flash memory for the programming mode before the mode transition becomes possible.

Note Wait time: 10 μ s

<When the interrupt vector has not been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are interrupt vector addresses in the ROM is given below.

- <1> Write A5H to the PFCMD register.
- <2> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
- <3> Write F7H to the FLPMC register (inverse of 08H).
- <4> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
- <5> Reading from the target flash memory area becomes possible after waiting for 10 μ s.

<When the interrupt vectors have been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are changed to specified addresses in the RAM is given below.

- <1> Write A5H to the PFCMD register.
- <2> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
- <3> Write FFH to the FLPMC register (inverse of 00H).
- <4> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
- <5> Reading from the target flash memory area becomes possible after waiting for 10 μ s.

30.6.4 Initializing the registers for use with the flash memory sequencer

The following registers can be initialized by setting the FLRST bit of the flash registers initialization register (FLRST) to 1.

Target registers: FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, FSSE

The procedure for clearing the target registers is given below.

- <1> Set the FLRST bit.
- <2> Use software code to wait for at least one cycle of the CPU operating clock.
- <3> Clear the FLRST bit.

30.6.5 Setting the operating frequency of the flash memory sequencer

Set the value corresponding to the operating frequency of the CPU (1 MHz to 32 MHz) in the FSET4 to FSET0 bits of the flash memory sequencer initial setting register (FSSET). When making the setting, round the CPU operating frequency value up to the nearest whole number.

(Example: When the CPU operating frequency is 4.5 MHz, set the bits for 5 MHz.)

How to set the operating frequency of the flash memory sequencer is described below.

- <1> Enter the code flash memory programming mode or data flash memory programming mode.
For the procedures for entry to each of these flash memory programming modes, see **30.6.3.1 Procedure for executing the specific sequence**, **30.6.3.2 Procedure for entry to the code flash memory programming mode**, and **30.6.3.3 Procedure for entry to the data flash memory programming mode**.
- <2> After reading from the flash memory sequencer initial setting register (FSSET), set the TMSPMD and TMBTSEL bits to the same values as those read from the FSSET register and the FSET4 to FSET0 bits to the value corresponding to the CPU operating frequency, respectively.

Caution When using the code/data flash memory area sequencer and the extra area sequencer to rewrite the code or data flash memory or the extra area, set the value corresponding to the CPU operating frequency in the FSET4 to FSET0 bits of the FSSET register before doing so.
Note that if rewriting of any of these areas is attempted while the value corresponding to the CPU operating frequency is not correct, operation is undefined and written data are not guaranteed. Even if the values in the flash memory are as expected immediately after rewriting, retaining the values for any specified period is not guaranteed.

30.6.6 Rewriting the flash memory

30.6.6.1 Overview

The flash memory sequencer serves as a code/data flash memory area sequencer or an extra area sequencer. In the former role, it is used to rewrite the code flash memory area or data flash memory area, while in the latter role it is used to rewrite the extra area. To rewrite a given area, execute the corresponding commands for use with the sequencer.

30.6.6.2 Selecting the area to be rewritten

Select the code flash memory area, data flash memory area, or extra area for rewriting by using the flash area selection register (FLARS).

30.6.6.3 Commands for use with the code/data flash memory area sequencer

Use the commands exclusively used for the code/data flash memory area sequencer to rewrite the code or data flash memory area. To execute a command, set the target command in the SQMD2 to SQMD0 bits of the flash memory sequencer control register (FSSQ) and set the SQST bit to 1. The SQMD2 to SQMD0 bits and the SQST bit can be set simultaneously.

For the commands exclusively for use with the code/data flash memory area sequencer, see **30.6.2.9 Flash memory sequencer control register (FSSQ)**.

30.6.6.4 Operations for rewriting the code flash memory area

To rewrite the code flash memory area, execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers. Allocate the processing software to rewrite the code flash memory area in the RAM and execute it from the RAM.

Units for block erasure and for writing in rewriting of the code flash memory area

- Unit for blocks to be erased: 2 Kbytes
- Unit for writing: 4 bytes

<Handling the commands>

- <1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **30.6.3.1 Procedure for executing the specific sequence** and **30.6.3.2 Procedure for entry to the code flash memory programming mode**.
- <2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).
- <3> Before executing each command, set the address data, write data, and command in the corresponding registers.
 - Block erasure
Set the block start address^{Note 1} (example: 0x002000) of the code flash memory to be erased in the FLAPH and FLAPL registers.
Set the block end address^{Note 1} (example: 0x0027FF) of the code flash memory to be erased in the FLSEDH and FLSEDL registers.
 - Writing
Set the start address^{Note 2} (example: 0x002000) of the flash memory to be written in the FLAPH and FLAPL registers.
Set 4-byte write data in the FLWH and FLWL registers.
 - Blank checking
Set the start address^{Note 2} (example: 0x002000) of the flash memory to be blank-checked in the FLAPH and FLAPL registers.
Set the end address (example: 0x0027FF) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers.
When blank checking is only to be applied to one word (four bytes), set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.
- <4> When the value of the command to be executed is set in the MDCH and SQMD2 to SQMD0 bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD2 to SQMD0 bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.
 - Block erasure: 84H
 - Writing: 81H
 - Blank checking of the code flash memory area: 83H
- <5> Wait until the command for use with the code/data flash memory area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled "Procedure for checking the completion of commands for use with the code/data flash memory area sequencer" in **30.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

- Continuing command processing

The same command or another command can be executed by continuously updating the address data and write data in step 3 with the state remaining in code flash memory programming mode.

- Completing command processing

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **30.6.3.1 Procedure for executing the specific sequence** and **30.6.3.4 Procedure for entry to the non-programmable mode**.

Note 1. The code flash memory area blocks can be erased in units of 2 Kbytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Note 2. The code flash memory area can be written and blank-checked in units of 4 bytes. Therefore, set the two lower-order bits of the FLAPL register for specifying the address to 00B (an integer of 4).

30.6.6.5 Operations for rewriting the data flash memory area

To rewrite the data flash memory area, execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers.

Units for block erasure and for writing in rewriting of the data flash memory area

- Unit for blocks to be erased: 256 bytes
- Unit for writing: 1 byte

<Handling the commands>

- <1> Enter the data flash memory programming mode. For the procedure for entry to the data flash memory programming mode, see **30.6.3.1 Procedure for executing the specific sequence** and **30.6.3.3 Procedure for entry to the data flash memory programming mode**.
- <2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).
- <3> Before executing each command, set the address data, write data, and command in the corresponding registers.
 - Block erasure
Set the block start address^{Note} (example: 0x0F1100) of the data flash memory to be erased in the FLAPH and FLAPL registers.
Set the block end address^{Note} (example: 0x0F11FF) of the data flash memory to be erased in the FLSEDH and FLSEDL registers.
 - Writing
Set the start address (example: 0x0F1101) of the flash memory to be written in the FLAPH and FLAPL registers.
Set the write data in the 8 lower-order bits of the FLWL register.
 - Blank checking:
Set the start address (example: 0x0F1100) of the flash memory to be blank-checked in the FLAPH and FLAPL registers.
Set the end address (example: 0x0F11FF) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers.
When blank checking is only to be applied to one byte, set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.
- <4> When the value of the command to be executed is set in the MDCH and SQMD2 to SQMD0 bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD2 to SQMD0 bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.
 - Block erasure: 84H
 - Writing: 81H
 - Blank checking of the data flash memory area: 8BH
- <5> Wait until the command for use with the code/data flash memory area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled "Procedure for checking the completion of commands for use with the code/data flash memory area sequencer" in **30.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

- Continuing command processing

The same command or another command can be executed by continuously updating the address data and write data in step 3 with the state remaining in data flash memory programming mode.

- Completing command processing

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see

30.6.3.1 Procedure for executing the specific sequence and **30.6.3.4 Procedure for entry to the non-programmable mode**.

Note

The data flash memory area blocks can be erased in units of 256 bytes. Therefore, set the 8 lower-order bits of the FLAPL register for specifying the start address to 0000 0000B (an integer of 256). Also set the 8 lower-order bits of the FLSEDL register for specifying the end address to 1111 1111B.

30.6.6.6 Commands for use with the extra area sequencer

Use the commands for use with the extra area sequencer to rewrite the settings of the flash shield window, flash read protection, flash security, and boot swap function in the extra area. To execute a command, set the command to be executed in the ESQMD3 to ESQMD0 bits of the flash extra area sequencer control register (FSSE), and then set the ESQST bit to 1. The ESQMD3 to ESQMD0 bits and the ESQST bit can be set simultaneously.

Allocate the extra area sequencer command processing software in the RAM and execute it from the RAM.

30.6.6.7 Operations for rewriting the extra area

To rewrite the extra area, enter the code flash memory programming mode and then execute commands for use with the extra area sequencer. Before starting to execute a command, set the data required for executing each command in the corresponding registers.

<Handling the commands>

- <1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **30.6.3.1 Procedure for executing the specific sequence** and **30.6.3.2 Procedure for entry to the code flash memory programming mode**.
- <2> Set the EXA bit of the FLARS register to 1 (extra area).
- <3> Before executing a command, set 4-byte data in the FLWH and FLWL registers. Each bit of the combined FLW31 to FLW0 bits of the FLWH and FLWL registers corresponds to EX bits 31 to 0 of the target extra area data. For details on data to be set for each command, see **30.6.6.8 Data to be set for the commands for use with the extra area sequencer**.
- <4> When the value of the command to be executed is set in the ESQMD3 to ESQMD0 bits of the FSSE register and the ESQST bit is set to 1, the extra area sequencer executes the specified command. The ESQMD3 to ESQMD0 bits and the ESQST bit can be set simultaneously. If these bits are set simultaneously, the FSSE register is set to the following values.
 - Write data to the flash shield window setting area: 81H
 - Write data to the flash read protection setting area: 86H
 - Write data to the security flag and boot swap function setting area: 87H
- <5> Wait until the command for use with the extra area sequencer is complete. For details on the procedure for waiting for the completion of command execution, see the section titled "Procedure for checking the completion of commands for use with the extra area sequencer" in **30.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.
- <6> Processing after executing a command
 - Continuing command processing
The same command or another command can be executed by continuously updating the FLWH and FLWL register data to be set in the extra area in step 3 with the state remaining in code flash memory programming mode.
 - Completing command processing
Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **30.6.3.1 Procedure for executing the specific sequence** and **30.6.3.4 Procedure for entry to the non-programmable mode**.

30.6.6.8 Data to be set for the commands for use with the extra area sequencer

Writing to the extra area proceeds per 4 bytes.

Each command for use with the extra area sequencer writes the data set in the combined FLW31 to FLW0 bits of the FLWH and FLWL registers to EX bits 31 to 0 in the extra area corresponding to the given command.

(1) Write to the flash shield window setting area

Set the data in the FLWH and FLWL registers to the flash shield window setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
FSWC	1	1	1	1	1	1	FSWE8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
FSPR	1	1	1	1	1	1	FSWS8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

Bit Name	Setting
FSWC	Specifies the range of the flash memory shield area. 0: Flash memory shield area: Inside the window range 1: Flash memory shield area: Outside the window range (default)
FSPR	Specifies whether to enable or disable changing of the flash shield window setting area. 0: Changing of the flash shield window setting area is disabled. 1: Changing of the flash shield window setting area is enabled (default).
FSWE8 - FSWE0	Flash shield window end block setting area Specify the block number (end block number + 1). ^{Note}
FSWS8 - FSWS0	Flash shield window start block setting area Specify the start block number. ^{Note}

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution The value of the FSPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.

Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.

- SEPR = 0 (block erasure is disabled)
- BTPR = 0 (rewriting of the boot area is disabled)

Moreover, setting the FSPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

(2) Write to the flash read protection setting area

Set the data in the FLWH and FLWL registers to the flash read protection setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
SWPR	1	1	1	1	1	1	UPAddr8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
UPAddr7	UPAddr6	UPAddr5	UPAddr4	UPAddr3	UPAddr2	UPAddr1	UPAddr0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
1	1	1	1	1	1	1	LOWAddr8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
LOWAddr7	LOWAddr6	LOWAddr5	LOWAddr4	LOWAddr3	LOWAddr2	LOWAddr1	LOWAddr0

Bit Name	Setting
SWPR	Specifies whether to enable or disable changing of the flash read protection setting area. 0: Changing of the flash read protection setting area is disabled. 1: Changing of the flash read protection setting area is enabled (default).
UPAddr8 to UPAddr0	Flash read protection end block setting area Specify the end block number. Note
LOWAddr8 to LOWAddr0	Flash read protection start block setting area Specify the start block number. Note

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**. The flash read protection setting area cannot be read after a reset is released.

Caution The value of the SWPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.

Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.

- SEPR = 0 (block erasure is disabled)
- BTPR = 0 (rewriting of the boot area is disabled)

Moreover, setting the SWPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

(3) Write to the security flag and boot swap function setting area

Set the data in the FLWH and FLWL registers to the security flag and boot swap function setting area. For details of the security settings, see **30.9 Security Settings**.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
1	1	1	1	1	1	1	1
EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
1	1	1	1	1	1	1	1
EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
1	1	1	WRPR	1	SEPR	BTPR	BTFLG
EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
1	1	1	1	1	IFPR	1	IDEN

Bit Name	Setting
WRPR	Specifies whether to enable or disable writing in serial programming mode. Writing in serial programming mode is disabled. 0: Writing in serial programming mode is disabled. 1: Writing in serial programming mode is enabled (default).
SEPR	Specifies whether to enable or disable block erasure. Block erasure in serial programming mode is disabled. 0: Block erasure in serial programming mode is disabled. 1: Block erasure in serial programming mode is enabled (default).
BTPR	Specifies whether to enable or disable rewriting of the boot area. Rewriting of the boot area and boot swapping are disabled. 0: Rewriting of the boot area and boot swapping are disabled. 1: Rewriting of the boot area and boot swapping are enabled (default).
BTFLG	Specifies the boot area when the TMSPMD bit in the FSSET register is 0. 0: Boot area: Boot cluster 1 1: Boot area: Boot cluster 0 (default)
IFPR	Specifies whether to enable or disable connection to the programmer and on-chip debugger. The serial programming mode and connection to the on-chip debugger are disabled. 0: Serial programming mode and connection to the on-chip debugger are disabled. 1: Serial programming mode and connection to the on-chip debugger are enabled (default).
IDEN	Specifies whether to enable or disable programmer connection ID authentication. ID authentication for connection in serial programming mode proceeds. 0: ID authentication for connection in serial programming mode is enabled. 1: ID authentication for connection in serial programming mode is disabled (default).

(Cautions are listed on the next page.)

Caution 1. When changing the value of the BTFLG bit, set all other bits to 1.

Caution 2. When changing the values of security flags other than BTFLG to 0 (disabling), read the register first and set the BTFLG bit to the same value as was read, and set the other bits to 1.

Caution 3. The value of the WRPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased.

Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.

- SEPR = 0 (block erasure is disabled)

- BTPR = 0 (rewriting of the boot area is disabled)

Moreover, setting the WRPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

Caution 4. Restoring the value of any among the SEPR, BTPR, IFPR, and IDEN bits to 1 after having set it to 0 is not possible.

30.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas

Perform the following procedure for checking the completion of commands when terminating an activated command for use with the code/data flash memory area sequencer and the extra area sequencer.

- Procedure for checking the completion of commands for use with the code/data flash memory area sequencer
 - (1) Activate a command for use with the code/data flash memory area sequencer, and then wait until the SQEND flag of the FSASTH register is set to 1.
 - (2) Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST bit of the FSSQ register.
 - (3) Wait until the SQEND flag of the FSASTH register is cleared. When the SQEND flag is cleared, the command is completed and the sequencer stops.
- Procedure for checking the completion of commands for use with the extra area sequencer
 - (1) Activate a command for use with the extra area sequencer, and then wait until the ESQEND flag of the FSASTH register is set to 1.
 - (2) Confirm that the ESQEND flag of the FSASTH register is set to 1, and then clear the ESQST bit of the FSSE register.
 - (3) Wait until the ESQEND flag of the FSASTH register is cleared. When the ESQEND flag is cleared, the command is completed and the sequencer stops.

30.6.6.10 Procedure for forcibly terminating a command for use with the code/data flash memory area sequencer

A command for use with the code/data flash memory area sequencer can be forcibly terminated while it is in progress. Note that no command for use with the extra area sequencer can be forcibly terminated while it is in progress.

<Procedure for forcible termination>

- (1) Set the FSSTP bit of the FSSQ register to 1 after a command for use with the code/data flash memory area sequencer is activated until the SQST bit of the FSSQ register in step 2 is cleared.
- (2) Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST and FSSTP bits of the FSSQ register.
- (3) Wait until the SQEND flag of the FSASTH register is automatically cleared. When the SQEND flag is cleared, the forcible termination is completed.

30.6.7 Interrupts in code flash memory programming mode

30.6.7.1 Overview

When an interrupt occurs, reference to the interrupt vector in the ROM proceeds, and execution branches to the interrupt processing that is allocated in the up to 64-Kbyte ROM space in accord with the address in the interrupt vector (16 bits), from which the interrupt processing is executed. However, in code flash memory programming mode, where the code flash memory and extra area can be rewritten, referring to the ROM is not possible. As a result, the interrupt processing cannot be executed in this way.

Note that even when referring to the ROM is not possible, the interrupt processing can be executed by changing the branch destinations of the interrupts. Specifically, changing the branch destinations of all interrupts to specified addresses in the RAM enables the execution of interrupt processing from the RAM instead of by using the interrupt vector and interrupt processing in the ROM.

30.6.7.2 Operation to change the branch destinations of the interrupts

To change the branch destinations of the interrupts, set the interrupt vector change registers 1, 0 (FLSIVC1, FLSIVC0) and interrupt vector jump enable register (VECTCTRL) so that execution branches to an address in the RAM following an interrupt. This enables execution of the interrupt processing from the RAM without reference to the interrupt vector in the ROM, even when an interrupt occurs in code flash memory programming mode.

The FLSIVC1 and FLSIVC0 registers are used to specify the branch destination address in response to interrupts which occur during rewriting of the code flash memory or extra area. Set the values of the 16 lower-order bits and 4 higher-order bits of the branch destination address in the FLSIVC0 and FLSIVC1 registers, respectively.

Settings for control of the branch destination of interrupts that occur during self-programming are as follows.

- For branching to vector addresses in the ROM: VECTCTRL = 0 or FWEDIS of FLPMC = 1
- For branching to a RAM address: VECTCTRL = 1 (with FWEDIS of FLPMC = 0)

- Caution 1.** The user must determine the type of interrupt by checking the interrupt flags. Therefore, the interrupt flags are not automatically cleared after the VECTCTRL register has been set.
- Caution 2.** A changed interrupt branch destination in the ROM is not specifiable.
- Caution 3.** The change to the interrupt branch destination made by the VECTCTRL register is only effective during self-programming.
- Caution 4.** Disable interrupts while changing the interrupt branch destination to an address in the RAM.

30.6.7.3 Operation to change the interrupt branch destination

To specify interrupt processing from the RAM, update the FLSIVC1, FLSIVC0, and VECTCTRL registers while the value of the FWEDIS bit of the flash programming mode control register (FLPMC) is 0. Execute the specific sequence to handle the FWEDIS bit of the FLPMC register, and set the FLSIVC1, FLSIVC0, and VECTCTRL registers. The interrupt branch destination is thus changed to an address in RAM.

<When changing the interrupt branch destinations to an address in RAM >

The following describes operation for changing all interrupt branch destinations to a specified address in the RAM.

- Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 0.
 - <1> Write A5H to the PFCMD register.
 - <2> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
 - <3> Write FFH to the FLPMC register (inverse of 00H).
 - <4> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
- Specify an address in RAM in the FLSIVC1 and FLSIVC0 registers.
- Set the VECTCTRL register to 01H to set the RAM address as the branch destination for interrupts.
- Restore the saved setting for enabling or disabling interrupts.

Caution 1. Retain the value 0 in the FWEDIS bit as long as interrupt processing from the RAM remains specified.

Caution 2. Do not set the interrupt branch destination in the saddr space at addresses from FFE20H to FFEFFH.

Caution 3. When instructions are being executed from the RAM area and RAM parity error resets are enabled (RPERDIS = 0), initialize the RAM area where data access is to proceed + 10 bytes.

<When returning the interrupt branch destinations from the RAM address to the vectors in the ROM>

The following describes operation for returning the interrupt branch destinations to the addresses indicated by the interrupt vectors in the ROM (default state).

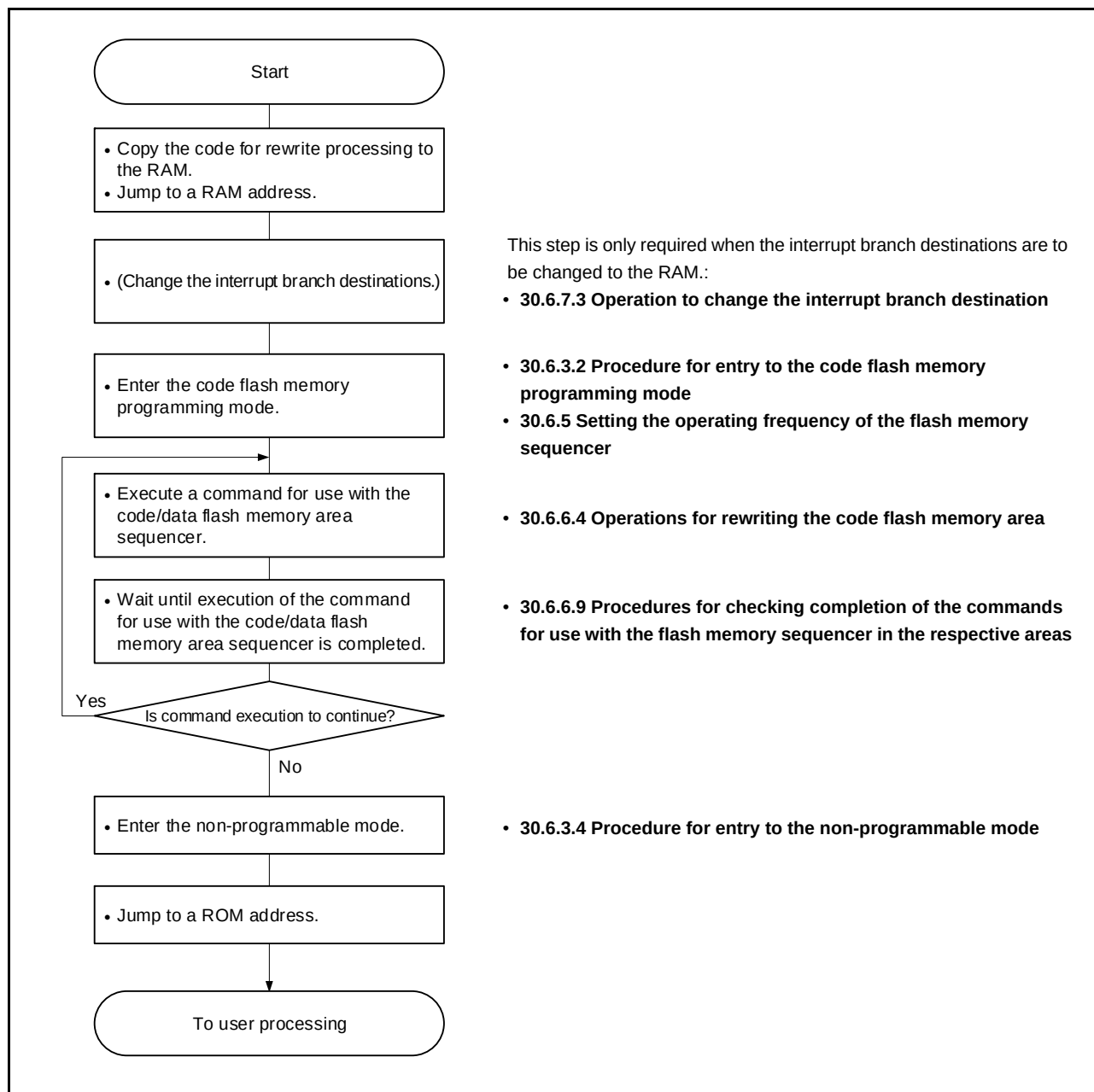
- Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 1.
 - <1> Write A5H to the PFCMD register.
 - <2> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
 - <3> Write F7H to the FLPMC register (inverse of 08H).
 - <4> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
- Set the VECTCTRL register to 00H to set the vector addresses in the ROM as the interrupt branch destinations.
- Restore the saved setting for enabling or disabling interrupts.

30.6.8 Example of executing the commands to rewrite the flash memory areas

30.6.8.1 Example of executing the commands to rewrite the code flash memory area

Figure 30 - 27 shows the flow of executing the commands to rewrite the code flash memory area.

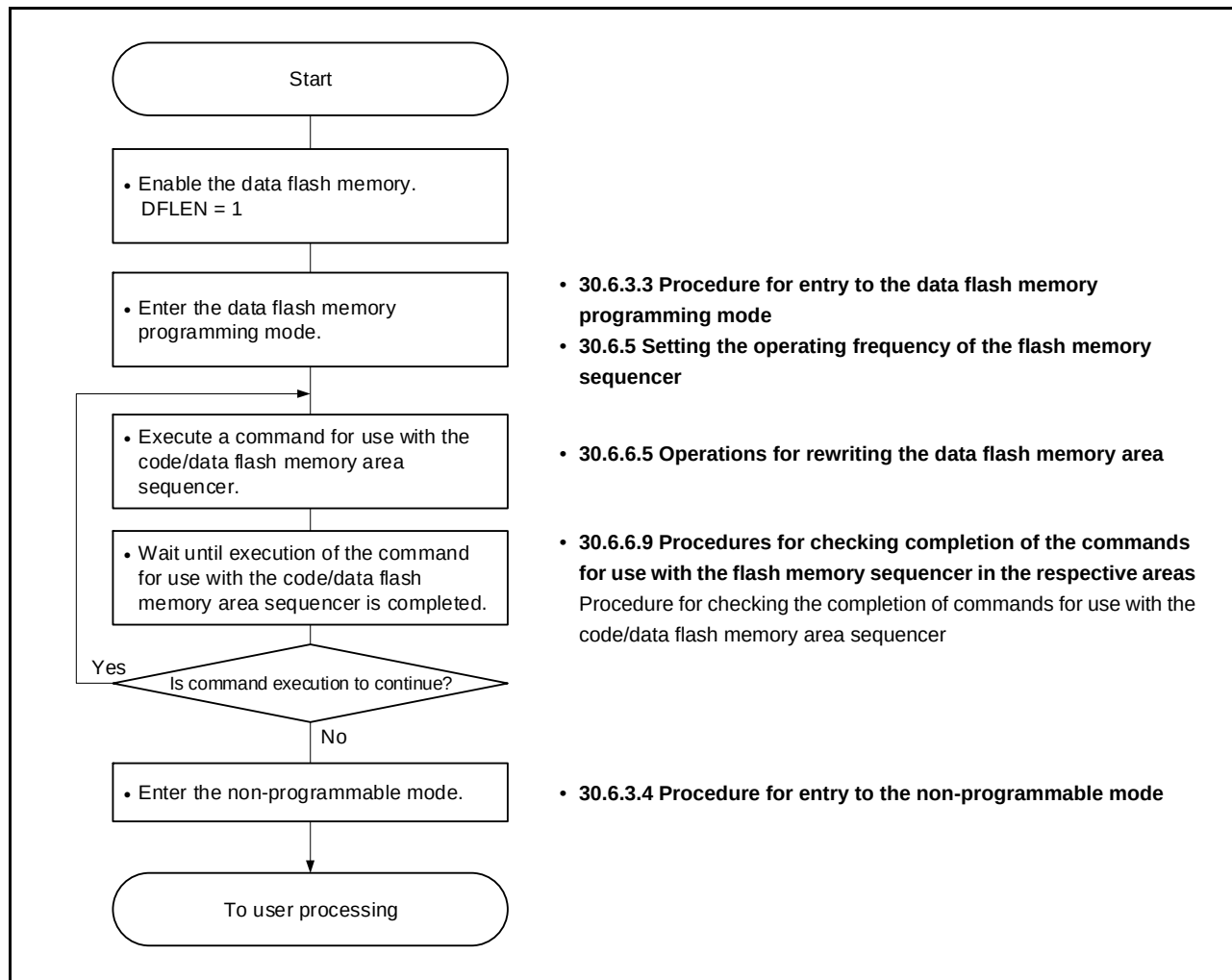
Figure 30 - 27 Flow of Executing the Commands to Rewrite the Code Flash Memory Area



30.6.8.2 Example of executing the commands to rewrite the data flash memory area

Figure 30 - 28 shows the flow of executing the commands to rewrite the data flash memory area.

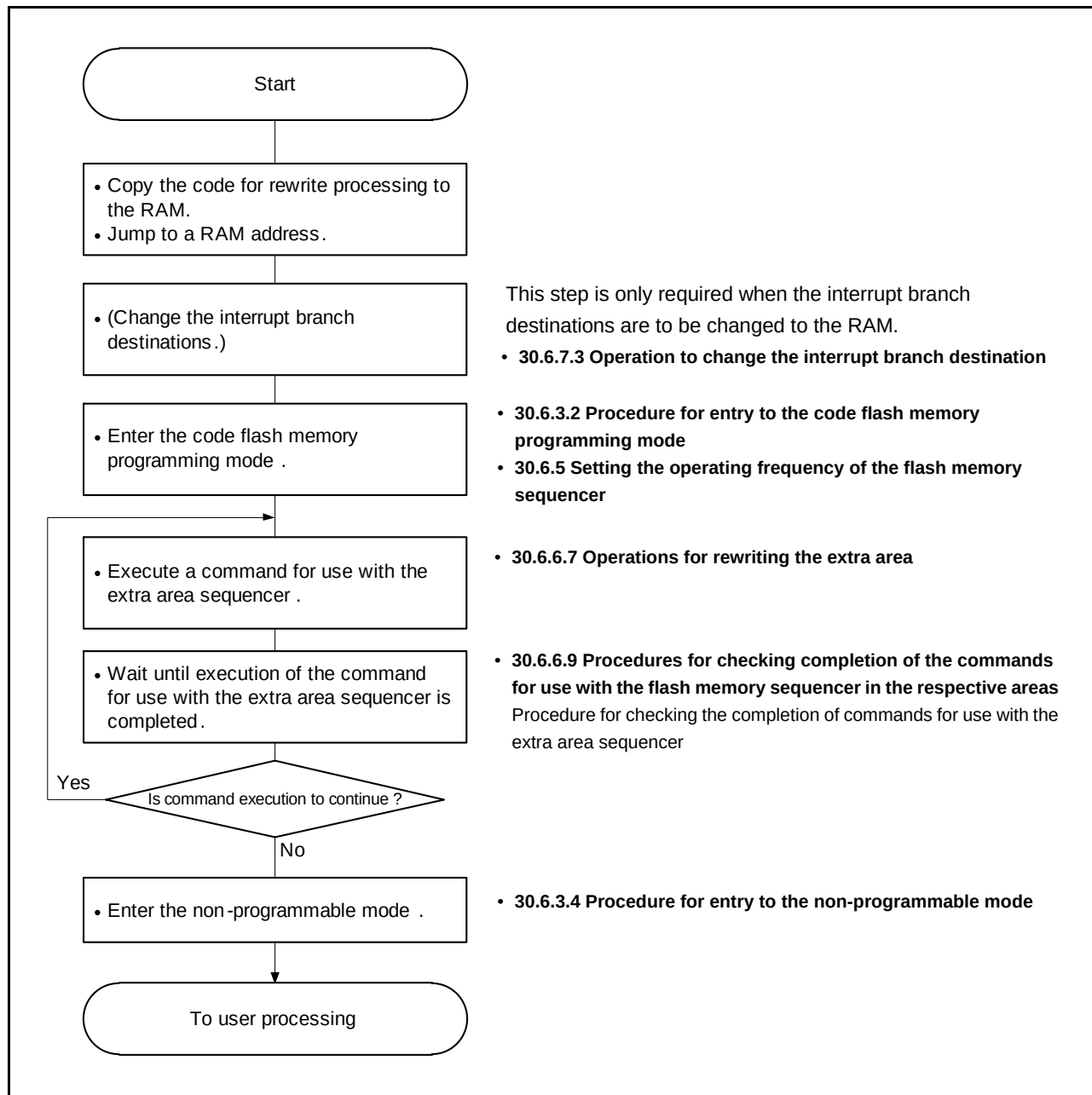
Figure 30 - 28 Flow of Executing the Commands to Rewrite the Data Flash Memory Area



30.6.8.3 Example of executing the commands to rewrite the extra area

Figure 30 - 29 shows the flow of executing the commands to rewrite the extra area.

Figure 30 - 29 Flow of Executing the Commands to Rewrite the Extra Area



30.6.9 Notes on self-programming

- (1) Rewriting the code flash memory or extra area

To rewrite the code flash memory or extra area, place the code or values in the RAM.

- (2) Precondition for manipulating the data flash memory area

Before manipulating the data flash memory area, set the DFLEN bit of the data flash control register (DFLCTL) to 1 (enabling access to the data flash memory).

- (3) Execution of programs during rewriting of the flash memory

The flash memory sequencer is used to control rewriting of the flash memory during self-programming.

In the flash memory control modes where rewriting of the flash memory is enabled, reference to the flash memory to be manipulated is not possible.

- In code flash memory programming mode, reference to the code flash memory is not possible. Accordingly, in code flash memory programming mode, copy the user program that is to be executed from the ROM (code flash memory) and its data for reference to the RAM in advance so that the program can be executed and reference to the data in the RAM is possible.
- In data flash memory programming mode, reference to the data flash memory is not possible. Accordingly, in data flash memory programming mode, copy data that are for reference to the RAM in advance so that reference to the data in the RAM is possible.

- (4) Specifying the range of unavailable area

Specify the range of blank checking and block erasure within the range of code flash memory area or data flash memory area. Do not specify any unavailable area or both the code flash memory area and data flash memory area including an unavailable area.

30.7 Boot Swap Function

If rewriting in the boot area through self-programming failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

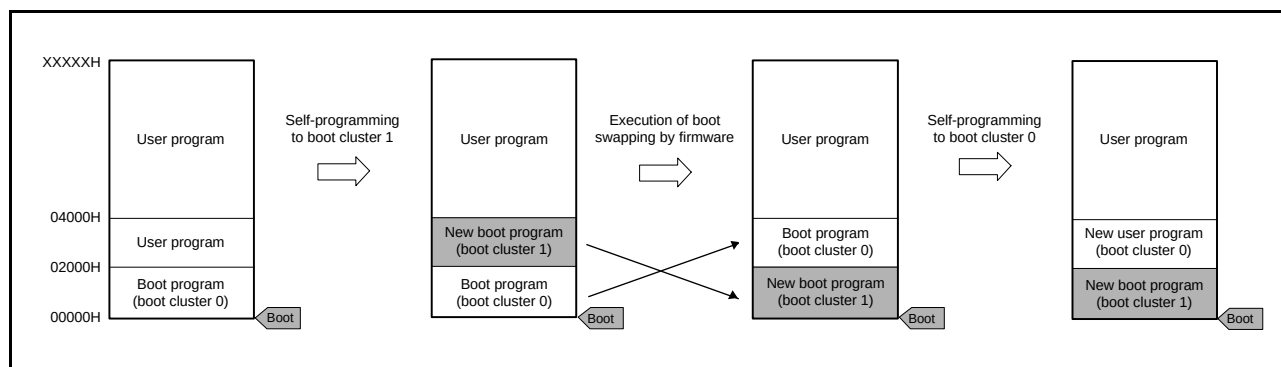
The boot swap function is used to avoid this problem.

When the boot area is set to boot cluster 0, write a new boot program to boot cluster 1 through self-programming before erasing boot cluster 0. After the boot program has been successfully written to boot cluster 1, change the boot area from boot cluster 0 to boot cluster 1 by self-programming to use boot cluster 1 as the boot area. After that, erase the data in boot cluster 0, and then write data to boot cluster 0.

As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 when the program is reset and started next.

Note A boot cluster is an 8-Kbyte area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 30 - 30 Boot Swap Function

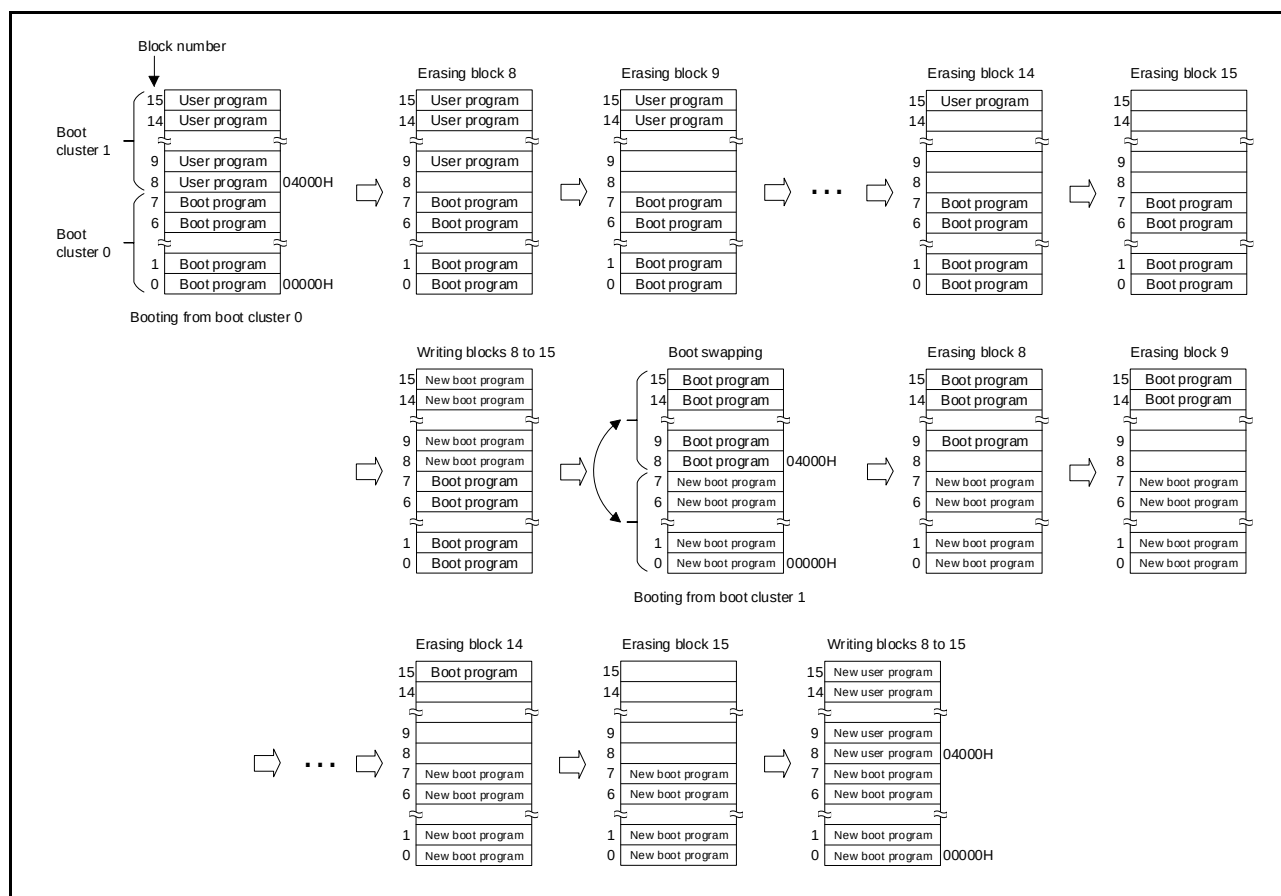


In the example of the above figure, the boot areas before and after boot swapping are as follows.

Boot cluster 0: Boot area before boot swapping

Boot cluster 1: Boot area after boot swapping

Figure 30 - 31 Example of Executing Boot Swapping



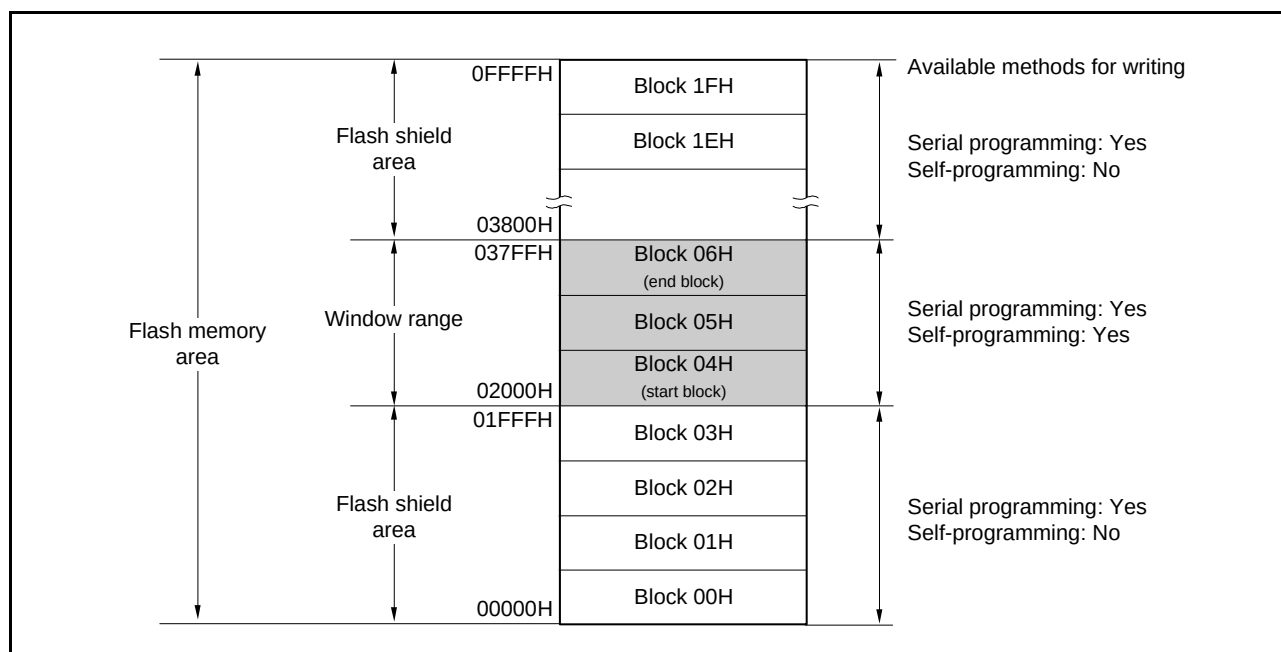
30.8 Flash Shield Window Function

The flash shield window function is provided as a security function which disables writing to and erasing of the selected flash memory shield area. This function is only effective for self-programming.

The flash memory shield area is selectable as either the area inside or areas outside the range specified as the window. The window range is set by specifying the blocks where it starts and ends. The flash memory shield area can be set or changed during both serial programming and self-programming.

Writing to and erasing of the flash memory shield area are disabled during self-programming. During serial programming, however, the flash memory shield area can also be written and erased.

Figure 30 - 32 Flash Shield Window Setting Example (Target Devices: R7F102GGE, Start Block Number: 04H, End Block Number: 06H, FSWC: 1)



Caution 1. If the non-programmable area of the boot area overlaps with the flash shield window range, disabling of rewriting the boot area takes priority.

Caution 2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 30 - 12 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming Condition	Window Range Setting/Change Method	Command to be Executed	
		Block erase	Write
Self-programming	Specify the window start block number and the (end block number + 1) block number (following the end block) in the flash shield window setting area using the self-programming.	Block erasure is not possible inside the flash memory shield area.	Writing is not possible inside the flash memory shield area.
Serial programming	Specify the start block number in the window range and the end block number in the window range on GUI of dedicated flash memory programmer, etc.	Block erasure is also possible inside the flash memory shield area.	Writing is also possible inside the flash memory shield area.

30.9 Security Settings

The RL78 microcontroller supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed by serial programming or self-programming.

- Disabling block erasure
Execution of the block erase command for a specific block in the code flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.
- Disabling writing
Execution of the write command for entire blocks in the code flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
After the setting to prohibit writing has been made, releasing the setting by the Security Release command is enabled by a reset.
- Disabling rewriting the boot area
Execution of the block erase command and write command for the boot area (00000H to 00FFFH) in the code flash memory is prohibited.
- Disabling connection to the programmer and on-chip debugger
Connection to a dedicated flash memory programmer and on-chip debugger is prohibited.
A dedicated flash memory programmer and on-chip debugger cannot be used to manipulate the flash memory.
- Enabling programmer connection ID authentication
Authentication for an arbitrary 10-byte ID code is enabled when connecting to a dedicated flash memory programmer. The 10-byte ID area is 000C4H to 000CDH^{Note}. If the ID does not match when using serial programming, the dedicated flash memory programmer cannot be used to manipulate the flash memory.

Block erasure, writing, and rewriting the boot area are enabled by the default setting at the time of shipment. Security can be set by serial programming and self-programming. Each security setting can be used in combination.

Table 30 - 13 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Note The 10-byte ID code area for the programmer connection ID is shared with the security ID code for on-chip debugging.

Caution The security function of the dedicated flash programmer does not support self-programming.

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **30.8 Flash Shield Window Function** for detail).

Table 30 - 13 Relationship between Enabling the Security Function and Commands

(1) During serial programming

Valid Security	Command to be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks cannot be erased.	Data can be written. Note
Prohibition of writing	Blocks can be erased.	Data cannot be written.
Prohibition of rewriting the boot area	The boot area cannot be erased	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks cannot be erased.	Data cannot be written.
Success in authentication with programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.
Failure in authentication with programmer connection ID authentication enabled	Blocks cannot be erased.	Data cannot be written.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

(2) During self-programming

Valid Security	Command to be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks can be erased.	Data can be written.
Prohibition of writing		
Prohibition of rewriting the boot area	The boot area cannot be erased.	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks can be erased.	Data can be written.
Programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **30.8 Flash Shield Window Function** for detail).

Table 30 - 14 Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set via GUI of dedicated flash memory programmer, etc.	Disabling the setting is not possible.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

Caution The setting to prohibit writing can only be released when the settings to prohibit erasing blocks and rewriting the boot area are not made and the code and data flash memory areas are blank. However, if connection for serial programming is prohibited due to the setting to prohibit connection to the programmer and on-chip debugger or to enable programmer connection ID authentication, releasing the setting to prohibit writing is not possible because serial programming cannot be executed.

(2) During self-programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set by using self-programming.	Disabling the setting is not possible.
Prohibition of writing		Disabling the setting is not possible during self-programming. Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

30.10 Data Flash Memory

30.10.1 Overview of the data flash memory

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the Renesas flash driver. For details, refer to the RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual (R20UT4830EJ).
- The data flash memory can also be rewritten to through serial programming using a dedicated flash memory programmer or an external device.
- The data flash memory can be erased in 1-block (256-byte) units.
- The data flash memory can be accessed only in 8-bit units.
- The data flash memory can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- The data flash memory area is exclusively to be used for data, so executing instructions from the data flash memory is prohibited.
- Accessing the data flash memory is prohibited while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is prohibited while rewriting the data flash memory.
- Transition to the STOP mode is prohibited while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (HIOSTOP = 0), and the Renesas flash driver should be re-executed after 5 μ s have elapsed.

Remark For rewriting the code flash memory via a user program, see **30.6 Self-Programming**.

30.10.2 Procedure for accessing the data flash memory

The data flash memory is stopped after release from a reset. For access to this memory, follow the procedure below to make the required initial setting.

<1> Set 1 in bit 0 (DFLEN) of the data flash control register (DFLCTL).

<2> Wait for the setup time by using a software timer, etc.

The setup time depends on the operating mode of the flash memory in terms of the main clock selection.

<Setup time for each operating mode of the flash memory>

- HS (high-speed main) mode: 250 ns
- LS (low-speed main) mode: 250 ns
- LP (low-power main) mode: 0 ns

<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is prohibited during the setup time.

Caution 2. Transition to the STOP mode is prohibited during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.

Caution 3. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (HIOSTOP = 0), and the Renesas flash driver should be re-executed after 5 μ s have elapsed.

Caution 4. When switching the CPU clock between the main system clock and sub system clock, accessing the data flash memory is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

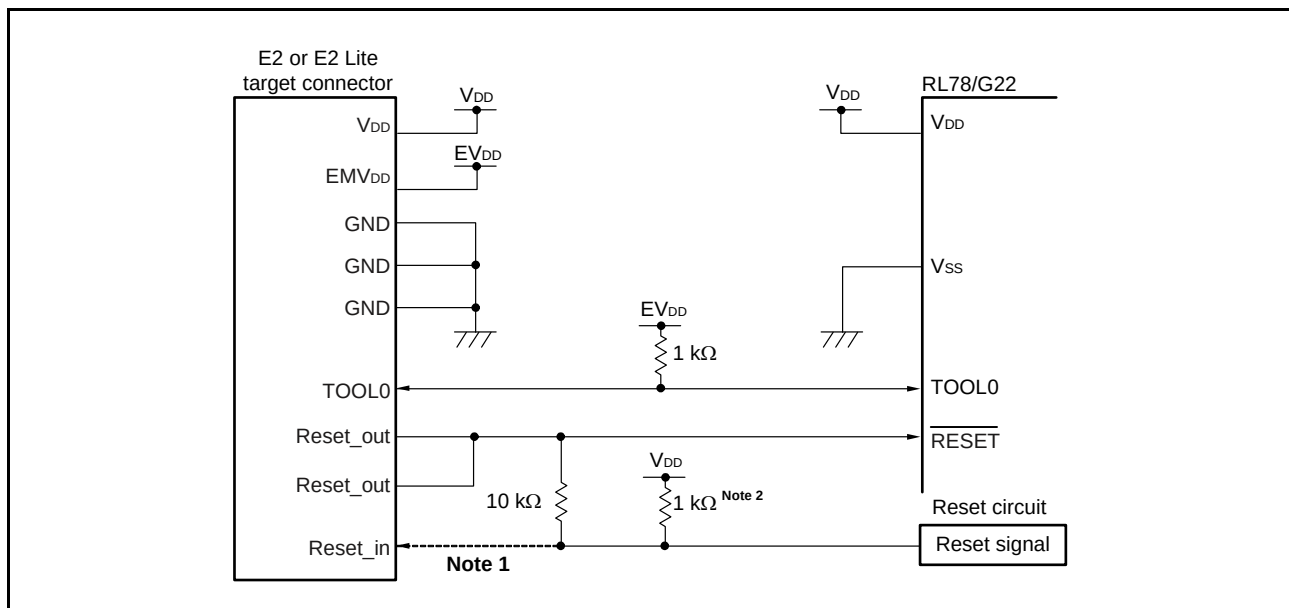
Section 31 On-chip Debugging

31.1 Connection between the E2 or E2 Lite and RL78/G22

On-chip debugging is handled by connecting the RL78 microcontroller and host machine through the E2 or E2 Lite. Pins V_{DD} , $\overline{RESET}$, TOOL0, and V_{SS} are used for connection to the E2 or E2 Lite. Serial communications are handled through the TOOL0 pin as a single-line UART connection. For details and usage notes on the circuit to make the connection, refer to E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (R20UT1994).

Caution RL78 microcontrollers have on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Figure 31 - 1 Example of Connection between the E2 or E2 Lite and RL78/G22



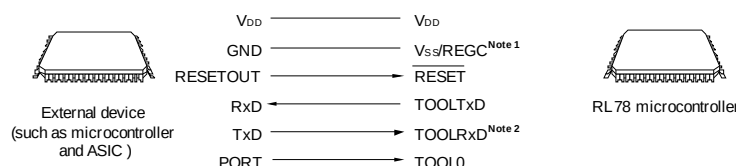
Note 1. The connection shown as a broken line is not required for serial programming.

Note 2. If the reset circuit on the target system does not have a buffer so the reset signal is only generated through resistors and capacitors, this pull-up resistor is not required.

Caution This circuit diagram is an example where the reset signal is output from an N-channel open-drain buffer with an output resistance no greater than 100 Ω .

31.2 Connection between the External Device that Incorporates UART and RL78/G22

On-board communications between an external device (a microcontroller or ASIC) that is connected to the RL78 microcontroller via a UART and the host machine is possible. Pins VDD, RESET, TOOL0, Vss, TOOLTxD, and TOOLRxD are used for the communications. Communications between the external device and the RL78 microcontroller is established by serial communications using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller. For details and usage notes on the circuit to make the connection, refer to RL78 Debugging Functions Using the Serial Port (R20AN0632EJ0100).



Note 1. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μ F).

Note 2. Set the port pin with which TOOLRxD is multiplexed as an input. The input to the input buffer must also be enabled by using the PDIDISx register.

31.3 Security Settings for On-Chip Debugging

To protect against third parties reading the contents of memory, on-chip debugging includes the following functionality.

- Disabling of connection between the RL78 microcontroller and the programmer or on-chip debugger (see **30.9 Security Settings in Section 30 Flash Memory**).
- On-chip debugging control bits in the flash memory at 000C3H (see **Section 29 Option Bytes**)
- An area in the range from 000C4H to 000CDH to hold the security ID code for on-chip debugging. ^{Note 1}

Table 31 - 1 On-Chip Debug Security ID

Address	Security ID Code for On-Chip Debugging
000C4H to 000CDH	Any 10-byte ID code ^{Note 2}
040C4H to 040CDH	

Note 1. The area to hold the security ID code for use in on-chip debugging is also used to hold the ID code for the programmer connection ID when a programmer is to be used.

Note 2. The setting FFFFFFFFFFFFFFFFFFH is not allowed.

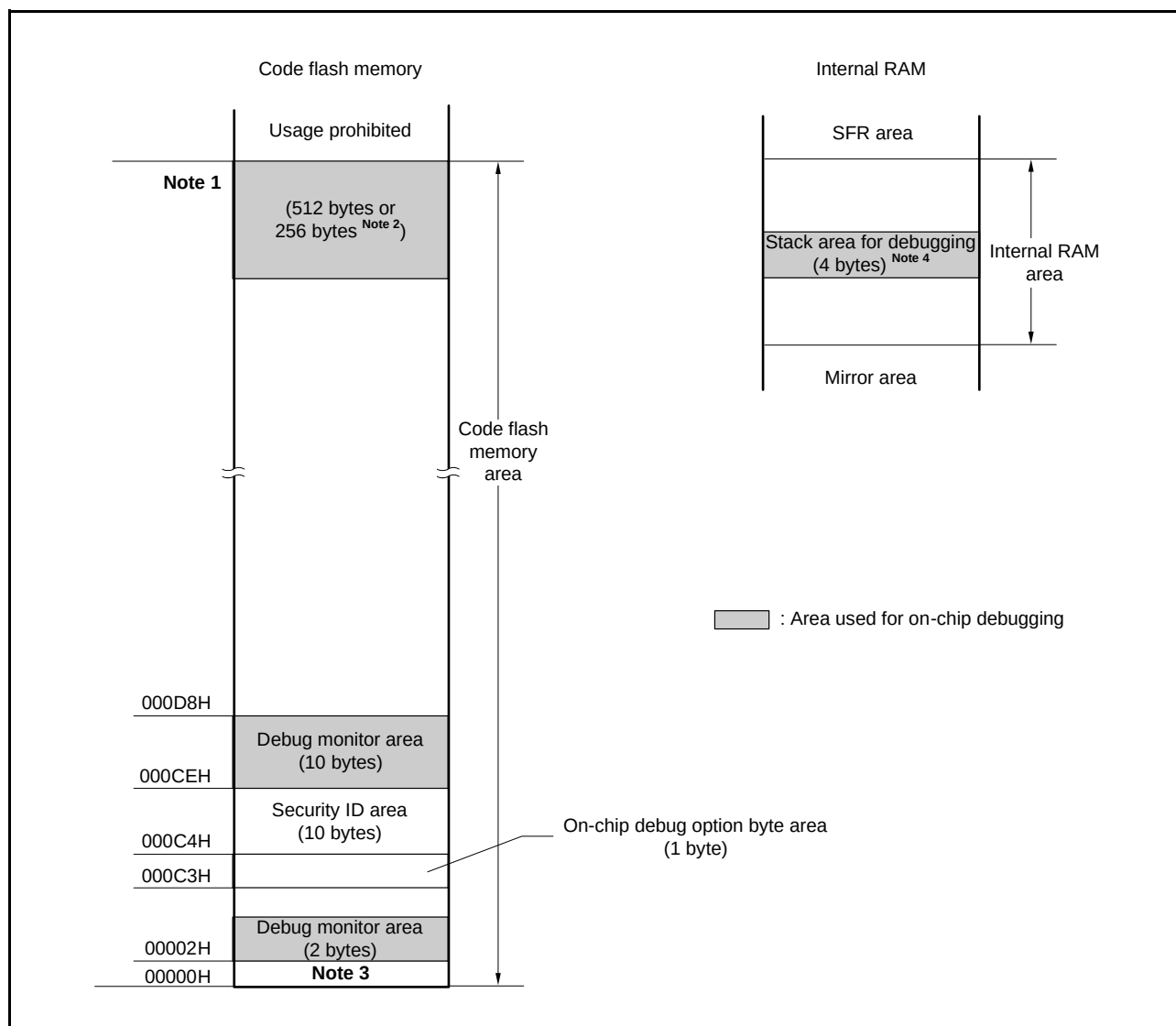
31.4 Allocation of Memory Spaces to User Resources

Allocation of memory spaces to user resources is required before communications between the RL78 microcontroller and E2 or E2 Lite, and on-chip debugging, can proceed. If you are using an assembler or compiler from Renesas Electronics, you can use linker options to allocate the memory spaces.

(1) Allocation of memory spaces to the user program

The shaded areas in **Figure 31 - 2** are reserved for the monitor program for debugging, and user programs and data cannot be allocated to these areas. When using on-chip debugging, ensure that nothing is allocated to these areas so that they can be secured for on-chip debugging. Also ensure that the contents of these areas are not modified by the user program.

Figure 31 - 2 Memory Spaces Allocated for Use by the Monitor Program for Debugging



Note 1. The address depends on the products as shown below.

Products	Address of Note 1
R7F102GxC (x = 4, 6, 7, 8, A to C, and E to G)	07FFFH
R7F102GxE (x = 4, 6, 7, 8, A to C, and E to G)	0FFFFH

Note 2. When the realtime RAM monitor (RRM) and dynamic memory modification (DMM) are not to be used, the size of this area is 256 bytes.

Note 3. During debugging, the reset vector is relocated to the address of the monitor program.

Note 4. Since this area is allocated immediately below the portion of the main stack area that is currently in use, the address range of this area depends on the amount of the stack in use other than for debugging. Accordingly, four additional bytes are required for the entire stack area. In the case of self-programming, this is a 12-byte area, so 12 additional bytes are required.

Section 32 BCD Correction Circuit

32.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit. The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/subtracting the BCD correction result register (BCDADJ).

32.2 Register for Controlling the BCD Correction Circuit

The following register is used to control the BCD correction circuit.

- BCD correction result register (BCDADJ)

32.2.1 BCD correction result register (BCDADJ)

The BCDADJ register holds correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand. The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

The value of this register following a reset is undefined.

Figure 32 - 1 Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH
 After reset: Undefined
 R/W: R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

32.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

- (1) Addition: Obtaining the result of the binary addition of BCD values and BCD values as a BCD value
 - <1> The BCD code value to which addition is performed is stored in the A register.
 - <2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).
 - <3> Decimal correction is performed by adding in binary the values of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

Examples are shown below.

Example 1: $99 + 89 = 188$

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
MOV A, #99H ; <1>	99H	—	—	—
ADD A, #89H ; <2>	22H	1	1	66H
ADD A, !BCDADJ ; <3>	88H	1	0	—

Example 2: $85 + 15 = 100$

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
MOV A, #85H ; <1>	85H	—	—	—
ADD A, #15H ; <2>	9AH	0	0	66H
ADD A, !BCDADJ ; <3>	00H	1	1	—

Example 3: $80 + 80 = 160$

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
MOV A, #80H ; <1>	80H	—	—	—
ADD A, #80H ; <2>	00H	1	0	60H
ADD A, !BCDADJ ; <3>	60H	1	0	—

- (2) Subtraction: Obtaining the result of the binary subtraction of BCD values and BCD values as a BCD value
- <1> The BCD code value from which subtraction is performed is stored in the A register.
 - <2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).
 - <3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: $91 - 52 = 39$

Instruction		A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #91H ; <1>	91H	—	—	—
SUB	A, #52H ; <2>	3FH	0	1	06H
SUB	A, !BCDADJ ; <3>	39H	0	0	—

Section 33 Instruction Set

This section lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

33.1 Conventions Used in Operation List

33.1.1 Operand identifiers and specification methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, \$, \$!, [], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- \$: 8-bit relative address specification
- \$!: 16-bit relative address specification
- []: Indirect address specification
- ES: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, \$, \$!, [], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 33 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only Note) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only Note)
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note)
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3 - 5 List of Special Function Registers (SFRs)** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)** for the symbols of the extended special function registers.

33.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 33 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X _H , X _L X _S , X _H , X _L	16-bit registers: X _H = higher 8 bits, X _L = lower 8 bits 20-bit registers: X _S = (bits 19 to 16), X _H = (bits 15 to 8), X _L = (bits 7 to 0)
^	Logical product (AND)
∨	Logical sum (OR)
⊕	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

33.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 33 - 3 Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
x	Set/cleared according to the result
R	Previously saved value is restored

33.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 33 - 4 Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES: [HL]	11H	8BH	—	—	—

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

33.2 Operation List

Table 33 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	—	$r \leftarrow \text{byte}$			
		PSW, #byte	3	3	—	$\text{PSW} \leftarrow \text{byte}$	x	x	x
		CS, #byte	3	1	—	$\text{CS} \leftarrow \text{byte}$			
		ES, #byte	2	1	—	$\text{ES} \leftarrow \text{byte}$			
		!addr16, #byte	4	1	—	$(\text{addr16}) \leftarrow \text{byte}$			
		ES:!addr16, #byte	5	2	—	$(\text{ES}, \text{addr16}) \leftarrow \text{byte}$			
		saddr, #byte	3	1	—	$(\text{saddr}) \leftarrow \text{byte}$			
		sfr, #byte	3	1	—	$\text{sfr} \leftarrow \text{byte}$			
		[DE+byte], #byte	3	1	—	$(\text{DE} + \text{byte}) \leftarrow \text{byte}$			
		ES:[DE+byte], #byte	4	2	—	$((\text{ES}, \text{DE}) + \text{byte}) \leftarrow \text{byte}$			
		[HL+byte], #byte	3	1	—	$(\text{HL} + \text{byte}) \leftarrow \text{byte}$			
		ES:[HL+byte], #byte	4	2	—	$((\text{ES}, \text{HL}) + \text{byte}) \leftarrow \text{byte}$			
		[SP+byte], #byte	3	1	—	$(\text{SP} + \text{byte}) \leftarrow \text{byte}$			
		word[B], #byte	4	1	—	$(\text{B} + \text{word}) \leftarrow \text{byte}$			
		ES:word[B], #byte	5	2	—	$((\text{ES}, \text{B}) + \text{word}) \leftarrow \text{byte}$			
		word[C], #byte	4	1	—	$(\text{C} + \text{word}) \leftarrow \text{byte}$			
		ES:word[C], #byte	5	2	—	$((\text{ES}, \text{C}) + \text{word}) \leftarrow \text{byte}$			
		word[BC], #byte	4	1	—	$(\text{BC} + \text{word}) \leftarrow \text{byte}$			
		ES:word[BC], #byte	5	2	—	$((\text{ES}, \text{BC}) + \text{word}) \leftarrow \text{byte}$			
		A, r ^{Note 3}	1	1	—	$\text{A} \leftarrow r$			
		r, A ^{Note 3}	1	1	—	$r \leftarrow \text{A}$			
		A, PSW	2	1	—	$\text{A} \leftarrow \text{PSW}$			
		PSW, A	2	3	—	$\text{PSW} \leftarrow \text{A}$	x	x	x
		A, CS	2	1	—	$\text{A} \leftarrow \text{CS}$			
		CS, A	2	1	—	$\text{CS} \leftarrow \text{A}$			
		A, ES	2	1	—	$\text{A} \leftarrow \text{ES}$			
		ES, A	2	1	—	$\text{ES} \leftarrow \text{A}$			
		A, !addr16	3	1	4	$\text{A} \leftarrow (\text{addr16})$			
		A, ES:!addr16	4	2	5	$\text{A} \leftarrow (\text{ES}, \text{addr16})$			
		!addr16, A	3	1	—	$(\text{addr16}) \leftarrow \text{A}$			
		ES:!addr16, A	4	2	—	$(\text{ES}, \text{addr16}) \leftarrow \text{A}$			
		A, saddr	2	1	—	$\text{A} \leftarrow (\text{saddr})$			
		saddr, A	2	1	—	$(\text{saddr}) \leftarrow \text{A}$			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, sfr	2	1	—	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	—	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	—	$(\text{DE}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (\text{ES}, \text{DE})$			
		ES:[DE], A	2	2	—	$(\text{ES}, \text{DE}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	—	$(\text{HL}) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (\text{ES}, \text{HL})$			
		ES:[HL], A	2	2	—	$(\text{ES}, \text{HL}) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE+byte], A	2	1	—	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{DE}) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((\text{ES}, \text{DE}) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow (\text{HL} + \text{byte})$			
		[HL+byte], A	2	1	—	$(\text{HL} + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{HL}) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((\text{ES}, \text{HL}) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow (\text{SP} + \text{byte})$			
		[SP+byte], A	2	1	—	$(\text{SP} + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (\text{B} + \text{word})$			
		word[B], A	3	1	—	$(\text{B} + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((\text{ES}, \text{B}) + \text{word})$			
		ES:word[B], A	4	2	—	$((\text{ES}, \text{B}) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (\text{C} + \text{word})$			
		word[C], A	3	1	—	$(\text{C} + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((\text{ES}, \text{C}) + \text{word})$			
		ES:word[C], A	4	2	—	$((\text{ES}, \text{C}) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (\text{BC} + \text{word})$			
		word[BC], A	3	1	—	$(\text{BC} + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((\text{ES}, \text{BC}) + \text{word})$			
		ES:word[BC], A	4	2	—	$((\text{ES}, \text{BC}) + \text{word}) \leftarrow A$			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	$A \leftarrow (HL + B)$			
		[HL+B], A	2	1	—	$(HL + B) \leftarrow A$			
		A, ES:[HL+B]	3	2	5	$A \leftarrow ((ES, HL) + B)$			
		ES:[HL+B], A	3	2	—	$((ES, HL) + B) \leftarrow A$			
		A, [HL+C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL+C], A	2	1	—	$(HL + C) \leftarrow A$			
		A, ES:[HL+C]	3	2	5	$A \leftarrow ((ES, HL) + C)$			
		ES:[HL+C], A	3	2	—	$((ES, HL) + C) \leftarrow A$			
		X, !addr16	3	1	4	$X \leftarrow (addr16)$			
		X, ES:!addr16	4	2	5	$X \leftarrow (ES, addr16)$			
		X, saddr	2	1	—	$X \leftarrow (saddr)$			
		B, !addr16	3	1	4	$B \leftarrow (addr16)$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, addr16)$			
		B, saddr	2	1	—	$B \leftarrow (saddr)$			
		C, !addr16	3	1	4	$C \leftarrow (addr16)$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, addr16)$			
		C, saddr	2	1	—	$C \leftarrow (saddr)$			
		ES, saddr	3	1	—	$ES \leftarrow (saddr)$			
	XCH	A, r ^{Note 3}	1 (r = X) 2 (other than r = X)	1	—	$A \longleftrightarrow r$			
		A, !addr16	4	2	—	$A \longleftrightarrow (addr16)$			
		A, ES:!addr16	5	3	—	$A \longleftrightarrow (ES, addr16)$			
		A, saddr	3	2	—	$A \longleftrightarrow (saddr)$			
		A, sfr	3	2	—	$A \longleftrightarrow sfr$			
		A, [DE]	2	2	—	$A \longleftrightarrow (DE)$			
		A, ES:[DE]	3	3	—	$A \longleftrightarrow (ES, DE)$			
		A, [HL]	2	2	—	$A \longleftrightarrow (HL)$			
		A, ES:[HL]	3	3	—	$A \longleftrightarrow (ES, HL)$			
		A, [DE+byte]	3	2	—	$A \longleftrightarrow (DE + \text{byte})$			
		A, ES:[DE+byte]	4	3	—	$A \longleftrightarrow ((ES, DE) + \text{byte})$			
		A, [HL+byte]	3	2	—	$A \longleftrightarrow (HL + \text{byte})$			
		A, ES:[HL+byte]	4	3	—	$A \longleftrightarrow ((ES, HL) + \text{byte})$			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	—	$A \longleftrightarrow (HL + B)$			
		A, ES:[HL+B]	3	3	—	$A \longleftrightarrow ((ES, HL) + B)$			
		A, [HL+C]	2	2	—	$A \longleftrightarrow (HL + C)$			
		A, ES:[HL+C]	3	3	—	$A \longleftrightarrow ((ES, HL) + C)$			
	ONEB	A	1	1	—	$A \leftarrow 01H$			
		X	1	1	—	$X \leftarrow 01H$			
		B	1	1	—	$B \leftarrow 01H$			
		C	1	1	—	$C \leftarrow 01H$			
		!addr16	3	1	—	$(addr16) \leftarrow 01H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 01H$			
		saddr	2	1	—	$(saddr) \leftarrow 01H$			
	CLRB	A	1	1	—	$A \leftarrow 00H$			
		X	1	1	—	$X \leftarrow 00H$			
		B	1	1	—	$B \leftarrow 00H$			
		C	1	1	—	$C \leftarrow 00H$			
		!addr16	3	1	—	$(addr16) \leftarrow 00H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 00H$			
		saddr	2	1	—	$(saddr) \leftarrow 00H$			
	MOVS	[HL+byte], X	3	1	—	$(HL + byte) \leftarrow X$	x		x
		ES:[HL+byte], X	4	2	—	$(ES, HL + byte) \leftarrow X$	x		x
16-bit data transfer	MOVW	rp, #word	3	1	—	$rp \leftarrow word$			
		saddrp, #word	4	1	—	$(saddrp) \leftarrow word$			
		sfrp, #word	4	1	—	$sfrp \leftarrow word$			
		AX, rp ^{Note 3}	1	1	—	$AX \leftarrow rp$			
		rp, AX ^{Note 3}	1	1	—	$rp \leftarrow AX$			
		AX, !addr16	3	1	4	$AX \leftarrow (addr16)$			
		!addr16, AX	3	1	—	$(addr16) \leftarrow AX$			
		AX, ES:!addr16	4	2	5	$AX \leftarrow (ES, addr16)$			
		ES:!addr16, AX	4	2	—	$(ES, addr16) \leftarrow AX$			
		AX, saddrp	2	1	—	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	1	—	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	1	—	$AX \leftarrow sfrp$			
		sfrp, AX	2	1	—	$sfrp \leftarrow AX$			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	$AX \leftarrow (DE)$			
		[DE], AX	1	1	—	$(DE) \leftarrow AX$			
		AX, ES:[DE]	2	2	5	$AX \leftarrow (ES, DE)$			
		ES:[DE], AX	2	2	—	$(ES, DE) \leftarrow AX$			
		AX, [HL]	1	1	4	$AX \leftarrow (HL)$			
		[HL], AX	1	1	—	$(HL) \leftarrow AX$			
		AX, ES:[HL]	2	2	5	$AX \leftarrow (ES, HL)$			
		ES:[HL], AX	2	2	—	$(ES, HL) \leftarrow AX$			
		AX, [DE+byte]	2	1	4	$AX \leftarrow (DE + \text{byte})$			
		[DE+byte], AX	2	1	—	$(DE + \text{byte}) \leftarrow AX$			
		AX, ES:[DE+byte]	3	2	5	$AX \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE+byte], AX	3	2	—	$((ES, DE) + \text{byte}) \leftarrow AX$			
		AX, [HL+byte]	2	1	4	$AX \leftarrow (HL + \text{byte})$			
		[HL+byte], AX	2	1	—	$(HL + \text{byte}) \leftarrow AX$			
		AX, ES:[HL+byte]	3	2	5	$AX \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL+byte], AX	3	2	—	$((ES, HL) + \text{byte}) \leftarrow AX$			
		AX, [SP+byte]	2	1	—	$AX \leftarrow (SP + \text{byte})$			
		[SP+byte], AX	2	1	—	$(SP + \text{byte}) \leftarrow AX$			
		AX, word[B]	3	1	4	$AX \leftarrow (B + \text{word})$			
		word[B], AX	3	1	—	$(B + \text{word}) \leftarrow AX$			
		AX, ES:word[B]	4	2	5	$AX \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], AX	4	2	—	$((ES, B) + \text{word}) \leftarrow AX$			
		AX, word[C]	3	1	4	$AX \leftarrow (C + \text{word})$			
		word[C], AX	3	1	—	$(C + \text{word}) \leftarrow AX$			
		AX, ES:word[C]	4	2	5	$AX \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], AX	4	2	—	$((ES, C) + \text{word}) \leftarrow AX$			
		AX, word[BC]	3	1	4	$AX \leftarrow (BC + \text{word})$			
		word[BC], AX	3	1	—	$(BC + \text{word}) \leftarrow AX$			
		AX, ES:word[BC]	4	2	5	$AX \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], AX	4	2	—	$((ES, BC) + \text{word}) \leftarrow AX$			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	$BC \leftarrow (addr16)$			
		BC, ES:!addr16	4	2	5	$BC \leftarrow (ES, addr16)$			
		DE, !addr16	3	1	4	$DE \leftarrow (addr16)$			
		DE, ES:!addr16	4	2	5	$DE \leftarrow (ES, addr16)$			
		HL, !addr16	3	1	4	$HL \leftarrow (addr16)$			
		HL, ES:!addr16	4	2	5	$HL \leftarrow (ES, addr16)$			
		BC, saddrp	2	1	—	$BC \leftarrow (saddrp)$			
		DE, saddrp	2	1	—	$DE \leftarrow (saddrp)$			
		HL, saddrp	2	1	—	$HL \leftarrow (saddrp)$			
	XCHW	AX, rp ^{Note 3}	1	1	—	$AX \longleftrightarrow rp$			
	ONEW	AX	1	1	—	$AX \leftarrow 0001H$			
		BC	1	1	—	$BC \leftarrow 0001H$			
	CLRW	AX	1	1	—	$AX \leftarrow 0000H$			
		BC	1	1	—	$BC \leftarrow 0000H$			
8-bit operation	ADD	A, #byte	2	1	—	$A, CY \leftarrow A + \text{byte}$	x	x	x
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) + \text{byte}$	x	x	x
		A, r ^{Note 4}	2	1	—	$A, CY \leftarrow A + r$	x	x	x
		r, A	2	1	—	$r, CY \leftarrow r + A$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A + (addr16)$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, addr16)$	x	x	x
		A, saddr	2	1	—	$A, CY \leftarrow A + (saddr)$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL)$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL)$	x	x	x
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte})$	x	x	x
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte})$	x	x	x
		A, [HL+B]	2	1	4	$A, CY \leftarrow A + (HL + B)$	x	x	x
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B)$	x	x	x
		A, [HL+C]	2	1	4	$A, CY \leftarrow A + (HL + C)$	x	x	x
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C)$	x	x	x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Note 4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	—	$A, CY \leftarrow A + \text{byte} + CY$	x	x	x
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) + \text{byte} + CY$	x	x	x
		A, r ^{Note 3}	2	1	—	$A, CY \leftarrow A + r + CY$	x	x	x
		r, A	2	1	—	$r, CY \leftarrow r + A + CY$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16}) + CY$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16}) + CY$	x	x	x
		A, saddr	2	1	—	$A, CY \leftarrow A + (saddr) + CY$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL) + CY$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL) + CY$	x	x	x
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte}) + CY$	x	x	x
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte}) + CY$	x	x	x
		A, [HL+B]	2	1	4	$A, CY \leftarrow A + (HL + B) + CY$	x	x	x
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B) + CY$	x	x	x
		A, [HL+C]	2	1	4	$A, CY \leftarrow A + (HL + C) + CY$	x	x	x
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C) + CY$	x	x	x
	SUB	A, #byte	2	1	—	$A, CY \leftarrow A - \text{byte}$	x	x	x
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) - \text{byte}$	x	x	x
		A, r ^{Note 3}	2	1	—	$A, CY \leftarrow A - r$	x	x	x
		r, A	2	1	—	$r, CY \leftarrow r - A$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16})$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (ES, \text{addr16})$	x	x	x
		A, saddr	2	1	—	$A, CY \leftarrow A - (saddr)$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL)$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES, HL)$	x	x	x
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte})$	x	x	x
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + \text{byte})$	x	x	x
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (HL + B)$	x	x	x
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + B)$	x	x	x
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (HL + C)$	x	x	x
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + C)$	x	x	x

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	—	$A, CY \leftarrow A - \text{byte} - CY$	x	x	x
		saddr, #byte	3	2	—	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	x	x	x
		A, r ^{Note 3}	2	1	—	$A, CY \leftarrow A - r - CY$	x	x	x
		r, A	2	1	—	$r, CY \leftarrow r - A - CY$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (ES, \text{addr16}) - CY$	x	x	x
		A, saddr	2	1	—	$A, CY \leftarrow A - (saddr) - CY$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL) - CY$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES, HL) - CY$	x	x	x
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte}) - CY$	x	x	x
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + \text{byte}) - CY$	x	x	x
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (HL + B) - CY$	x	x	x
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + B) - CY$	x	x	x
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (HL + C) - CY$	x	x	x
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + C) - CY$	x	x	x
	AND	A, #byte	2	1	—	$A \leftarrow A \wedge \text{byte}$	x		
		saddr, #byte	3	2	—	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	x		
		A, r ^{Note 3}	2	1	—	$A \leftarrow A \wedge r$	x		
		r, A	2	1	—	$r \leftarrow r \wedge A$	x		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	x		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (ES:\text{addr16})$	x		
		A, saddr	2	1	—	$A \leftarrow A \wedge (saddr)$	x		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (HL)$	x		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (ES:HL)$	x		
		A, [HL+byte]	2	1	4	$A \leftarrow A \wedge (HL + \text{byte})$	x		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + \text{byte})$	x		
		A, [HL+B]	2	1	4	$A \leftarrow A \wedge (HL + B)$	x		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + B)$	x		
		A, [HL+C]	2	1	4	$A \leftarrow A \wedge (HL + C)$	x		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + C)$	x		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	—	$A \leftarrow A \vee \text{byte}$	x		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	x		
		A, r ^{Note 3}	2	1	—	$A \leftarrow A \vee r$	x		
		r, A	2	1	—	$r \leftarrow r \vee A$	x		
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	x		
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	x		
		A, saddr	2	1	—	$A \leftarrow A \vee (\text{saddr})$	x		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	x		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	x		
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	x		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	x		
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	x		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	x		
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	x		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	x		
	XOR	A, #byte	2	1	—	$A \leftarrow A \nabla \text{byte}$	x		
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	x		
		A, r ^{Note 3}	2	1	—	$A \leftarrow A \nabla r$	x		
		r, A	2	1	—	$r \leftarrow r \nabla A$	x		
		A, !addr16	3	1	4	$A \leftarrow A \nabla (\text{addr16})$	x		
		A, ES:!addr16	4	2	5	$A \leftarrow A \nabla (\text{ES:addr16})$	x		
		A, saddr	2	1	—	$A \leftarrow A \nabla (\text{saddr})$	x		
		A, [HL]	1	1	4	$A \leftarrow A \nabla (\text{HL})$	x		
		A, ES:[HL]	2	2	5	$A \leftarrow A \nabla (\text{ES:HL})$	x		
		A, [HL+byte]	2	1	4	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	x		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + \text{byte})$	x		
		A, [HL+B]	2	1	4	$A \leftarrow A \nabla (\text{HL} + B)$	x		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + B)$	x		
		A, [HL+C]	2	1	4	$A \leftarrow A \nabla (\text{HL} + C)$	x		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + C)$	x		

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	—	A - byte	x	x	x
		!addr16, #byte	4	1	4	(addr16) - byte	x	x	x
		ES:!addr16, #byte	5	2	5	(ES:addr16) - byte	x	x	x
		saddr, #byte	3	1	—	(saddr) - byte	x	x	x
		A, r ^{Note 3}	2	1	—	A - r	x	x	x
		r, A	2	1	—	r - A	x	x	x
		A, !addr16	3	1	4	A - (addr16)	x	x	x
		A, ES:!addr16	4	2	5	A - (ES:addr16)	x	x	x
		A, saddr	2	1	—	A - (saddr)	x	x	x
		A, [HL]	1	1	4	A - (HL)	x	x	x
		A, ES:[HL]	2	2	5	A - (ES:HL)	x	x	x
		A, [HL+byte]	2	1	4	A - (HL + byte)	x	x	x
		A, ES:[HL+byte]	3	2	5	A - ((ES:HL) + byte)	x	x	x
		A, [HL+B]	2	1	4	A - (HL + B)	x	x	x
		A, ES:[HL+B]	3	2	5	A - ((ES:HL) + B)	x	x	x
		A, [HL+C]	2	1	4	A - (HL + C)	x	x	x
		A, ES:[HL+C]	3	2	5	A - ((ES:HL) + C)	x	x	x
	CMP0	A	1	1	—	A - 00H	x	0	0
		X	1	1	—	X - 00H	x	0	0
		B	1	1	—	B - 00H	x	0	0
		C	1	1	—	C - 00H	x	0	0
		!addr16	3	1	4	(addr16) - 00H	x	0	0
		ES:!addr16	4	2	5	(ES:addr16) - 00H	x	0	0
		saddr	2	1	—	(saddr) - 00H	x	0	0
	CMPS	X, [HL+byte]	3	1	4	X - (HL + byte)	x	x	x
		X, ES:[HL+byte]	4	2	5	X - ((ES:HL) + byte)	x	x	x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	—	$AX, CY \leftarrow AX + \text{word}$	x	x	x
		AX, AX	1	1	—	$AX, CY \leftarrow AX + AX$	x	x	x
		AX, BC	1	1	—	$AX, CY \leftarrow AX + BC$	x	x	x
		AX, DE	1	1	—	$AX, CY \leftarrow AX + DE$	x	x	x
		AX, HL	1	1	—	$AX, CY \leftarrow AX + HL$	x	x	x
		AX, !addr16	3	1	4	$AX, CY \leftarrow AX + (\text{addr16})$	x	x	x
		AX, ES:!addr16	4	2	5	$AX, CY \leftarrow AX + (\text{ES:addr16})$	x	x	x
		AX, saddrp	2	1	—	$AX, CY \leftarrow AX + (\text{saddrp})$	x	x	x
		AX, [HL+byte]	3	1	4	$AX, CY \leftarrow AX + (\text{HL} + \text{byte})$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX, CY \leftarrow AX + ((\text{ES:HL}) + \text{byte})$	x	x	x
	SUBW	AX, #word	3	1	—	$AX, CY \leftarrow AX - \text{word}$	x	x	x
		AX, BC	1	1	—	$AX, CY \leftarrow AX - BC$	x	x	x
		AX, DE	1	1	—	$AX, CY \leftarrow AX - DE$	x	x	x
		AX, HL	1	1	—	$AX, CY \leftarrow AX - HL$	x	x	x
		AX, !addr16	3	1	4	$AX, CY \leftarrow AX - (\text{addr16})$	x	x	x
		AX, ES:!addr16	4	2	5	$AX, CY \leftarrow AX - (\text{ES:addr16})$	x	x	x
		AX, saddrp	2	1	—	$AX, CY \leftarrow AX - (\text{saddrp})$	x	x	x
		AX, [HL+byte]	3	1	4	$AX, CY \leftarrow AX - (\text{HL} + \text{byte})$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX, CY \leftarrow AX - ((\text{ES:HL}) + \text{byte})$	x	x	x
	CMPW	AX, #word	3	1	—	$AX - \text{word}$	x	x	x
		AX, BC	1	1	—	$AX - BC$	x	x	x
		AX, DE	1	1	—	$AX - DE$	x	x	x
		AX, HL	1	1	—	$AX - HL$	x	x	x
		AX, !addr16	3	1	4	$AX - (\text{addr16})$	x	x	x
		AX, ES:!addr16	4	2	5	$AX - (\text{ES:addr16})$	x	x	x
		AX, saddrp	2	1	—	$AX - (\text{saddrp})$	x	x	x
		AX, [HL+byte]	3	1	4	$AX - (\text{HL} + \text{byte})$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX - ((\text{ES:HL}) + \text{byte})$	x	x	x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumulate	MULU	X	1	1	—	$AX \leftarrow A \times X$			
	MULHU		3	2	—	$BCAX \leftarrow AX \times BC$ (unsigned)			
	MULH		3	2	—	$BCAX \leftarrow AX \times BC$ (signed)			
	DIVHU		3	9	—	AX (quotient), DE (remainder) $\leftarrow AX \div DE$ (unsigned)			
	DIVWU		3	17	—	$BCAX$ (quotient), $HLDE$ (remainder) $\leftarrow BCAX \div HLDE$ (unsigned)			
	MACHU		3	3	—	$MACR \leftarrow MACR + AX \times BC$ (unsigned)		x	x
	MACH		3	3	—	$MACR \leftarrow MACR + AX \times BC$ (signed)		x	x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- V. 1.01.00 and later versions of CC-RL (Renesas Electronics compiler), for both C and assembly language source code
- V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code
- Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code
- LLVM RL78 (CyberTHOR compiler), for both C and C++ language source code
- GNURL78 (CyberTHOR compiler), for both C and C++ language source code

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 33 - 5 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/decrement	INC	r	1	1	—	$r \leftarrow r + 1$	x	x	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$	x	x	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$	x	x	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) + 1$	x	x	
		[HL+byte]	3	2	—	$(HL + byte) \leftarrow (HL + byte) + 1$	x	x	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$	x	x	
	DEC	r	1	1	—	$r \leftarrow r - 1$	x	x	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$	x	x	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$	x	x	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) - 1$	x	x	
		[HL+byte]	3	2	—	$(HL + byte) \leftarrow (HL + byte) - 1$	x	x	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$	x	x	
	INCW	rp	1	1	—	$rp \leftarrow rp + 1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) + 1$			
		[HL+byte]	3	2	—	$(HL + byte) \leftarrow (HL + byte) + 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$			
	DECW	rp	1	1	—	$rp \leftarrow rp - 1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) - 1$			
		[HL+byte]	3	2	—	$(HL + byte) \leftarrow (HL + byte) - 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$			
Shift	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_m - 1 \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			x
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_m - 1 \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			x
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_m - 1, A_0 \leftarrow 0) \times cnt$			x
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_m - 1, B_0 \leftarrow 0) \times cnt$			x
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_m - 1, C_0 \leftarrow 0) \times cnt$			x
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_m - 1, AX_0 \leftarrow 0) \times cnt$			x
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_m - 1, BC_0 \leftarrow 0) \times cnt$			x
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_m - 1 \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			x
	SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_m - 1 \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. cnt indicates the bit shift count.

Table 33 - 5 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	$(CY, A7 \leftarrow A0, A_{m-1} \leftarrow A_m) \times 1$			x
	ROL	A, 1	2	1	—	$(CY, A0 \leftarrow A7, A_{m+1} \leftarrow A_m) \times 1$			x
	RORC	A, 1	2	1	—	$(CY \leftarrow A0, A7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			x
	ROLC	A, 1	2	1	—	$(CY \leftarrow A7, A0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			x
	ROLWC	AX,1	2	1	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			x
		BC,1	2	1	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			x
Bit manipulate	MOV1	CY, A.bit	2	1	—	$CY \leftarrow A.bit$			x
		A.bit, CY	2	1	—	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	—	$CY \leftarrow PSW.bit$			x
		PSW.bit, CY	3	4	—	$PSW.bit \leftarrow CY$	x	x	
		CY, saddr.bit	3	1	—	$CY \leftarrow (saddr).bit$			x
		saddr.bit, CY	3	2	—	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	—	$CY \leftarrow sfr.bit$			x
		sfr.bit, CY	3	2	—	$sfr.bit \leftarrow CY$			
		CY,[HL].bit	2	1	4	$CY \leftarrow (HL).bit$			x
		[HL].bit, CY	2	2	—	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			x
		ES:[HL].bit, CY	3	3	—	$(ES, HL).bit \leftarrow CY$			
	AND1	CY, A.bit	2	1	—	$CY \leftarrow CY \wedge A.bit$			x
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \wedge PSW.bit$			x
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \wedge (saddr).bit$			x
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \wedge sfr.bit$			x
		CY,[HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			x
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			x
	OR1	CY, A.bit	2	1	—	$CY \leftarrow CY \vee A.bit$			x
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \vee PSW.bit$			x
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \vee (saddr).bit$			x
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \vee sfr.bit$			x
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			x
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			x

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	$CY \leftarrow CY \nabla A.bit$			x
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \nabla PSW.bit$			x
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \nabla (saddr).bit$			x
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \nabla sfr.bit$			x
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \nabla (HL).bit$			x
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \nabla (ES, HL).bit$			x
	SET1	A.bit	2	1	—	$A.bit \leftarrow 1$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 1$	x	x	x
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 1$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 1$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 1$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 1$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 1$			
	CLR1	A.bit	2	1	—	$A.bit \leftarrow 0$			
		PSW.bit	3	4	—	$PSW.bit \leftarrow 0$	x	x	x
		laddr16.bit	4	2	—	$(addr16).bit \leftarrow 0$			
		ES:laddr16.bit	5	3	—	$(ES, addr16).bit \leftarrow 0$			
		saddr.bit	3	2	—	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	—	$sfr.bit \leftarrow 0$			
		[HL].bit	2	2	—	$(HL).bit \leftarrow 0$			
		ES:[HL].bit	3	3	—	$(ES, HL).bit \leftarrow 0$			
	SET1	CY	2	1	—	$CY \leftarrow 1$			1
	CLR1	CY	2	1	—	$CY \leftarrow 0$			0
	NOT1	CY	2	1	—	$CY \leftarrow \overline{CY}$			x

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	CALL	rp	2	3	—	(SP - 2) ← (PC + 2) _S , (SP - 3) ← (PC + 2) _H , (SP - 4) ← (PC + 2) _L , PC ← CS, rp, SP ← SP - 4			
		\$!addr20	3	3	—	(SP - 2) ← (PC + 3) _S , (SP - 3) ← (PC + 3) _H , (SP - 4) ← (PC + 3) _L , PC ← PC + 3 + jdisp16, SP ← SP - 4			
		!addr16	3	3	—	(SP - 2) ← (PC + 3) _S , (SP - 3) ← (PC + 3) _H , (SP - 4) ← (PC + 3) _L , PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	—	(SP - 2) ← (PC + 4) _S , (SP - 3) ← (PC + 4) _H , (SP - 4) ← (PC + 4) _L , PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	—	(SP - 2) ← (PC + 2) _S , (SP - 3) ← (PC + 2) _H , (SP - 4) ← (PC + 2) _L , PC _S ← 0000, PC _H ← (0000, addr5 + 1), PC _L ← (0000, addr5), SP ← SP - 4			
	BRK	—	2	5	—	(SP - 1) ← PSW, (SP - 2) ← (PC + 2) _S , (SP - 3) ← (PC + 2) _H , (SP - 4) ← (PC + 2) _L , PC _S ← 0000, PC _H ← (0007FH), PC _L ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	—	1	6	—	PC _L ← (SP), PC _H ← (SP + 1), PC _S ← (SP + 2), SP ← SP + 4			
	RETI	—	2	6	—	PC _L ← (SP), PC _H ← (SP + 1), PC _S ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	—	2	6	—	PC _L ← (SP), PC _H ← (SP + 1), PC _S ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	—	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow 00H, SP \leftarrow SP - 2$			
		rp	1	1	—	$(SP - 1) \leftarrow r_{pH}, (SP - 2) \leftarrow r_{pL}, SP \leftarrow SP - 2$			
	POP	PSW	2	3	—	$PSW \leftarrow (SP + 1), SP \leftarrow SP + 2$	R	R	R
		rp	1	1	—	$r_{pL} \leftarrow (SP), r_{pH} \leftarrow (SP + 1), SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	1	—	$SP \leftarrow word$			
		SP, AX	2	1	—	$SP \leftarrow AX$			
		AX, SP	2	1	—	$AX \leftarrow SP$			
		HL, SP	3	1	—	$HL \leftarrow SP$			
		BC, SP	3	1	—	$BC \leftarrow SP$			
		DE, SP	3	1	—	$DE \leftarrow SP$			
	ADDW	SP, #byte	2	1	—	$SP \leftarrow SP + byte$			
	SUBW	SP, #byte	2	1	—	$SP \leftarrow SP - byte$			
Unconditional branch	BR	AX	2	3	—	$PC \leftarrow CS, AX$			
		\$addr20	2	3	—	$PC \leftarrow PC + 2 + jdisp8$			
		\$!addr20	3	3	—	$PC \leftarrow PC + 3 + jdisp16$			
		!addr16	3	3	—	$PC \leftarrow 0000, addr16$			
		!!addr20	4	3	—	$PC \leftarrow addr20$			
Conditional branch	BC	\$addr20	2	2/4Note 3	—	$PC \leftarrow PC + 2 + jdisp8$ if CY = 1			
	BNC	\$addr20	2	2/4Note 3	—	$PC \leftarrow PC + 2 + jdisp8$ if CY = 0			
	BZ	\$addr20	2	2/4Note 3	—	$PC \leftarrow PC + 2 + jdisp8$ if Z = 1			
	BNZ	\$addr20	2	2/4Note 3	—	$PC \leftarrow PC + 2 + jdisp8$ if Z = 0			
	BH	\$addr20	3	2/4Note 3	—	$PC \leftarrow PC + 3 + jdisp8$ if $(Z \vee CY) = 0$			
	BNH	\$addr20	3	2/4Note 3	—	$PC \leftarrow PC + 3 + jdisp8$ if $(Z \vee CY) = 1$			
	BT	saddr.bit, \$addr20	4	3/5Note 3	—	$PC \leftarrow PC + 4 + jdisp8$ if (saddr).bit = 1			
		sfr.bit, \$addr20	4	3/5Note 3	—	$PC \leftarrow PC + 4 + jdisp8$ if sfr.bit = 1			
		A.bit, \$addr20	3	3/5Note 3	—	$PC \leftarrow PC + 3 + jdisp8$ if A.bit = 1			
		PSW.bit, \$addr20	4	3/5Note 3	—	$PC \leftarrow PC + 4 + jdisp8$ if PSW.bit = 1			
		[HL].bit, \$addr20	3	3/5Note 3	6/7	$PC \leftarrow PC + 3 + jdisp8$ if (HL).bit = 1			
		ES:[HL].bit, \$addr20	4	4/6Note 3	7/8	$PC \leftarrow PC + 4 + jdisp8$ if (ES, HL).bit = 1			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks "when condition is not met/when condition is met".

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 5 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, \$addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, \$addr20	3	3/5Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, \$addr20	4	4/6Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, \$addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, \$addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	x	x	x
		[HL].bit, \$addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, \$addr20	4	4/6Note 3	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	—	2	1	—	Next instruction skip if CY = 1			
	SKNC	—	2	1	—	Next instruction skip if CY = 0			
	SKZ	—	2	1	—	Next instruction skip if Z = 1			
	SKNZ	—	2	1	—	Next instruction skip if Z = 0			
	SKH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	—	2	1	—	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SELNote 4	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks "when condition is not met/when condition is met".

Note 4. n indicates the number of register banks (n = 0 to 3)

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Section 34 Electrical Characteristics for TA = -40 to +105°C

34.1 Absolute Maximum Ratings

Item	Symbols	Conditions		Ratings	Unit
Supply voltage	V _{DD}			-0.5 to +6.5	V
REGC pin input voltage	V _I REGC	REGC		-0.3 to +2.1 and -0.3 to V _{DD} + 0.3 ^{Note 1}	V
Input voltage	V _{I1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140, P146, P147		-0.3 to V _{DD} + 0.3 ^{Note 2}	V
	V _{I2}	P60 to P63 (N-ch open-drain)		-0.3 to +6.5	V
	V _{I3}	P20 to P27, P121 to P124, P137, EXCLK, EXCLKS, <u>RESET</u>		-0.3 to V _{DD} + 0.3 ^{Note 2}	V
Output voltage	V _{O1}	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P130, P140, P146, P147		-0.3 to V _{DD} + 0.3 ^{Note 2}	V
	V _{O2}	P20 to P27, P121, P122		-0.3 to V _{DD} + 0.3 ^{Note 2}	V
Analog input voltage	V _{AI1}	ANI16 to ANI19		-0.3 to V _{DD} + 0.3 and -0.3 to AV _{REFP} + 0.3 Notes 2, 3	V
	V _{AI2}	ANI0 to ANI7		-0.3 to V _{DD} + 0.3 and -0.3 to AV _{REFP} + 0.3 Notes 2, 3	V
High-level output current	I _{OH1}	Per pin	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140, P146, P147	-40	mA
		Total of all pins -170 mA	P00, P01, P40, P41, P120, P130, P140	-70	mA
			P10 to P17, P30, P31, P50, P51, P70 to P75, P146, P147	-100	mA
	I _{OH2}	Per pin	P20 to P27, P121, P122	-0.5	mA
		Total of all pins		-2	mA
Low-level output current	I _{OL1}	Per pin	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140, P146, P147	40	mA
		Total of all pins 170 mA	P00, P01, P40, P41, P120, P130, P140	70	mA
			P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75, P146, P147	100	mA
	I _{OL2}	Per pin	P20 to P27, P121, P122	1	mA
		Total of all pins		5	mA
Ambient operating temperature	T _A	In normal operation mode		-40 to +105	°C
		In flash memory programming mode			
Storage temperature	T _{stg}			-65 to +150	°C

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). The listed value is the absolute maximum rating of the REGC pin. Only use the capacitor connection. Do not apply a specific voltage to this pin.

Note 2. This voltage must be no higher than 6.5 V.

Note 3. The voltage on a pin in use for A/D conversion must not exceed AV_{REFP} + 0.3.

(Caution and Remarks are listed on the next page.)

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark 1. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. AVREFP refers to the positive reference voltage of the A/D converter.

Remark 3. The reference voltage is V_{SS}.

34.2 Characteristics of the Oscillators

34.2.1 Characteristics of the X1 oscillator

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
X1 clock oscillation allowable input cycle time Note	Ceramic resonator/ crystal resonator		0.05		1	μs

Note The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. See AC Characteristics for instruction execution time.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS) after having sufficiently evaluated the oscillation stabilization time with the resonator to be used.

34.2.2 Characteristics of the XT1 oscillator

(TA = -40 to +105°C, 2.4 V ≤ VDD ≤ 5.5 V (16- to 36-pin products), 1.6 V ≤ VDD ≤ 5.5 V (40- to 48-pin products), VSS = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
XT1 clock oscillation frequency (fXT) Note	Crystal resonator			32.768		kHz

Note The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. See AC Characteristics for instruction execution time.

34.2.3 Characteristics of the On-chip Oscillators

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
High-speed on-chip oscillator clock frequency	f _{IH}				1		32	MHz
High-speed on-chip oscillator clock frequency accuracy ^{Note 1}		HIPREC = 1	+85 to +105°C	1.8 V ≤ V _{DD} ≤ 5.5 V	-2.0		+2.0	%
				1.6 V ≤ V _{DD} ≤ 5.5 V	-6.0		+6.0	%
			-20 to +85°C	1.8 V ≤ V _{DD} ≤ 5.5 V	-1.0		+1.0	%
				1.6 V ≤ V _{DD} ≤ 5.5 V	-5.0		+5.0	%
			-40 to -20°C	1.8 V ≤ V _{DD} ≤ 5.5 V	-1.5		+1.5	%
				1.6 V ≤ V _{DD} ≤ 5.5 V	-5.5		+5.5	%
		HIPREC = 0 ^{Note 4}			-15		0	%
High-speed on-chip oscillator clock correction resolution						0.05		%
Middle-speed on-chip oscillator clock frequency ^{Note 2}	f _{IM}				1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy ^{Note 1}					-12		+12	%
Middle-speed on-chip oscillator clock correction resolution						0.15		%
Middle-speed on-chip oscillator frequency temperature coefficient							±0.17 ^{Note 3}	%/°C
Low-speed on-chip oscillator clock frequency ^{Note 2}	f _{IL}					32.768		kHz
Low-speed on-chip oscillator clock frequency accuracy ^{Note 1}					-15		+15	%
Low-speed on-chip oscillator clock correction resolution						0.3		%
Low-speed on-chip oscillator frequency temperature coefficient							±0.21 ^{Note 3}	%/°C

Note 1. The accuracy values were obtained in testing of this product.

Note 2. The listed values only indicate the characteristics of the oscillators. See AC Characteristics for instruction execution time.

Note 3. Guaranteed by characterization results.

Note 4. The listed condition applies when the setting of the FRQSEL3 bit is 1.

34.3 DC Characteristics

34.3.1 Characteristics of Pins

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/5)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Allowable high-level output current Note 1	IOH1	Per pin for P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140, P146, P147	1.6 V ≤ VDD ≤ 5.5 V		-10.0 Note 2	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (when duty ≤ 70% Note 3)	4.0 V ≤ VDD ≤ 5.5 V		-55.0 Note 4	mA
			2.7 V ≤ VDD < 4.0 V		-10.0	mA
			1.8 V ≤ VDD < 2.7 V		-5.0	mA
			1.6 V ≤ VDD < 1.8 V		-2.5	mA
		Total of P10 to P17, P30, P31, P50, P51, P70 to P75, P146, P147 (when duty ≤ 70% Note 3)	4.0 V ≤ VDD ≤ 5.5 V		-80.0 Note 5	mA
			2.7 V ≤ VDD < 4.0 V		-19.0	mA
			1.8 V ≤ VDD < 2.7 V		-10.0	mA
			1.6 V ≤ VDD < 1.8 V		-5.0	mA
		Total of all pins (when duty ≤ 70% Note 3)	1.6 V ≤ VDD ≤ 5.5 V		-135.0 Note 6	mA
	IOH2	Per pin for P20 to P27, P121, P122	4.0 V ≤ VDD ≤ 5.5 V		-3.0 Note 2	mA
			2.7 V ≤ VDD < 4.0 V		-1.0 Note 2	mA
			1.8 V ≤ VDD < 2.7 V		-1.0 Note 2	mA
			1.6 V ≤ VDD < 1.8 V		-0.5 Note 2	mA
		Total of all pins (when duty ≤ 70% Note 3)	4.0 V ≤ VDD ≤ 5.5 V		-20.0	mA
			2.7 V ≤ VDD < 4.0 V		-10.0	mA
			1.8 V ≤ VDD < 2.7 V		-5.0	mA
			1.6 V ≤ VDD < 1.8 V		-5.0	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the VDD pin to an output pin.

Note 2. The combination of these and other pins must also not exceed the value for maximum total current.

Note 3. The listed currents apply when the duty cycle is no greater than 70%.

Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

• Total output current from the listed pins = (IOH × 0.7)/(n × 0.01)

Example when n = 80% and IOH = -10.0 mA

Total output current from the listed pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. The maximum value is -30 mA in the products for industrial applications (R7F102Gxx3xxx) with an ambient operating temperature range of 85°C to 105°C.

Note 5. The maximum value is -50 mA in the products for industrial applications (R7F102Gxx3xxx) with an ambient operating temperature range of 85°C to 105°C.

Note 6. The maximum values are respectively -100 mA and -60 mA in the products for industrial applications (R7F102Gxx3xxx) with an ambient operating temperature range of -40°C to 85°C and of 85°C to 105°C.

Caution The following pins are not capable of the output of high-level signals in the N-ch open-drain mode.

P00, P10 to P15, P17, P50, P71, P72, P74, and P120

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/5)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Allowable low-level output current ^{Note 1}	IOL1	Per pin for P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140, P146, P147			20.0 ^{Note 2}	mA
		Per pin for P60 to P63			15.0 ^{Note 2}	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (when duty ≤ 70% ^{Note 3})	4.0 V ≤ VDD ≤ 5.5 V		70.0 ^{Note 4}	mA
			2.7 V ≤ VDD < 4.0 V		15.0	mA
			1.8 V ≤ VDD < 2.7 V		9.0	mA
			1.6 V ≤ VDD < 1.8 V		4.5	mA
		Total of P10 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75, P146, P147 (when duty ≤ 70% ^{Note 3})	4.0 V ≤ VDD ≤ 5.5 V		80.0 ^{Note 4}	mA
			2.7 V ≤ VDD < 4.0 V		35.0	mA
			1.8 V ≤ VDD < 2.7 V		20.0	mA
			1.6 V ≤ VDD < 1.8 V		10.0	mA
		Total of all pins (when duty ≤ 70% ^{Note 3})			150.0 ^{Note 5}	mA
	IOL2	Per pin for P20 to P27, P121, P122	4.0 V ≤ VDD ≤ 5.5 V		8.5 ^{Note 2}	mA
			2.7 V ≤ VDD < 4.0 V		1.5 ^{Note 2}	mA
			1.8 V ≤ VDD < 2.7 V		0.6 ^{Note 2}	mA
			1.6 V ≤ VDD < 1.8 V		0.4 ^{Note 2}	mA
		Total of all pins (when duty ≤ 70% ^{Note 3})	4.0 V ≤ VDD ≤ 5.5 V		20	mA
			2.7 V ≤ VDD < 4.0 V		20	mA
			1.8 V ≤ VDD < 2.7 V		15	mA
			1.6 V ≤ VDD < 1.8 V		10	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the VSS pin.

Note 2. The combination of these and other pins must also not exceed the value for maximum total current.

Note 3. The listed currents apply when the duty cycle is no greater than 70%.

Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

•Total output current from the listed pins = (IOL × 0.7)/(n × 0.01)

Example when n = 80% and IOL = 10.0 mA

Total output current from the listed pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. The maximum value is 40 mA in the products for industrial applications (R7F102Gxx3xxxC) with an ambient operating temperature range of 85°C to 105°C.

Note 5. The maximum value is 80 mA in the products for industrial applications (R7F102Gxx3xxxC) with an ambient operating temperature range of 85°C to 105°C.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(3/5)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	VIH1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140, P146, P147	Normal input buffer	0.8 VDD		VDD	V
	VIH2	P01, P10, P11, P13 to P17	TTL input buffer 4.0 V ≤ VDD ≤ 5.5 V	2.2		VDD	V
			TTL input buffer 3.3 V ≤ VDD < 4.0 V	2.0		VDD	V
			TTL input buffer 1.6 V ≤ VDD < 3.3 V	1.5		VDD	V
	VIH3	P20 to P27		0.7 VDD		VDD	V
	VIH4	P60 to P63		0.7 VDD		6.0	V
	VIH5	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 VDD		VDD	V
Input voltage, low	VIL1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140, P146, P147	Normal input buffer	0		0.2 VDD	V
	VIL2	P01, P10, P11, P13 to P17	TTL input buffer 4.0 V ≤ VDD ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ VDD < 4.0 V	0		0.5	V
			TTL input buffer 1.6 V ≤ VDD < 3.3 V	0		0.32	V
	VIL3	P20 to P27		0		0.3 VDD	V
	VIL4	P60 to P63		0		0.3 VDD	V
	VIL5	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 VDD	V

Caution The maximum value of VIH of pins P00, P10 to P15, P17, P50, P71, P72, P74, and P120 is VDD, even in the N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(4/5)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output voltage, high	VOH1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140, P146, P147	4.0 V ≤ VDD ≤ 5.5 V, IOH1 = -10.0 mA	VDD - 1.5		V
			4.0 V ≤ VDD ≤ 5.5 V, IOH1 = -3.0 mA	VDD - 0.7		V
			2.7 V ≤ VDD ≤ 5.5 V, IOH1 = -2.0 mA	VDD - 0.6		V
			1.8 V ≤ VDD ≤ 5.5 V, IOH1 = -1.5 mA	VDD - 0.5		V
			1.6 V ≤ VDD < 5.5 V, IOH1 = -1.0 mA	VDD - 0.5		V
	VOH2	P20 to P27, P121, P122	4.0 V ≤ VDD ≤ 5.5 V, IOH2 = -3.0 mA	VDD - 0.7		V
			2.7 V ≤ VDD < 4.0 V, IOH2 = -1.0 mA	VDD - 0.5		V
			1.8 V ≤ VDD < 2.7 V, IOH2 = -1.0 mA	VDD - 0.5		V
			1.6 V ≤ VDD < 1.8 V, IOH2 = -0.5 mA	VDD - 0.5		V
Output voltage, low	VOL1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140, P146, P147	4.0 V ≤ VDD ≤ 5.5 V, IOL1 = 20.0 mA		1.3	V
			4.0 V ≤ VDD ≤ 5.5 V, IOL1 = 8.5 mA		0.7	V
			2.7 V ≤ VDD ≤ 5.5 V, IOL1 = 3.0 mA		0.6	V
			2.7 V ≤ VDD ≤ 5.5 V, IOL1 = 1.5 mA		0.4	V
			1.8 V ≤ VDD ≤ 5.5 V, IOL1 = 0.6 mA		0.4	V
			1.6 V ≤ VDD ≤ 5.5 V, IOL1 = 0.3 mA		0.4	V
	VOL2	P20 to P27, P121, P122	4.0 V ≤ VDD ≤ 5.5 V, IOL2 = 8.5 mA		0.7	V
			2.7 V ≤ VDD < 4.0 V, IOL2 = 1.5 mA		0.5	V
			1.8 V ≤ VDD < 2.7 V, IOL2 = 0.6 mA		0.4	V
			1.6 V ≤ VDD < 1.8 V, IOL2 = 0.4 mA		0.4	V
	VOL3	P60 to P63	4.0 V ≤ VDD ≤ 5.5 V, IOL3 = 15.0 mA		2.0	V
			4.0 V ≤ VDD ≤ 5.5 V, IOL3 = 5.0 mA		0.4	V
			2.7 V ≤ VDD ≤ 5.5 V, IOL3 = 3.0 mA		0.4	V
			1.8 V ≤ VDD ≤ 5.5 V, IOL3 = 2.0 mA		0.4	V
			1.6 V ≤ VDD ≤ 5.5 V, IOL3 = 1.0 mA		0.4	V

Caution P00, P10 to P15, P17, P50, P71, P72, P74, and P120 do not output high-level signals in the N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(5/5)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leakage current, high	LIH1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140, P146, P147	$V_I = V_{DD}$			0.5	μA
	LIH2	P20 to P27, P137, $\overline{RESET}$	$V_I = V_{DD}$			0.5	μA
	LIH3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$			0.5	μA
Input leakage current, low	LIL1	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140, P146, P147	$V_I = V_{SS}$			0.5	μA
	LIL2	P20 to P27, P137, $\overline{RESET}$	$V_I = V_{SS}$			0.5	μA
	LIL3	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$			0.5	μA
On-chip pull-up resistance	Ru	P00, P01, P10 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120 to P122, P140, P146, P147	$V_I = V_{SS}$, In input port	10	20	100	k Ω

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

34.3.2 Characteristics of the supply current

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current Note 1	IDD1	Operating mode	HS (high-speed main) mode	fIH = 32 MHz ^{Note 2}	Basic operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				Normal operation		VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
			LS (low-speed main) mode	fIH = 24 MHz ^{Note 2}	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fIH = 16 MHz ^{Note 2}	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fIM = 4 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.6 V		T.B.D.	T.B.D.	
			LP (low-power main) mode	fIM = 2 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	μA
						VDD = 1.6 V		T.B.D.	T.B.D.	
				fIM = 1 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	μA
						VDD = 1.6 V		T.B.D.	T.B.D.	
			HS (high-speed main) mode	fMX = 20 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
			LS (low-speed main) mode	fMX = 20 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fMX = 20 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fMX = 10 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fMX = 10 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fMX = 8 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	
				fMX = 8 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		T.B.D.	T.B.D.	mA
						VDD = 1.8 V		T.B.D.	T.B.D.	

Note 1. The listed currents are the total currents flowing into VDD, including the input leakage currents flowing when the level of the input pin is fixed to VDD or VSS. The currents in the Max. column include the peripheral operation current, but do not include those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

Note 2. The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. fIH: High-speed on-chip oscillator clock frequency

Remark 2. fIM: Middle-speed on-chip oscillator clock frequency

Remark 3. fMX: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. The typical value for the ambient operating temperature (TA) is 25°C unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

(2/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current Note 1	IDD1	Operating mode	Subsystem clock operation mode	f _{SUB} = 32.768 kHz ^{Note 2} , Low-speed on-chip oscillator operation	Normal operation	T _A = -40°C		T.B.D.	T.B.D.	μA
						T _A = +25°C		T.B.D.	T.B.D.	
						T _A = +50°C		T.B.D.	T.B.D.	
						T _A = +70°C		T.B.D.	T.B.D.	
						T _A = +85°C		T.B.D.	T.B.D.	
						T _A = +105°C		T.B.D.	T.B.D.	
				f _{SUB} = 32.768 kHz ^{Note 3} , Square wave input	Normal operation	T _A = -40°C		T.B.D.	T.B.D.	μA
						T _A = +25°C		T.B.D.	T.B.D.	
						T _A = +50°C		T.B.D.	T.B.D.	
						T _A = +70°C		T.B.D.	T.B.D.	
						T _A = +85°C		T.B.D.	T.B.D.	
						T _A = +105°C		T.B.D.	T.B.D.	
				f _{SUB} = 32.768 kHz ^{Note 3} , Resonator connection	Normal operation	T _A = -40°C		T.B.D.	T.B.D.	μA
						T _A = +25°C		T.B.D.	T.B.D.	
						T _A = +50°C		T.B.D.	T.B.D.	
						T _A = +70°C		T.B.D.	T.B.D.	
						T _A = +85°C		T.B.D.	T.B.D.	
						T _A = +105°C		T.B.D.	T.B.D.	

Note 1. The listed currents are the total currents flowing into V_{DD}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The currents in the Max. column include the peripheral operation current, but do not include those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. They do not include the current flowing into the RTC, 32-bit interval timer, and watchdog timer.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 1, 1). They do not include the currents flowing into the RTC, 32-bit interval timer, and watchdog timer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(3/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current ^{Note 1}	IDD2 ^{Note 2}	HALT mode	HS (high-speed main) mode	f _{IH} = 32 MHz ^{Note 3}	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
			LS (low-speed main) mode	f _{IH} = 24 MHz ^{Note 3}	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{IH} = 16 MHz ^{Note 3}	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{IM} = 4 MHz ^{Note 4}	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.6 V		T.B.D.	T.B.D.	
			LP (low-power main) mode	f _{IM} = 2 MHz ^{Note 4}	VDD = 5.0 V		T.B.D.	T.B.D.	μA
					VDD = 1.6 V		T.B.D.	T.B.D.	
				f _{IM} = 1 MHz ^{Note 4}	VDD = 5.0 V		T.B.D.	T.B.D.	μA
					VDD = 1.6 V		T.B.D.	T.B.D.	
			HS (high-speed main) mode	f _{MX} = 20 MHz ^{Note 5} , Square wave input	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
			LS (low-speed main) mode	f _{MX} = 20 MHz ^{Note 5} , Square wave input	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{MX} = 20 MHz ^{Note 5} , Resonator connection	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{MX} = 10 MHz ^{Note 5} , Square wave input	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{MX} = 10 MHz ^{Note 5} , Resonator connection	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{MX} = 8 MHz ^{Note 5} , Square wave input	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	
				f _{MX} = 8 MHz ^{Note 5} , Resonator connection	VDD = 5.0 V		T.B.D.	T.B.D.	mA
					VDD = 1.8 V		T.B.D.	T.B.D.	

- Note 1.** The listed currents are the total currents flowing into VDD, including the input leakage currents flowing when the level of the input pin is fixed to VDD or VSS. The currents in the Max. column include the peripheral operation current, but do not include those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.
- Note 2.** The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.
- Note 3.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.
- Note 4.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock are stopped.
- Note 5.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency

Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 4. The typical value for the ambient operating temperature (TA) is 25°C unless otherwise specified.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(4/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit	
Supply current Note 1	IDD2 Note 2	HALT mode	Subsystem clock operation mode	fsUB = 32.768 kHzNote 3, Low-speed on-chip oscillator operation	TA = -40°C		T.B.D.	T.B.D.	μA	
					TA = +25°C		T.B.D.	T.B.D.		
					TA = +50°C		T.B.D.	T.B.D.		
					TA = +70°C		T.B.D.	T.B.D.		
					TA = +85°C		T.B.D.	T.B.D.		
					TA = +105°C		T.B.D.	T.B.D.		
				fsUB = 32.768 kHz, Square wave input Note 4	TA = -40°C		T.B.D.	T.B.D.	μA	
					TA = +25°C		T.B.D.	T.B.D.		
					TA = +50°C		T.B.D.	T.B.D.		
					TA = +70°C		T.B.D.	T.B.D.		
					TA = +85°C		T.B.D.	T.B.D.		
					TA = +105°C		T.B.D.	T.B.D.		
			fsUB = 32.768 kHz, Resonator connection Note 5	TA = -40°C		T.B.D.	T.B.D.	μA		
				TA = +25°C		T.B.D.	T.B.D.			
				TA = +50°C		T.B.D.	T.B.D.			
				TA = +70°C		T.B.D.	T.B.D.			
				TA = +85°C		T.B.D.	T.B.D.			
				TA = +105°C		T.B.D.	T.B.D.			
	IDD3	STOP mode	TA = -40°C					T.B.D.	T.B.D.	μA
			TA = +25°C					T.B.D.	T.B.D.	
			TA = +50°C					T.B.D.	T.B.D.	
			TA = +70°C					T.B.D.	T.B.D.	
			TA = +85°C					T.B.D.	T.B.D.	
			TA = +105°C					T.B.D.	T.B.D.	

Note 1. The listed currents are the total currents flowing into VDD, including the input leakage currents flowing when the level of the input pin is fixed to VDD or VSS. The currents in the Max. column include the peripheral operation current, but do not include those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

Note 3. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. They include the currents flowing into the RTC, but do not include those into the 32-bit interval timer and watchdog timer.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped. They include the currents flowing into the RTC, but do not include those into the 32-bit interval timer and watchdog timer.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 1, 1). They include the currents flowing into the RTC, but do not include those into the 32-bit interval timer and watchdog timer.

Remark 1. fIL: Low-speed on-chip oscillator clock frequency

Remark 2. fsUB: Subsystem clock frequency (XT1 clock oscillation frequency)

Peripheral Functions (Common to all products)

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit	
High-speed on-chip oscillator operating current	I _{FIH} Note 1	HIPREC = 1				T.B.D.	—	μA	
		HIPREC = 0				T.B.D.	—	μA	
Middle-speed on-chip oscillator operating current	I _{FIM} Note 1					20	—	μA	
Low-speed on-chip oscillator operating current	I _{FIL} Note 1					0.3	—	μA	
RTC operating current	I _{RTC} Notes 1, 2, 3	f _{RTCCLK} = 32.768 kHz				0.005	—	μA	
		f _{RTCCLK} = 128 Hz				0.002	—	μA	
32-bit interval timer operating current	I _{IT} Notes 1, 2, 4					0.04	—	μA	
Watchdog timer operating current	I _{WDT} Notes 1, 2, 5	f _{IL} = 32.768 kHz (typ.)				0.32	—	μA	
A/D converter operating current	I _{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV _{REFP} = V _{DD} = 5.0 V			1.3	1.7	mA	
			Low voltage mode, AV _{REFP} = V _{DD} = 3.0 V			0.5	0.7	mA	
A/D converter internal reference voltage current	I _{ADREF} Note 1					100	—	μA	
Temperature sensor operating current	I _{TMPS} Note 1					110	—	μA	
LVD operating current	I _{LVD0} Notes 1, 8					0.02	—	μA	
	I _{LVD1} Notes 1, 8					0.02	—	μA	
Self-programming operating current	I _{FSP} Notes 1, 9					2.5	12.2	mA	
Data flash rewrite operating current	I _{BGO} Notes 1, 10					2.5	12.2	mA	
SNOOZE mode sequencer operating current	I _{SMS} Notes 1, 11	f _{IH} = 32 MHz				T.B.D.	—	mA	
		f _{IL} = 32.768 kHz				T.B.D.	—	μA	
SNOOZE operating current	I _{SNOZ} Note 1	f _{IH} =32 MHz	ADC to be in use	The ADC is shifting from the STOP mode to the SNOOZE mode. Note 12		0.6	0.81	mA	
				The ADC is operating in the low-voltage mode. AV _{REFP} = V _{DD} = 3.0 V		1.2	1.56		
			Simplified SPI (CSI)/UART to be in use				0.7	0.92	mA
			SMS Note 14				T.B.D.	—	mA
Low-speed peripheral clock supply current	I _{SXP} Notes 1, 13	RTCLPC = 0				0.22	—	μA	
Operating current of the true random number generator	I _{TRNG} Note 1					1.1	—	mA	

(Notes and Remarks continue on the next page.)

- Note 1.** This current flows into VDD.
- Note 2.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.
- Note 3.** This current flows into the realtime clock (RTC). It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2, and IRTC, when the realtime clock is operating or in the HALT mode. When the low-speed on-chip oscillator is selected, IFIL should be included in the supply current. IDD2 in the subsystem clock operation mode includes the operating current of the realtime clock.
- Note 4.** This current only flows to the 32-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2, and IIT, when the 32-bit interval timer is operating or in the HALT mode. When the low-speed on-chip oscillator is selected, IFIL should be included in the supply current.
- Note 5.** This current only flows to the watchdog timer. It includes the operating current of the low-speed on-chip oscillator. The supply current of the RL78 microcontrollers is the sum of IDD1, IDD2 or IDD3 and IWDI when the watchdog timer is operating.
- Note 6.** This current only flows to the A/D converter. The supply current of the RL78 microcontrollers is the sum of IDD1 or IDD2 and IADC when the A/D converter is operating or in the HALT mode.
- Note 7.** This current flows into AVREFP.
- Note 8.** This current only flows to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of IDD1, IDD2 or IDD3 and ILVD when the LVD circuit is in operation.
- Note 9.** This current only flows during self-programming.
- Note 10.** This current only flows while the data flash memory is being rewritten.
- Note 11.** This current only flows into the SNOOZE mode sequencer. Note that the operating current of the low-speed on-chip oscillator and the XT1 oscillator are not included. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2, and ISMS, when the SNOOZE mode sequencer is operating or in the HALT mode.
- Note 12.** For shift time to the SNOOZE mode, see **20.3.3 SNOOZE mode**.
- Note 13.** This current is added to the supply current in the HALT mode when the setting of RTCLPC is 0 in the STOP mode, or when the setting of RTCLPC is 0 with the subsystem clock X (fsx) selected as the CPU clock, while the subsystem clock X (fsx) is oscillating.
- Note 14.** The listed values apply when the SNOOZE mode sequencer is in normal operation equivalent to IDD1. They do not include the current flowing into the peripheral functions other than the SNOOZE mode sequencer.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{sx}: Subsystem clock X frequency

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. The typical value for the ambient operating temperature (TA) is 25°C unless otherwise specified.

34.4 AC Characteristics

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Instruction cycle (minimum instruction execution time)	TCY	Main system clock (fMAIN) operation	HS (high-speed main) mode	1.8 V ≤ VDD ≤ 5.5 V	0.03125	1	μs
				1.6 V ≤ VDD ≤ 1.8 V	0.25	1	μs
			LS (low-speed main) mode	1.8 V ≤ VDD ≤ 5.5 V	0.04167	1	μs
				1.6 V ≤ VDD ≤ 1.8 V	0.25	1	μs
		LP (low-power main) mode		1.6 V ≤ VDD ≤ 5.5 V	0.5	1	μs
Subsystem clock (fsUB) operation				1.8 V ≤ VDD ≤ 5.5 V	26.041	30.5	μs
In the self-programming mode		HS (high-speed main) mode		1.8 V ≤ VDD ≤ 5.5 V	0.03125	1	μs
				1.6 V ≤ VDD ≤ 1.8 V	0.5	1	μs
		LS (low-speed main) mode		1.8 V ≤ VDD ≤ 5.5 V	0.04167	1	μs
				1.6 V ≤ VDD ≤ 1.8 V	0.5	1	μs
External system clock frequency	fEX	1.8 V ≤ VDD ≤ 5.5 V		1.0		20.0	MHz
		1.6 V ≤ VDD < 1.8 V		1.0		4.0	MHz
	fEXS			32		38.4	kHz
External system clock input high-level width, low-level width	tEXH, tEXL	1.8 V ≤ VDD ≤ 5.5 V		15			ns
		1.6 V ≤ VDD < 1.8 V		120			ns
	tEXHS, tEXLS			13.7			μs
TI00 to TI07 input high-level width, low-level width	tTIH, tTIL			1/fMCK + 10			ns ^{Note}
TO00 to TO07 output frequency	fTO	HS (high-speed main) mode LS (low-speed main) mode		4.0 V ≤ VDD ≤ 5.5 V		16	MHz
				2.7 V ≤ VDD < 4.0 V		8	MHz
				1.8 V ≤ VDD < 2.7 V		4	MHz
				1.6 V ≤ VDD < 1.8 V		2	MHz
		LP (low-power main) mode		1.6 V ≤ VDD ≤ 5.5 V		2	MHz
PCLBUZ0, PCLBUZ1 output frequency	fPCL	HS (high-speed main) mode LS (low-speed main) mode		4.0 V ≤ VDD ≤ 5.5 V		16	MHz
				2.7 V ≤ VDD < 4.0 V		8	MHz
				1.8 V ≤ VDD < 2.7 V		4	MHz
				1.6 V ≤ VDD < 1.8 V		2	MHz
		LP (low-power main) mode		1.6 V ≤ VDD < 1.8 V		2	MHz
Interrupt input high-level width, low-level width	fINTH, fINTL	INTP0 to INTP6, INTP8, INTP9		1.6 V ≤ VDD ≤ 5.5 V	1		μs
Key interrupt input low- level width	fKRH, fKRL	KR0 to KR5		1.8 V ≤ VDD ≤ 5.5 V	250		ns
				1.6 V ≤ VDD < 1.8 V	1		μs
RESET low-level width	fRSL				10		μs

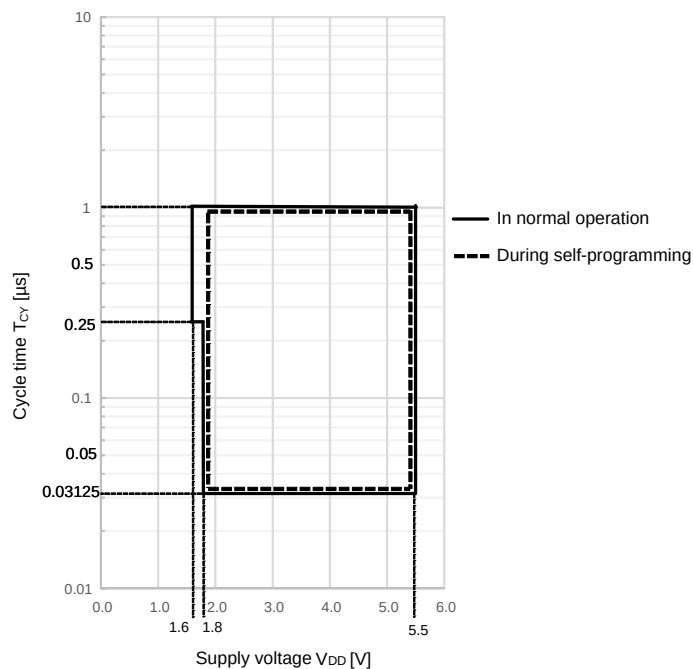
Remark fMCK: Timer array unit operating clock frequency

To set this operating clock, use the CKSmn0 and CKSmn1 bits of the timer mode register mn (TMRmn).

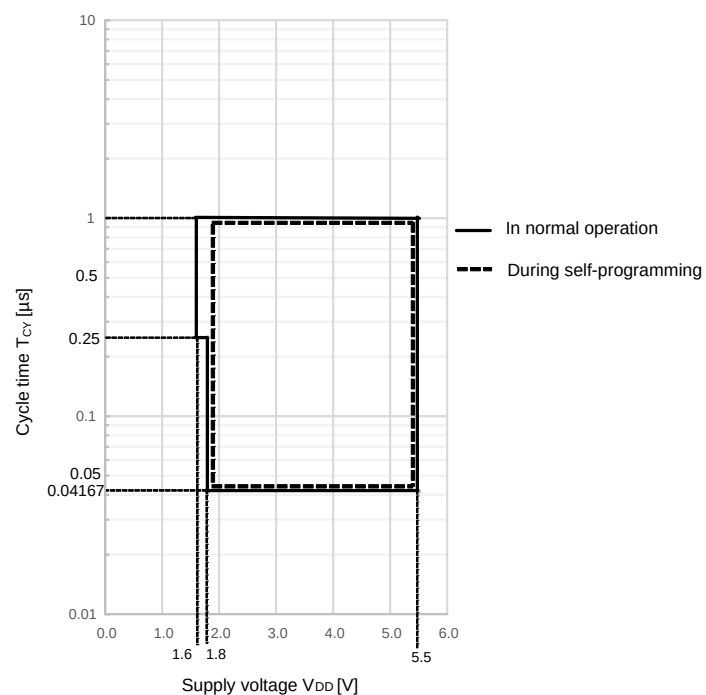
m: Unit number (m = 0), n: Channel number (n = 0 to 3)

- Minimum Instruction Execution Time during Main System Clock Operation

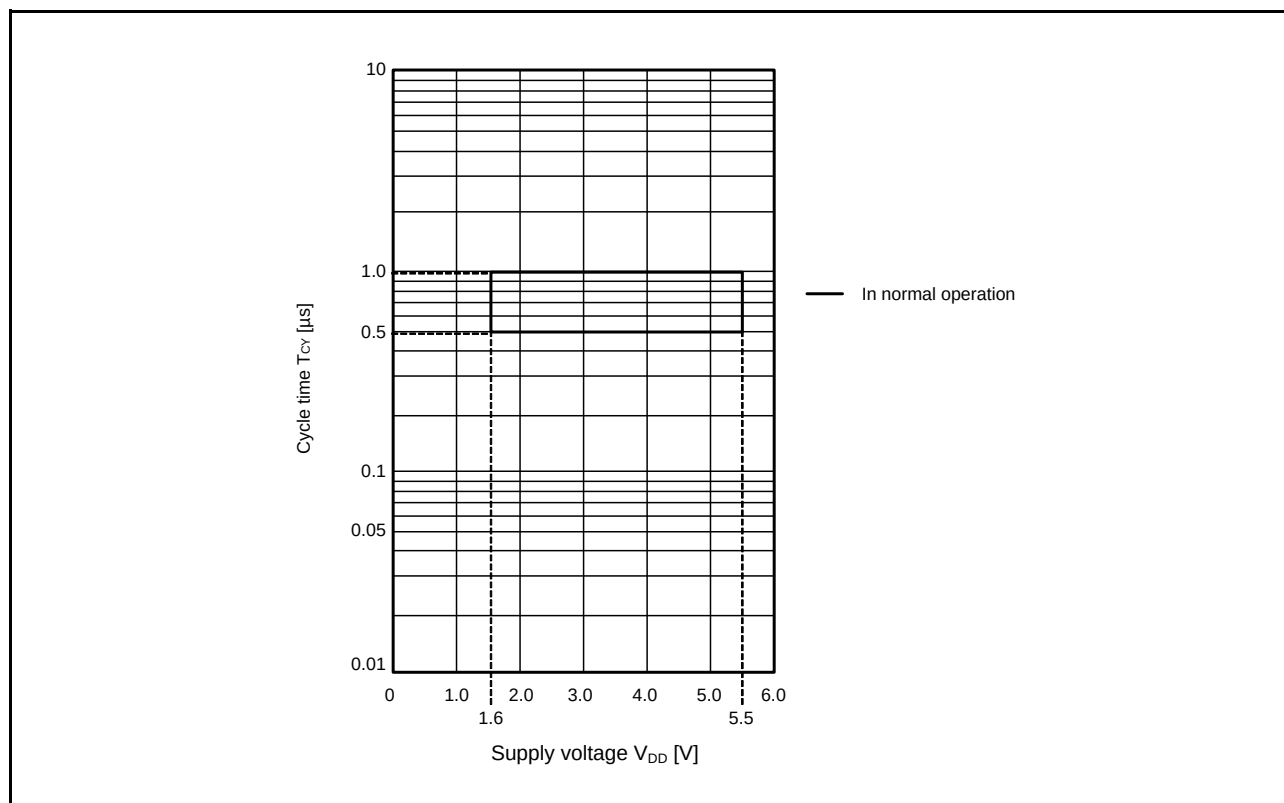
(a) T_{CY} vs V_{DD} in HS (high-speed main) mode



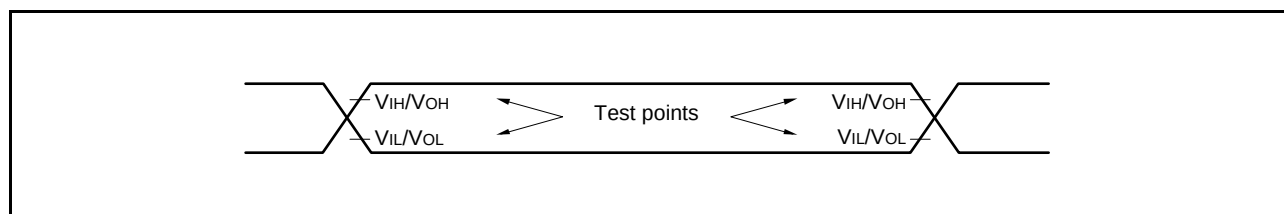
(b) T_{CY} vs V_{DD} in LS (low-speed main) mode



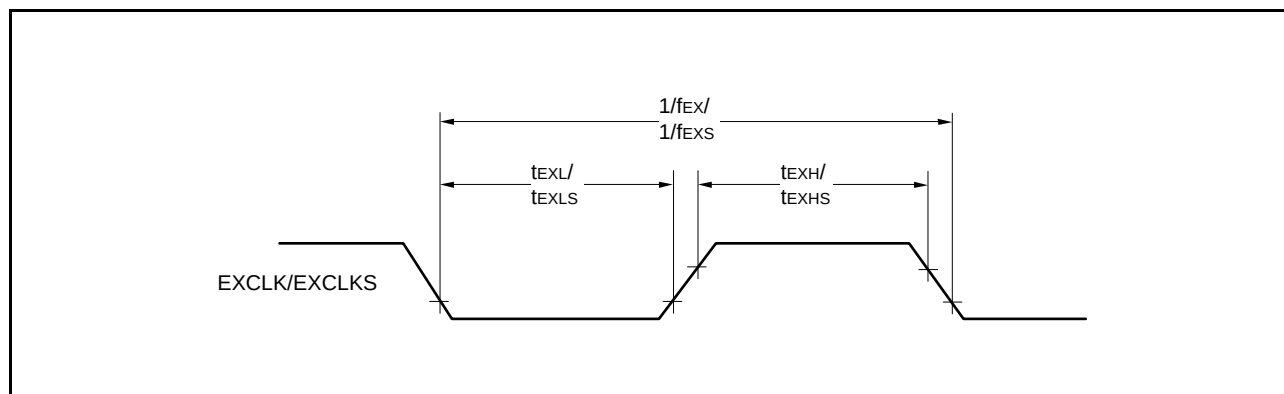
(c) T_{cy} vs V_{DD} in LP (low-power main) mode



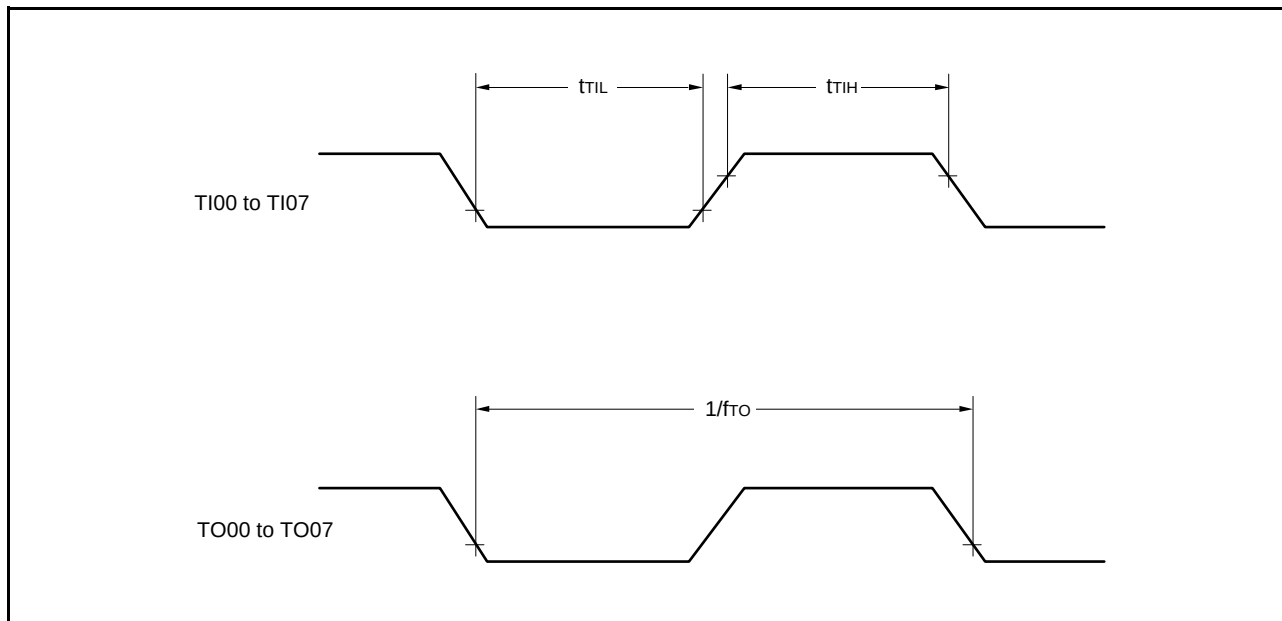
(d) AC Timing Test Points



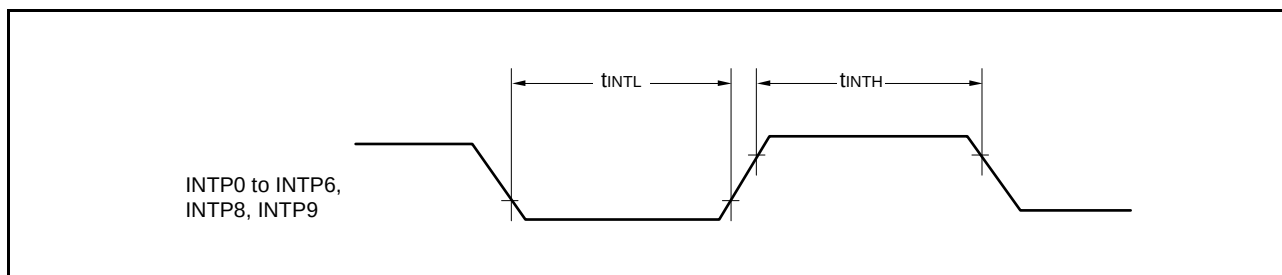
(e) External System Clock Timing



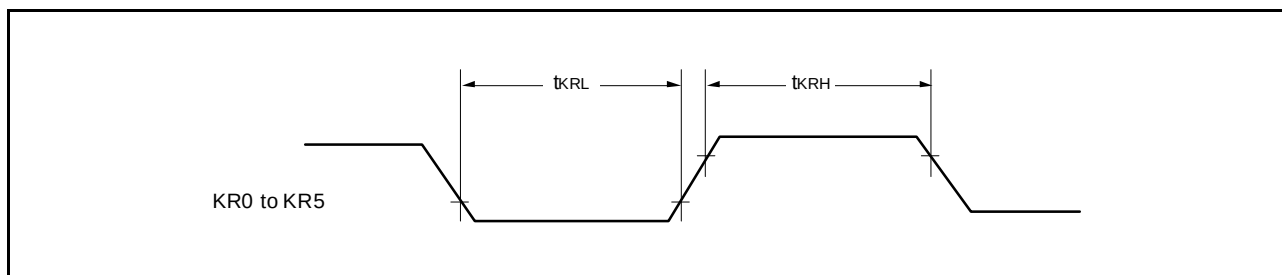
(f) TI/TO Timing



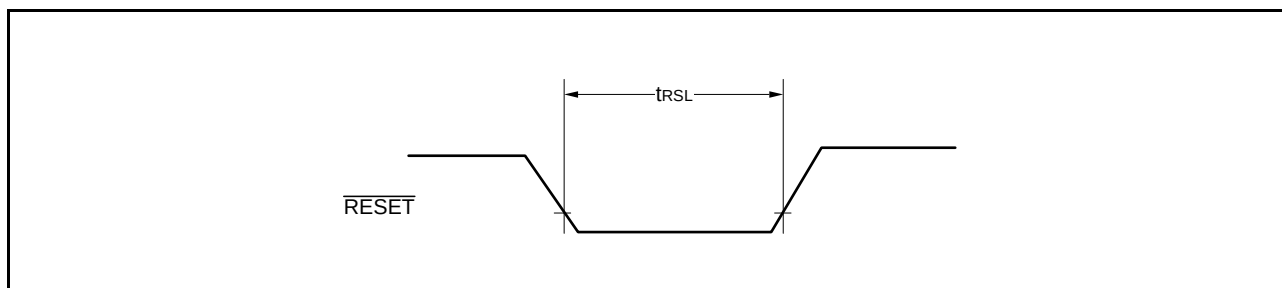
(g) Interrupt Request Input Timing



(h) Key Interrupt Input Timing

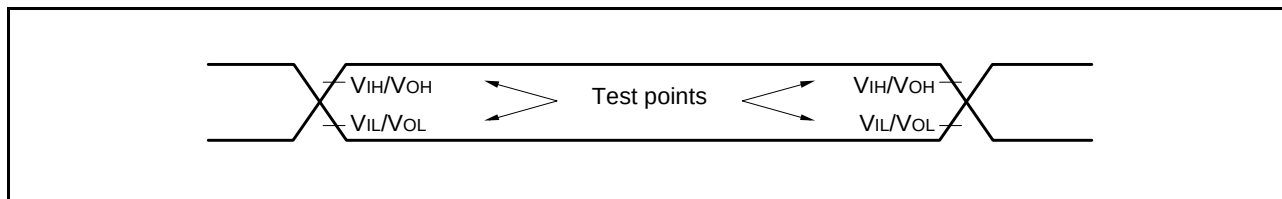


(i) $\overline{\text{RESET}}$ Input Timing



34.5 Characteristics of the Peripheral Functions

AC Timing Test Points



34.5.1 Serial array unit

(1) In UART communications with devices operating at same voltage levels

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate Note 1		1.6 V ≤ VDD ≤ 5.5 V		fMCK/6		fMCK/6		fMCK/6	bps
		Theoretical value of the maximum transfer rate fMCK = fCLKNote 2		5.3		4		0.33	Mbps

Note 1. The transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (fCLK) are as follows.

HS (high-speed main) mode: 32 MHz (1.8 V ≤ VDD ≤ 5.5 V)

4 MHz (1.6 V ≤ VDD ≤ 5.5 V)

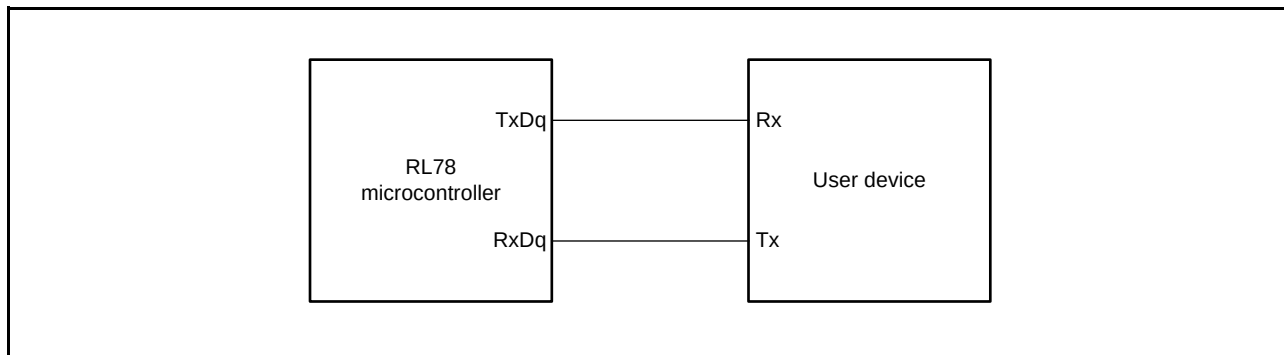
LS (low-speed main) mode: 24 MHz (1.8 V ≤ VDD ≤ 5.5 V)

4 MHz (1.6 V ≤ VDD ≤ 5.5 V)

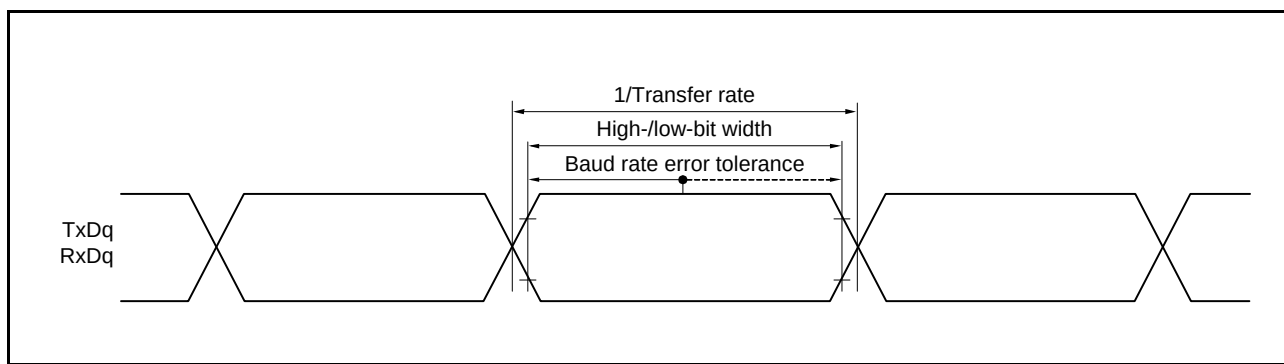
LP (low-power main) mode: 2 MHz (1.6 V ≤ VDD ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

(a) Connection in the UART communications with devices operating at same voltage levels



(b) Bit width in the UART communications when interfacing devices operate at the same voltage level (reference)



Remark 1. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency

To set this operating clock, set the CKSMn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

- (2) In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock (the ratings below are only applicable to CSI00)

(TA = -40 to +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions		HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	tkCY1	tkCY1 ≥ 2/fCLK	4.0 V ≤ VDD ≤ 5.5 V	62.5		83.3		1000		ns
			2.7 V ≤ VDD ≤ 5.5 V	83.3		125		1000		ns
SCKp high-/low-level width	tkH1, tkL1	4.0 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 7		tkCY1/2 - 10		tkCY1/2 - 50		ns
		2.7 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 10		tkCY1/2 - 15		tkCY1/2 - 50		ns
Slp setup time (to SCKp↑) ^{Note 1}	tSIK1	4.0 V ≤ VDD ≤ 5.5 V		23		33		110		ns
		2.7 V ≤ VDD ≤ 5.5 V		33		50		110		ns
Slp hold time (from SCKp↑) Note 1	tKSI1	2.7 V ≤ VDD ≤ 5.5 V		10		10		10		ns
Delay time from SCKp↓ to SOp output ^{Note 2}	tKSO1	C = 20 pF ^{Note 3}			10		10		10	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the Slp setup time becomes “to SCKp↓” and that for the Slp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the Slp pin and the normal output mode for the SOp pin and SCKp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg).

Remark 1. The listed times are only valid when the peripheral I/O redirect function of CSI00 is not in use.

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 1)

Remark 3. fMCK: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00)

- (3) In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions		HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	tkCY1	tkCY1 ≥ 4/fCLK	2.7 V ≤ VDD ≤ 5.5 V	125		166		2000		ns
			2.4 V ≤ VDD ≤ 5.5 V	250		250		2000		ns
			1.8 V ≤ VDD ≤ 5.5 V	500		500		2000		ns
			1.6 V ≤ VDD ≤ 5.5 V	1000		1000		2000		ns
SCKp high-/low-level width	tkH1, tkL1	4.0 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 12		tkCY1/2 - 21		tkCY1/2 - 50		ns
		2.7 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 18		tkCY1/2 - 25		tkCY1/2 - 50		ns
		2.4 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 38		tkCY1/2 - 38		tkCY1/2 - 50		ns
		1.8 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 50		tkCY1/2 - 50		tkCY1/2 - 50		ns
		1.6 V ≤ VDD ≤ 5.5 V		tkCY1/2 - 100		tkCY1/2 - 100		tkCY1/2 - 100		ns
Slp setup time (to SCKp↑) Note 1	tSIK1	4.0 V ≤ VDD ≤ 5.5 V		44		54		110		ns
		2.7 V ≤ VDD ≤ 5.5 V		44		54		110		ns
		2.4 V ≤ VDD ≤ 5.5 V		75		75		110		ns
		1.8 V ≤ VDD ≤ 5.5 V		110		110		110		ns
		1.6 V ≤ VDD ≤ 5.5 V		220		220		220		ns
Slp hold time (from SCKp↑) Note 1	tKSI1	1.6 V ≤ VDD ≤ 5.5 V		19		19		19		ns
Delay time from SCKp↓ to SOp output Note 2	tKSO1	1.6 V ≤ VDD ≤ 5.5 V C = 30 pF Note 3			25		25		25	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the Slp setup time becomes "to SCKp↓" and that for the Slp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the Slp pin and the normal output mode for the SOp pin and SCKp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg).

Remark 1. p: CSI number (p = 00, 01, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), g: PIM and POM numbers (g = 0, 1, 5, 7)

Remark 2. fMCK: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01, 11)

- (4) In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the SCKp external clock

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/2)

Item	Symbol	Conditions		HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time Note 4	tkCY2	4.0 V ≤ VDD ≤ 5.5 V	20 MHz < fMCK	8/fMCK		8/fMCK		—		ns
			fMCK ≤ 20 MHz	6/fMCK		6/fMCK		6/fMCK		ns
		2.7 V ≤ VDD ≤ 5.5 V	16 MHz < fMCK	8/fMCK		8/fMCK		—		ns
			fMCK ≤ 16 MHz	6/fMCK		6/fMCK		6/fMCK		ns
		2.4 V ≤ VDD ≤ 5.5 V		6/fMCK and 500		6/fMCK and 500		6/fMCK and 500		ns
		1.8 V ≤ VDD ≤ 5.5 V		6/fMCK and 750		6/fMCK and 750		6/fMCK and 750		ns
		1.6 V ≤ VDD ≤ 5.5 V		6/fMCK and 1500		6/fMCK and 1500		6/fMCK and 1500		ns
SCKp high-/ low-level width	tkH2, tkL2	4.0 V ≤ VDD ≤ 5.5 V		tkCY2/2 - 7		tkCY2/2 - 7		tkCY2/2 - 7		ns
		2.7 V ≤ VDD ≤ 5.5 V		tkCY2/2 - 8		tkCY2/2 - 8		tkCY2/2 - 8		ns
		1.8 V ≤ VDD ≤ 5.5 V		tkCY2/2 - 18		tkCY2/2 - 18		tkCY2/2 - 18		ns
		1.6 V ≤ VDD ≤ 5.5 V		tkCY2/2 - 66		tkCY2/2 - 66		tkCY2/2 - 66		ns

(Notes, Caution, and Remarks are listed on the next page.)

- (4) In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the SCKp external clock

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/2)

Item	Symbol	Conditions		HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Slp setup time (to SCKp↑) Note 1	tsIK2	2.7 V ≤ VDD ≤ 5.5 V		1/fMCK + 20		1/fMCK + 30		1/fMCK + 30		ns
		1.8 V ≤ VDD ≤ 5.5 V		1/fMCK + 30		1/fMCK + 30		1/fMCK + 30		ns
		1.6 V ≤ VDD ≤ 5.5 V		1/fMCK + 40		1/fMCK + 40		1/fMCK + 40		ns
Slp hold time (from SCKp↑) Note 1	tkSI2	1.8 V ≤ VDD ≤ 5.5 V		1/fMCK + 31		1/fMCK + 31		1/fMCK + 31		ns
		1.6 V ≤ VDD ≤ 5.5 V		1/fMCK + 250		1/fMCK + 250		1/fMCK + 250		ns
Delay time from SCKp↓ to SOp output Note 2	tkSO2	C = 30 pF Note 3	2.7 V ≤ VDD ≤ 5.5 V		2/fMCK + 44		2/fMCK + 110		2/fMCK + 110	ns
			2.4 V ≤ VDD ≤ 5.5 V		2/fMCK + 75		2/fMCK + 110		2/fMCK + 110	ns
			1.8 V ≤ VDD ≤ 5.5 V		2/fMCK + 110		2/fMCK + 110		2/fMCK + 110	ns
			1.6 V ≤ VDD ≤ 5.5 V		2/fMCK + 220		2/fMCK + 220		2/fMCK + 220	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the Slp setup time becomes “to SCKp↓” and that for the Slp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SOp output line.

Note 4. Transfer rate in the SNOOZE mode is 1 Mbps at the maximum.

Caution Select the normal input buffer for the Slp pin and SCKp pin and the normal output mode for the SOp pin by using the port input mode register g (PIMg) and the port output mode register g (POMg).

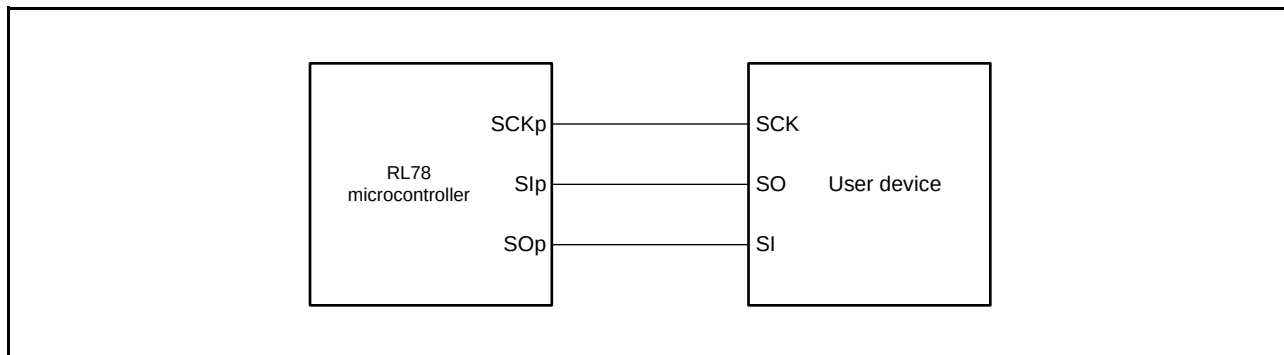
Remark 1. p: CSI number (p = 00, 01, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), g: PIM and POM numbers (g = 0, 1, 5, 7)

Remark 2. fMCK: Serial array unit operation clock frequency

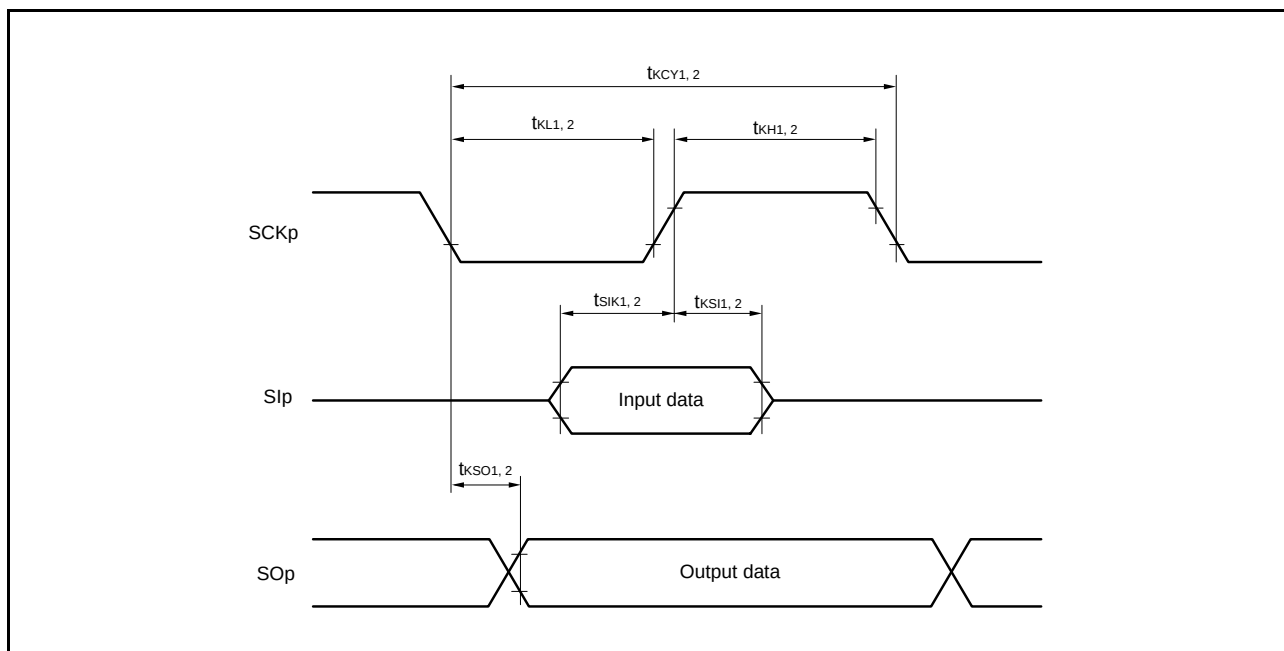
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

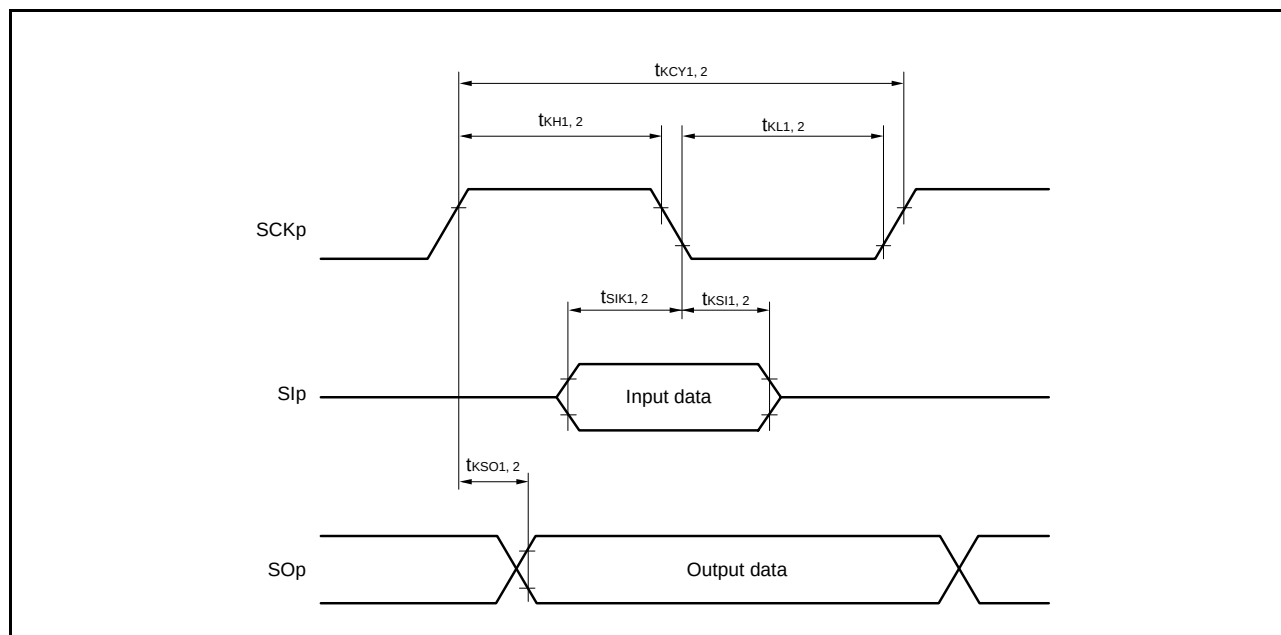
- (a) Connection in the simplified SPI (CSI) communications with devices operating at same voltage levels



- (b) Timing of serial transfer in the simplified SPI (CSI) communications with devices operating at same voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



- (c) Timing of serial transfer in the simplified SPI (CSI) communications with devices operating at same voltage levels when $\text{DAPmn} = 0$ and $\text{CKPmn} = 1$, or $\text{DAPmn} = 1$ and $\text{CKPmn} = 0$



Remark 1. p: CSI number (p = 00, 01, 11, 20, 21)

Remark 2. m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

(5) In simplified I²C communications with devices operating at same voltage levels

(TA = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

(1/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f _{SCL}	2.7 V ≤ V _{DD} ≤ 5.5 V, C _b = 50 pF, R _b = 2.7 kΩ		1000 Note 1		1000 Note 1		400 Note 1	kHz
		1.8 V ≤ V _{DD} ≤ 5.5 V, C _b = 100 pF, R _b = 3 kΩ		400 Note 1		400 Note 1		400 Note 1	kHz
		1.8 V ≤ V _{DD} < 2.7 V, C _b = 100 pF, R _b = 5 kΩ		300 Note 1		300 Note 1		300 Note 1	kHz
		1.6 V ≤ V _{DD} < 1.8 V, C _b = 100 pF, R _b = 5 kΩ		250 Note 1		250 Note 1		250 Note 1	kHz
Hold time when SCLr is low	t _{LOW}	2.7 V ≤ V _{DD} ≤ 5.5 V, C _b = 50 pF, R _b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ V _{DD} ≤ 5.5 V, C _b = 100 pF, R _b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ V _{DD} < 2.7 V, C _b = 100 pF, R _b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V _{DD} < 1.8 V, C _b = 100 pF, R _b = 5 kΩ	1850		1850		1850		ns
Hold time when SCLr is high	t _{HIGH}	2.7 V ≤ V _{DD} ≤ 5.5 V, C _b = 50 pF, R _b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ V _{DD} ≤ 5.5 V, C _b = 100 pF, R _b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ V _{DD} < 2.7 V, C _b = 100 pF, R _b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ V _{DD} < 1.8 V, C _b = 100 pF, R _b = 5 kΩ	1850		1850		1850		ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

(5) In simplified I²C communications with devices operating at same voltage levels

(TA = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

(2/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	t _{SU:DAT}	2.7 V ≤ V _{DD} ≤ 5.5 V, C _b = 50 pF, R _b = 2.7 kΩ	1/f _{MCK} + 85 Note 2		1/f _{MCK} + 85 Note 2		1/f _{MCK} + 145 Note 2		ns
		1.8 V ≤ V _{DD} ≤ 5.5 V, C _b = 100 pF, R _b = 3 kΩ	1/f _{MCK} + 145 Note 2		1/f _{MCK} + 145 Note 2		1/f _{MCK} + 145 Note 2		ns
		1.8 V ≤ V _{DD} < 2.7 V, C _b = 100 pF, R _b = 5 kΩ	1/f _{MCK} + 230 Note 2		1/f _{MCK} + 230 Note 2		1/f _{MCK} + 230 Note 2		ns
		1.6 V ≤ V _{DD} < 1.8 V, C _b = 100 pF, R _b = 5 kΩ	1/f _{MCK} + 290 Note 2		1/f _{MCK} + 290 Note 2		1/f _{MCK} + 290 Note 2		ns
Data hold time (transmission)	t _{HD:DAT}	2.7 V ≤ V _{DD} ≤ 5.5 V, C _b = 50 pF, R _b = 2.7 kΩ	0	305	0	305	0	305	ns
		1.8 V ≤ V _{DD} ≤ 5.5 V, C _b = 100 pF, R _b = 3 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ V _{DD} < 2.7 V, C _b = 100 pF, R _b = 5 kΩ	0	405	0	405	0	405	ns
		1.6 V ≤ V _{DD} < 1.8 V, C _b = 100 pF, R _b = 5 kΩ	0	405	0	405	0	405	ns

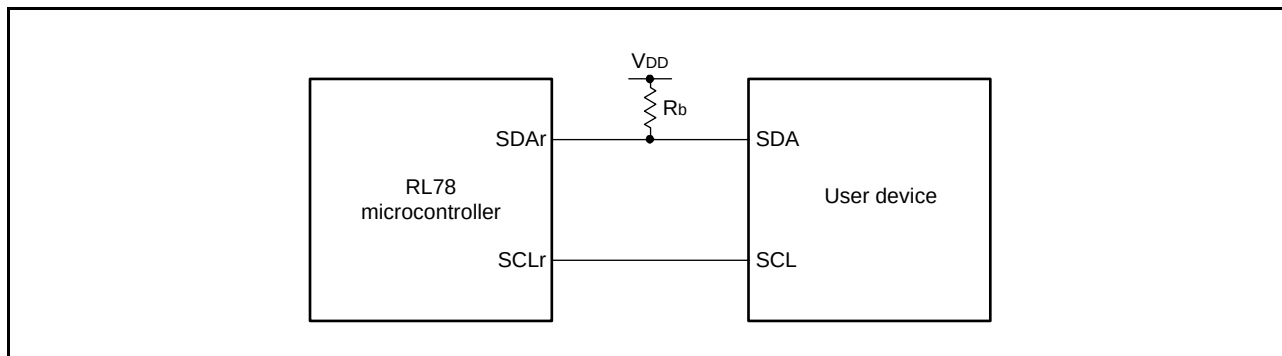
Note 1. The listed times must be no greater than f_{MCK}/4.

Note 2. Set f_{MCK} so that it will not exceed the hold time when SCLr is low or high.

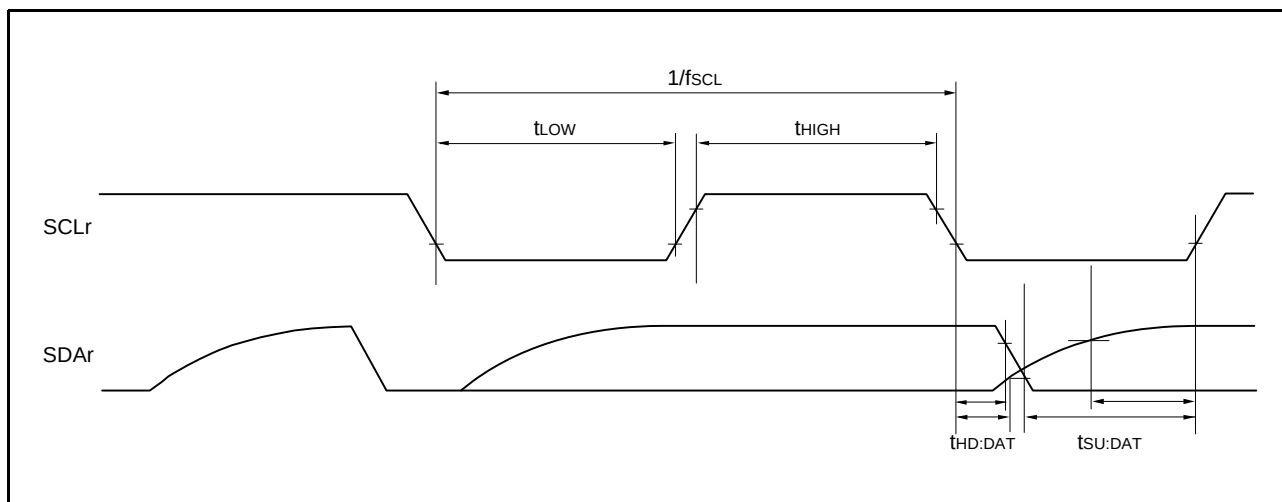
Caution Select the normal input buffer and the N-ch open drain output (withstand voltage of V_{DD}) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

(Remarks are listed on the next page.)

(a) Connection in the simplified I²C communications with devices operating at same voltage levels



(b) Timing of serial transfer in the simplified I²C communications with devices operating at same voltage levels



Remark 1. $R_b[Q]$: Communication line (SDAr) pull-up resistance, $C_b[F]$: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 01, 10, 11, 20, 21), g: PIM number (g = 0, 1, 3, 7), h: POM number (g = 1, 5, 7)

Remark 3. f_{MCK} : Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

(6) In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Reception	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V		fMCK/6 Note 1		fMCK/6 Note 1		bps
					5.3		4		Mbps
			2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V		fMCK/6 Note 1		fMCK/6 Note 1		bps
					5.3		4		Mbps
			1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V		fMCK/6 Notes 1, 2		fMCK/6 Notes 1, 2		bps
					5.3		4		Mbps

Note 1. Transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. Use this rate with VDD ≥ Vb.

Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (fCLK) are:
 HS (high-speed main) mode: 32 MHz (1.8 V ≤ VDD ≤ 5.5 V)
 4 MHz (1.6 V ≤ VDD ≤ 5.5 V)
 LS (low-speed main) mode: 24 MHz (1.8 V ≤ VDD ≤ 5.5 V)
 4 MHz (1.6 V ≤ VDD ≤ 5.5 V)
 LP (low-power main) mode: 2 MHz (1.6 V ≤ VDD ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (withstand voltage of VDD) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

Remark 1. Vb[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1)

Remark 3. fMCK: Serial array unit operation clock frequency
 To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
 m: Unit number, n: Channel number (mn = 00, 01)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when the setting of bit 1 (PIOR1) of the peripheral I/O redirection register (PIOR) is 1.

(6) In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Transmission	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V			Note 1		Note 1	bps
			Theoretical value of the maximum transfer rate Cb = 50 pF, Rb = 1.4 kΩ, Vb = 2.7 V			2.8Note 2		2.8Note 2	Mbps
			2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V			Note 3		Note 3	bps
			Theoretical value of the maximum transfer rate Cb = 50 pF, Rb = 2.7 kΩ, Vb = 2.3 V			1.2Note 4		1.2Note 4	Mbps
			1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V			Notes 5, 6		Notes 5, 6	bps
			Theoretical value of the maximum transfer rate Cb = 50 pF, Rb = 5.5 kΩ, Vb = 1.6 V			0.43 Note 7		0.43 Note 7	Mbps

Note 1. The smaller maximum transfer rate derived by using fmck/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 [\%]$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This rate is calculated as an example when the conditions described in the "Conditions" column are met. See **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

(Notes and Caution continue in the next page.)

- Note 3.** The smaller maximum transfer rate derived by using fmck/6 or the following expression is the valid maximum transfer rate. Expression for calculating the transfer rate when 2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

- Note 4.** This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

- Note 5.** Use this rate with VDD ≥ Vb.

- Note 6.** The smaller maximum transfer rate derived by using fmck/6 or the following expression is the valid maximum transfer rate. Expression for calculating the transfer rate when 1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

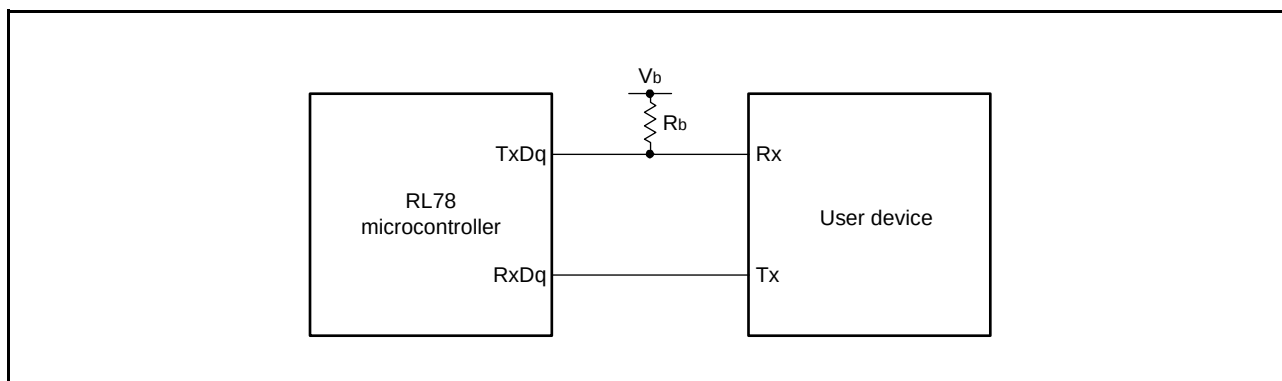
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

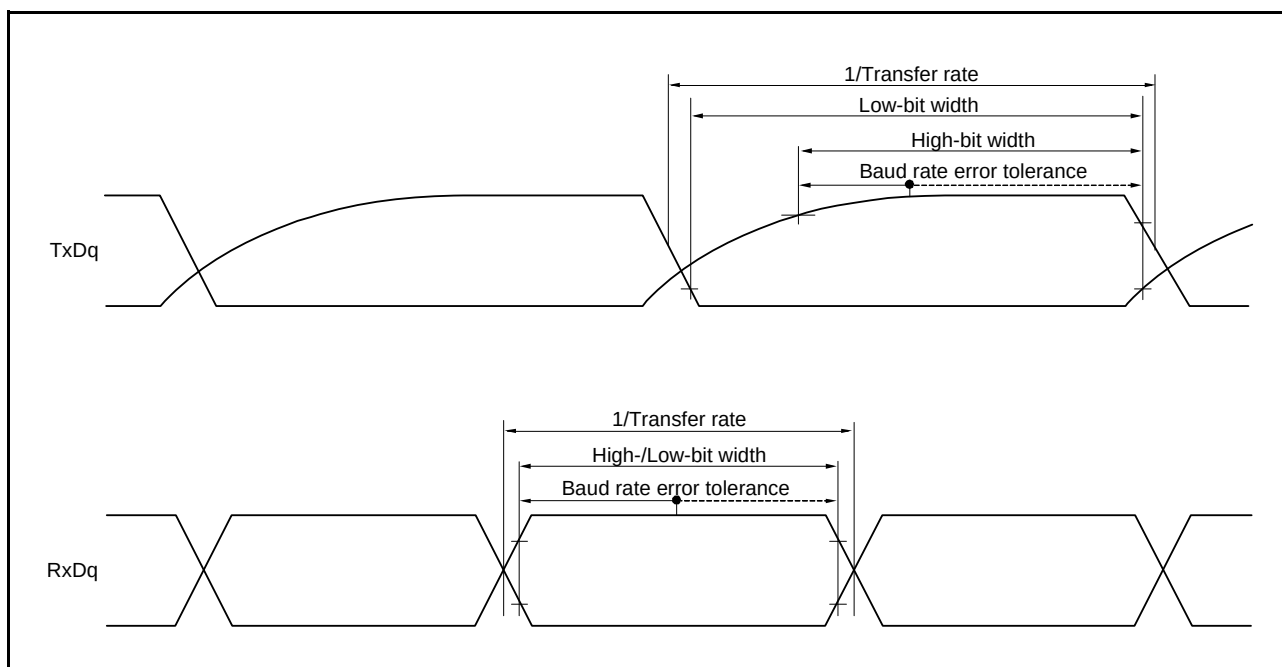
- Note 7.** This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

- Caution** Select the TTL input buffer for the RxDq pin and the N-ch open drain output (withstand voltage of VDD) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(a) In UART communications with devices operating at different voltage levels



(b) Bit width in the UART communications with devices operating at different voltage levels (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when the setting of bit 1 (PIOR1) of the peripheral I/O redirection register (PIOR) is 1.

- (7) In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(TA = -40 to +105°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	tKCY1	tKCY1 ≥ 2/fCLK 4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	200		200		2300		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	300		300		2300		ns
SCKp high-level width	tKH1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	tKCY1/2 - 50		tKCY1/2 - 50		tKCY1/2 - 50		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	tKCY1/2 - 120		tKCY1/2 - 120		tKCY1/2 - 120		ns
SCKp low-level width	tKL1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	tKCY1/2 - 7		tKCY1/2 - 7		tKCY1/2 - 50		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	tKCY1/2 - 10		tKCY1/2 - 10		tKCY1/2 - 50		ns
Slp setup time (to SCKp↑)Note 1	tSIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	58		58		479		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	121		121		479		ns
Slp hold time (from SCKp↑)Note 1	tSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↓ to SOp outputNote 1	tKSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ		60		60		60	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ		130		130		130	ns

(Notes, Caution, and Remarks are listed on the next page.)

- (7) In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(TA = -40 to +105°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Slp setup time (to SCKp↓) ^{Note 2}	tsIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	23		23		110		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	33		33		110		ns
Slp hold time (from SCKp↓) ^{Note 2}	tKSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↑ to SOp output ^{Note 2}	tKSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 20 pF, Rb = 1.4 kΩ		10		10		10	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 20 pF, Rb = 2.7 kΩ		10		10		10	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the Slp pin and the N-ch open drain output (withstand voltage of VDD) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

Remark 1. Rb[Ω]: Communication line (SCKp, SOp) pull-up resistance, Cb[F]: Communication line (SCKp, SOp) load capacitance, Vb[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 1)

Remark 3. fMCK: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn= 00)

Remark 4. The listed times are only valid when the peripheral I/O redirect function of CSI00 is not in use.

- (8) In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/3)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	tKCY1	tKCY1 ≥ 4/fCLK 4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	300		300		2300		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	500		500		2300		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note} , Cb = 30 pF, Rb = 5.5 kΩ	1150		1150		2300		ns
SCKp high-level width	tKH1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	tKCY1/2 - 75		tKCY1/2 - 75		tKCY1/2 - 75		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	tKCY1/2 - 170		tKCY1/2 - 170		tKCY1/2 - 170		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note} , Cb = 30 pF, Rb = 5.5 kΩ	tKCY1/2 - 458		tKCY1/2 - 458		tKCY1/2 - 458		ns
SCKp low-level width	tKL1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	tKCY1/2 - 12		tKCY1/2 - 12		tKCY1/2 - 50		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	tKCY1/2 - 18		tKCY1/2 - 18		tKCY1/2 - 50		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note} , Cb = 30 pF, Rb = 5.5 kΩ	tKCY1/2 - 50		tKCY1/2 - 50		tKCY1/2 - 50		ns

Note Use this setting with VDD ≥ Vb.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of VDD) mode for the SOP pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(Remarks are listed two pages after the next page.)

- (8) In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/3)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Slp setup time (to SCKp↑) ^{Note 1}	tSIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	81		81		479		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	177		177		479		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ	479		479		479		ns
Slp hold time (from SCKp↑) ^{Note 1}	tKSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↓ to SOp output ^{Note 1}	tKSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ		100		100		100	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		195		195		195	ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ		483		483		483	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. Use this setting with VDD ≥ Vb.

Caution Select the TTL input buffer for the Slp pin and the N-ch open drain output (withstand voltage of VDD) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the page after the next page.)

- (8) In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(3/3)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Slp setup time (to SCKp↓) ^{Note 1}	tsIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	44		44		110		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	44		44		110		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ	110		110		110		ns
Slp hold time (from SCKp↓) ^{Note 1}	tKSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp output ^{Note 1}	tKSI01	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ		25		25		25	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		25		25		25	ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ		25		25		25	ns

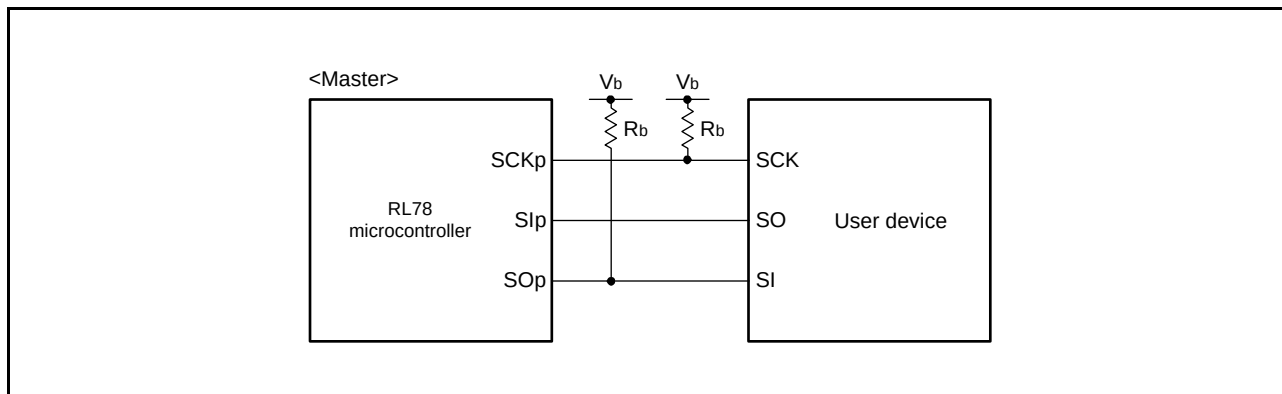
Note 1. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. Use this setting with VDD ≥ Vb.

Caution Select the TTL input buffer for the Slp pin and the N-ch open drain output (withstand voltage of VDD) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

(a) Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



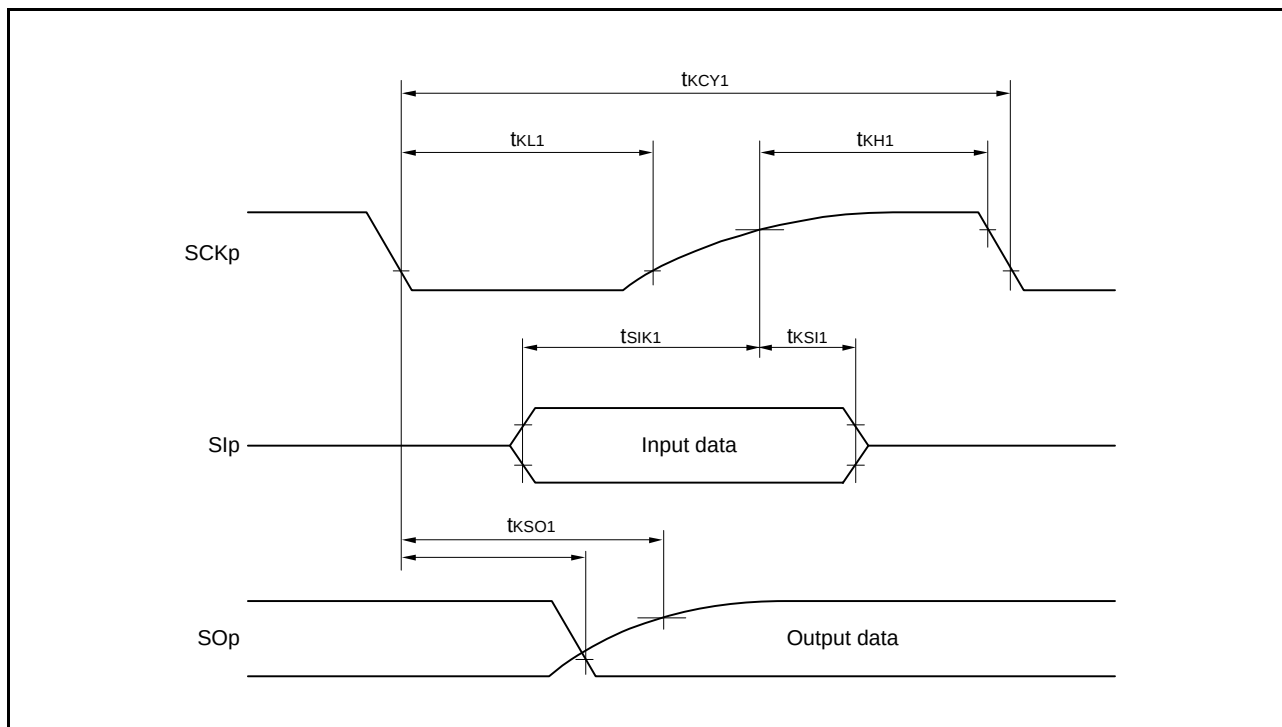
Remark 1. $R_b[\Omega]$: Communication line (SCKp, SOp) pull-up resistance, $C_b[F]$: Communication line (SCKp, SOp) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 20), m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11), g: PIM and POM number (g = 1, 3, 7)

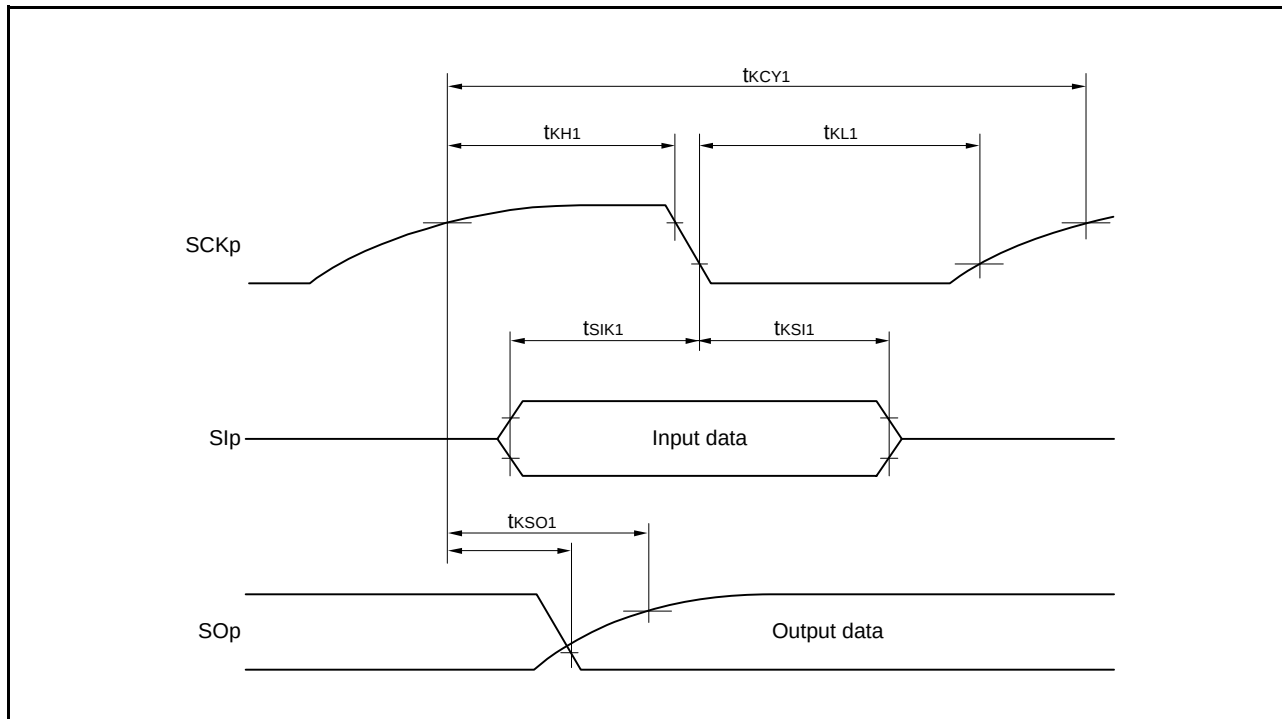
Remark 3. f_{MCK} : Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

Remark 4. Communications by using CSI01 of 48-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

(b) Timing of serial transfer in the simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



- (c) Timing of serial transfer in the simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels when $\text{DAPmn} = 0$ and $\text{CKPmn} = 1$, or $\text{DAPmn} = 1$ and $\text{CKPmn} = 0$



Remark 1. p: CSI number ($p = 00, 01, 20$), m: Unit number, n: Channel number ($mn = 00, 01, 03, 10, 11$), g: PIM and POM number ($g = 1, 3, 7$)

Remark 2. Communications by using CSI01 of 48-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

- (9) In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(1/2)

Item	Symbol	Conditions		HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time Note 1	tkCY2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V	24 MHz < fMCK	14/fMCK		—		—		ns
			20 MHz < fMCK ≤ 24 MHz	12/fMCK		12/fMCK		—		ns
			8 MHz < fMCK ≤ 20 MHz	10/fMCK		10/fMCK		—		ns
			4 MHz < fMCK ≤ 8 MHz	8/fMCK		8/fMCK		—		ns
			fMCK ≤ 4 MHz	6/fMCK		6/fMCK		10/fMCK		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V,	24 MHz < fMCK	20/fMCK		—		—		ns
			20 MHz < fMCK ≤ 24 MHz	16/fMCK		16/fMCK		—		ns
			16 MHz < fMCK ≤ 20 MHz	14/fMCK		14/fMCK		—		ns
			8 MHz < fMCK ≤ 16 MHz	12/fMCK		12/fMCK		—		ns
			4 MHz < fMCK ≤ 8 MHz	8/fMCK		8/fMCK		—		ns
			fMCK ≤ 4 MHz	6/fMCK		6/fMCK		10/fMCK		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V Note 2	24 MHz < fMCK	48/fMCK		—		—		ns
			20 MHz < fMCK ≤ 24 MHz	36/fMCK		36/fMCK		—		ns
			16 MHz < fMCK ≤ 20 MHz	32/fMCK		32/fMCK		—		ns
			8 MHz < fMCK ≤ 16 MHz	26/fMCK		26/fMCK		—		ns
			4 MHz < fMCK ≤ 8 MHz	16/fMCK		16/fMCK		—		ns
			fMCK ≤ 4 MHz	10/fMCK		10/fMCK		10/fMCK		ns

(Notes and Caution are listed on the next page, and Remarks are listed on the page after the next page.)

- (9) In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCKp high-/low-level width	tkH2, tkL2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V	tkCY2/2 - 12		tkCY2/2 - 12		tkCY2/2 - 50		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V	tkCY2/2 - 18		tkCY2/2 - 18		tkCY2/2 - 50		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2}	tkCY2/2 - 50		tkCY2/2 - 50		tkCY2/2 - 50		ns
Slp setup time (to SCKp↑) ^{Note 3}	tsIK2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V	1/fMCK + 20		1/fMCK + 20		1/fMCK + 30		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V	1/fMCK + 20		1/fMCK + 20		1/fMCK + 30		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2}	1/fMCK + 30		1/fMCK + 30		1/fMCK + 30		ns
Slp hold time (from SCKp↑) ^{Note 3}	tsIS2		1/fMCK + 31		1/fMCK + 31		1/fMCK + 31		ns
Delay time from SCKp↓ to SOp output ^{Note 4}	tkSO2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 30 pF, Rb = 1.4 kΩ		2/fMCK + 120		2/fMCK + 120		2/fMCK + 573	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 30 pF, Rb = 2.7 kΩ		2/fMCK + 214		2/fMCK + 214		2/fMCK + 573	ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V ^{Note 2} , Cb = 30 pF, Rb = 5.5 kΩ		2/fMCK + 573		2/fMCK + 573		2/fMCK + 573	ns

Note 1. Transfer rate in the SNOOZE mode: 1 Mbps (max.)

Note 2. Use this setting with VDD ≥ Vb.

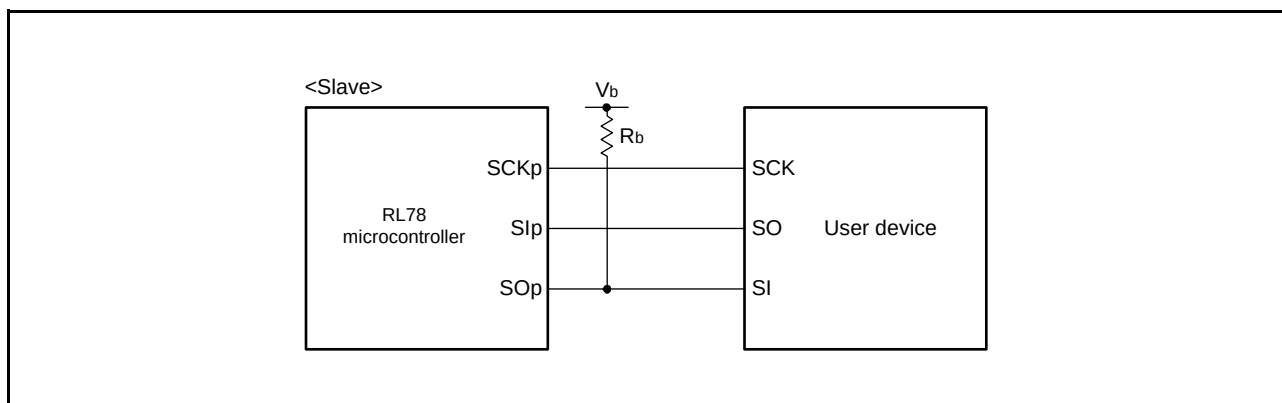
Note 3. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The Slp setup time becomes "to SCKp↓" and Slp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the Slp pin and the N-ch open drain output (withstand voltage of VDD) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

(a) Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



Remark 1. $R_b[\Omega]$: Communication line (SO_p) pull-up resistance, $C_b[F]$: Communication line (SO_p) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 20), m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11), g: PIM and POM number (g = 1, 3, 7)

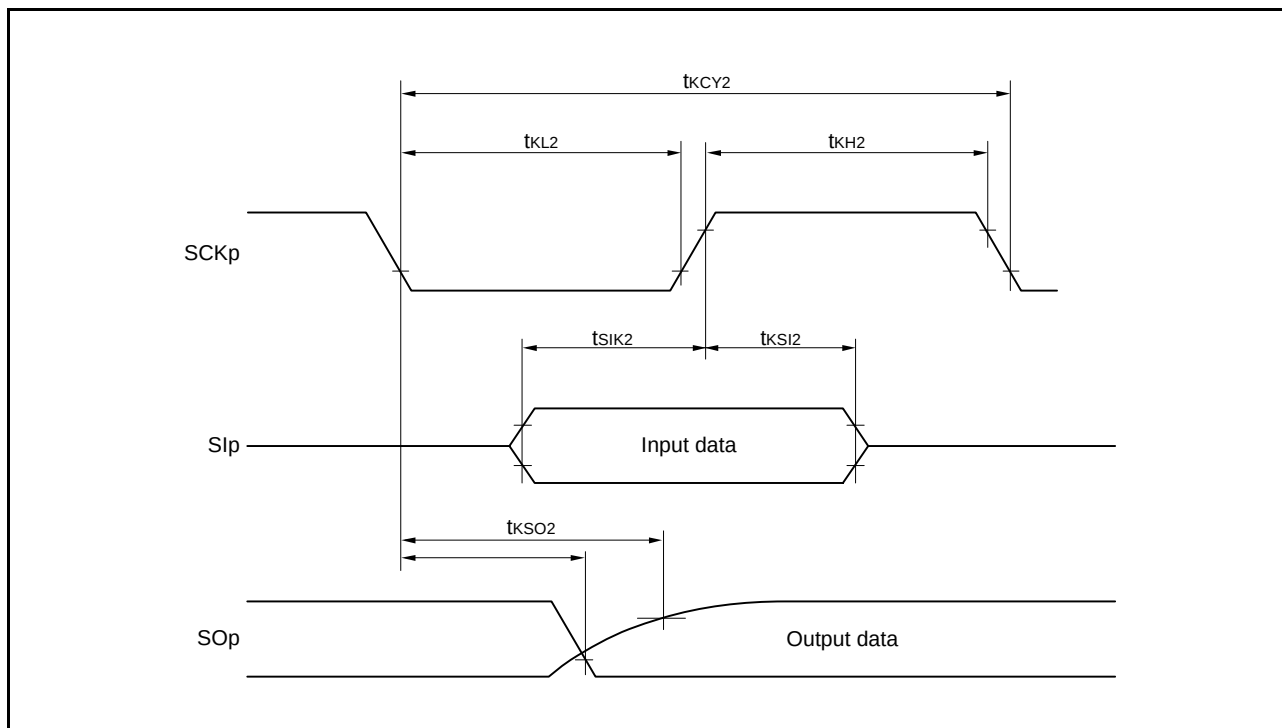
Remark 3. f_{MCK} : Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

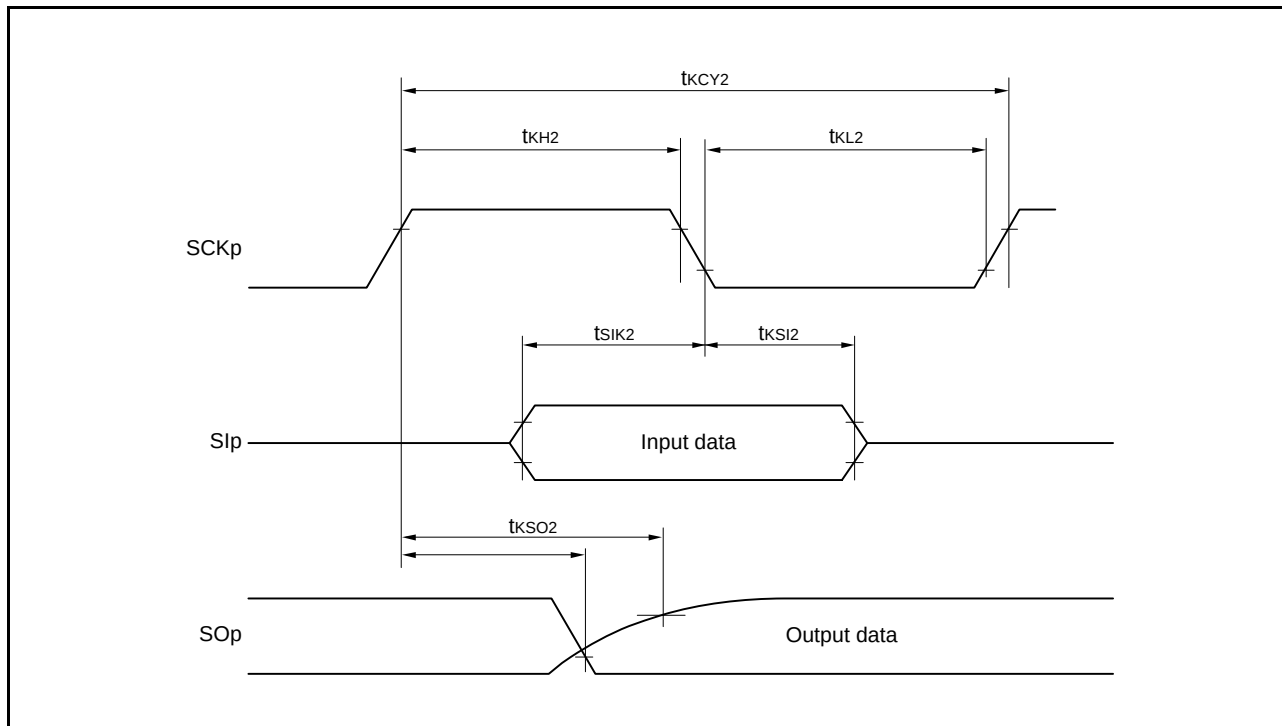
m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

Remark 4. Communications by using CSI01 of 48-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

(b) Timing of serial transfer in the simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



- (c) Timing of serial transfer in the simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels when $\text{DAPmn} = 0$ and $\text{CKPmn} = 1$, or $\text{DAPmn} = 1$ and $\text{CKPmn} = 0$



Remark 1. p: CSI number ($p = 00, 01, 20$), m: Unit number, n: Channel number ($mn = 00, 01, 03, 10, 11$), g: PIM and POM number ($g = 1, 3, 7$)

Remark 2. Communications by using CSI01 of 48-pin products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

(10) Simplified I²C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V)

(TA = -40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

(1/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f _{SCL}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 2.7 kΩ		1000 Note 1		1000 Note 1		300 Note 1	kHz
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ		1000 Note 1		1000 Note 1		300 Note 1	kHz
		4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 2.8 kΩ		400 Note 1		400 Note 1		300 Note 1	kHz
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ		400 Note 1		400 Note 1		300 Note 1	kHz
		1.8 V ≤ V _{DD} < 3.3 V, 1.6 V ≤ V _b ≤ 2.0 V ^{Note 2} , C _b = 100 pF, R _b = 5.5 kΩ		300 Note 1		300 Note 1		300 Note 1	kHz
Hold time when SCLr is low	t _{LOW}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 2.7 kΩ	475		475		1550		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	475		475		1550		ns
		4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 2.8 kΩ	1150		1550		1550		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	1150		1550		1550		ns
		1.8 V ≤ V _{DD} < 3.3 V, 1.6 V ≤ V _b ≤ 2.0 V ^{Note 2} , C _b = 100 pF, R _b = 5.5 kΩ	1550		1550		1550		ns
Hold time when SCLr is high	t _{HIGH}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 2.7 kΩ	245		245		610		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	200		200		610		ns
		4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 2.8 kΩ	675		675		610		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	600		600		610		ns
		1.8 V ≤ V _{DD} < 3.3 V, 1.6 V ≤ V _b ≤ 2.0 V ^{Note 2} , C _b = 100 pF, R _b = 5.5 kΩ	610		610		610		ns

(10) Simplified I²C communications with devices operating at different voltage levels (1.8 V, 2.5 V, and 3 V)

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

(2/2)

Item	Symbol	Conditions	HS (High-Speed Main) Mode		LS (Low-Speed Main) Mode		LP (Low-Power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	tSU:DAT	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 2.7 kΩ	1/fMCK + 135 Note 3		1/fMCK + 135 Note 3		1/fMCK + 190 Note 3		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	1/fMCK + 135 Note 3		1/fMCK + 135 Note 3		1/fMCK + 190 Note 3		ns
		4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 2.8 kΩ	1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 VNote 2, Cb = 100 pF, Rb = 5.5 kΩ	1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		1/fMCK + 190 Note 3		ns
Data hold time (transmission)	tHD:DAT	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	0	305	0	305	0	305	ns
		4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 2.8 kΩ	0	355	0	355	0	355	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 VNote 2, Cb = 100 pF, Rb = 5.5 kΩ	0	405	0	405	0	405	ns

Note 1. The listed times must be no greater than fMCK/4.

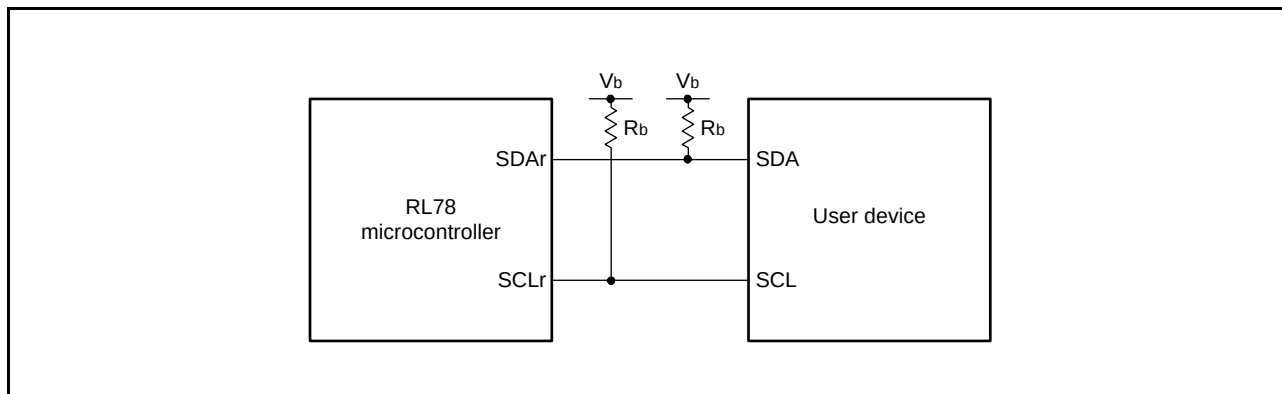
Note 2. Use this setting with VDD ≥ Vb.

Note 3. Set fMCK so that it will not exceed the hold time when SCLr is low or high.

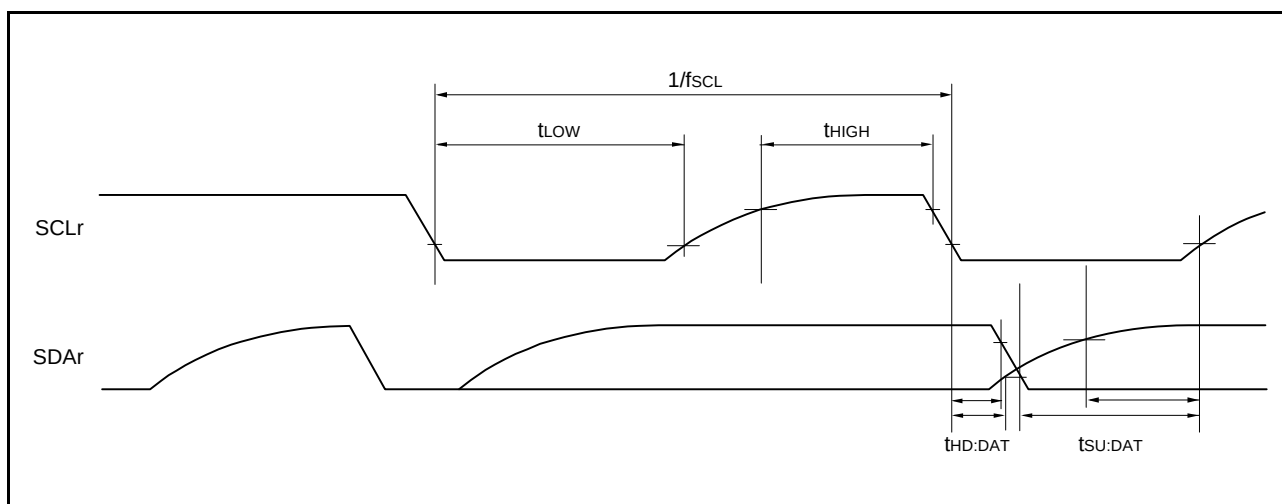
Caution Select the TTL input buffer and the N-ch open drain output (withstand voltage of VDD) mode for the SDAr pin and the N-ch open drain output (withstand voltage of VDD) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

(a) Connection in the I²C communications with devices operating at different voltage levels



(b) Timing of serial transfer in the simplified I²C communications with devices operating at different voltage levels



Remark 1. $R_b[\Omega]$: Communication line (SDAr, SCLr) pull-up resistance, $C_b[F]$: Communication line (SDAr, SCLr) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. r: IIC number (r = 00, 01, 11, 20, 21), g: PIM and POM number (g = 0, 1, 3, 7), POM number (h = 1, 5, 7)

Remark 3. f_{MCK} : Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).

m: Unit number, n: Channel number (mn = 00, 01, 03, 10, 11)

34.5.2 Serial interface UARTA

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate			200	0	153600	bps

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark g: PIM number (g = 0, 1), h: POM number (h = 0, 1)

34.5.3 Serial interface IICA

(1) I²C standard mode

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	fSCL	Standard mode: fCLK ≥ 1 MHz	0		100	kHz
Setup time of restart condition	tSU:STA		4.7			μs
Hold time ^{Note 1}	tHD:STA		4.0			μs
Hold time when SCLA0 is low	tLOW		4.7			μs
Hold time when SCLA0 is high	tHIGH		4.0			μs
Data setup time (reception)	tSU:DAT		250			ns
Data hold time (transmission) ^{Note 2}	tHD:DAT		0		3.45	μs
Setup time of stop condition	tSU:STO		4.0			μs
Bus-free time	tBUF		4.7			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of tHD:DAT applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution The listed frequency and times apply even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (IOH1, IOL1, VOH1, VOL1) must satisfy the values in the redirect destination.

Remark The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 400 pF, Rb = 2.7 kΩ

(2) I²C fast mode

(TA = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f _{SCL}	Fast mode: f _{CLK} ≥ 3.5 MHz 1.8 V ≤ V _{DD} ≤ 5.5 V	0		400	kHz
Setup time of restart condition	t _{SU:STA}	1.8 V ≤ V _{DD} ≤ 5.5 V	0.6			μs
Hold time ^{Note 1}	t _{HD:STA}	1.8 V ≤ V _{DD} ≤ 5.5 V	0.6			μs
Hold time when SCLA0 is low	t _{LOW}	1.8 V ≤ V _{DD} ≤ 5.5 V	1.3			μs
Hold time when SCLA0 is high	t _{HIGH}	1.8 V ≤ V _{DD} ≤ 5.5 V	0.6			μs
Data setup time (reception)	t _{SU:DAT}	1.8 V ≤ V _{DD} ≤ 5.5 V	100			ns
Data hold time (transmission) ^{Note 2}	t _{HD:DAT}	1.8 V ≤ V _{DD} ≤ 5.5 V	0		0.9	μs
Setup time of stop condition	t _{SU:STO}	1.8 V ≤ V _{DD} ≤ 5.5 V	0.6			μs
Bus-free time	t _{BUF}	1.8 V ≤ V _{DD} ≤ 5.5 V	1.3			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of t_{HD:DAT} applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution The values in the above table apply even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (IOH1, IOL1, VOH1, VOL1) must satisfy the values in the redirect destination.

Remark The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 320 pF, Rb = 1.1 kΩ

(3) I²C fast mode plus

(TA = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLAO clock frequency	f _{SCL}	Fast mode plus: f _{CLK} ≥ 10 MHz 2.7 V ≤ V _{DD} ≤ 5.5 V	0		1000	kHz
Setup time of restart condition	t _{SU:STA}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.26			μs
Hold time ^{Note 1}	t _{HD:STA}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.26			μs
Hold time when SCLAO is low	t _{LOW}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.5			μs
Hold time when SCLAO is high	t _{HIGH}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.26			μs
Data setup time (reception)	t _{SU:DAT}	2.7 V ≤ V _{DD} ≤ 5.5 V	50			ns
Data hold time (transmission) ^{Note 2}	t _{HD:DAT}	2.7 V ≤ V _{DD} ≤ 5.5 V	0		0.45	μs
Setup time of stop condition	t _{SU:STO}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.26			μs
Bus-free time	t _{BUF}	2.7 V ≤ V _{DD} ≤ 5.5 V	0.5			μs

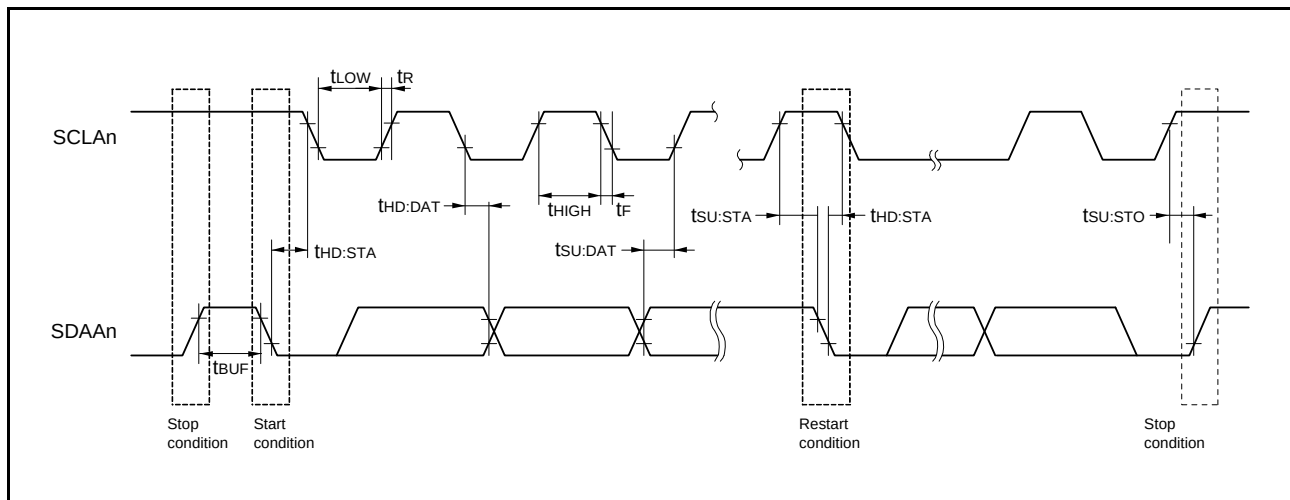
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of t_{HD:DAT} applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution The values in the above table apply even when bit 2 (PIOR2) in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (IOH1, IOL1, VOH1, VOL1) must satisfy the values in the redirect destination.

Remark The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 120 pF, Rb = 1.1 kΩ

I²C serial transfer timing



Remark n = 0

34.6 Characteristics of the Analog Circuits

34.6.1 Characteristics of the A/D converter

Reference for the characteristics of the A/D converter

Reference Voltage Input Channel	Reference Voltage (+) = AVREFP Reference Voltage (-) = AVREFM	Reference Voltage (+) = VDD Reference Voltage (-) = VSS	Reference Voltage (+) = VBGR Reference Voltage (-) = AVREFM
ANI0 to ANI7	See 34.6.1 (1)	See 34.6.1 (3)	See 34.6.1 (4)
ANI16 to ANI19	See 34.6.1 (2)		
Internal reference voltage and temperature sensor output voltage	See 34.6.1 (1)		—

- (1) Reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target: ANI2 to ANI7, internal reference voltage, and temperature sensor output voltage

(TA = -40 to +105°C, 2.4 V ≤ AVREFP ≤ VDD ≤ 5.5 V, VSS = 0 V, reference voltage (+) = AVREFP, reference voltage (-) = AVREFM = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		10	Bit
Overall error ^{Note 1}	AINL	10-bit resolution AVREFP = VDD ^{Note 3} 2.4 V ≤ AVREFP ≤ 5.5 V		1.2	±3.5	LSB
Conversion time	tCONV	10-bit resolution Conversion target: ANI2 to ANI7	3.6 V ≤ VDD ≤ 5.5 V	2.125	39	μs
			2.7 V ≤ VDD ≤ 5.5 V	3.1875	39	μs
			2.4 V ≤ VDD ≤ 5.5 V	17	39	μs
		10-bit resolution Conversion target: Internal reference voltage,	3.6 V ≤ VDD ≤ 5.5 V	2.375	39	μs
		temperature sensor output voltage in HS mode	2.7 V ≤ VDD ≤ 5.5 V	3.5625	39	μs
			2.4 V ≤ VDD ≤ 5.5 V	17	39	μs
Zero-scale error ^{Notes 1, 2}	EZS	10-bit resolution AVREFP = VDD ^{Note 3} 2.4 V ≤ AVREFP ≤ 5.5 V			±0.25	%FSR
Full-scale error ^{Notes 1, 2}	EFS	10-bit resolution AVREFP = VDD ^{Note 3} 2.4 V ≤ AVREFP ≤ 5.5 V			±0.25	%FSR
Integral linearity error ^{Note 1}	ILE	10-bit resolution AVREFP = VDD ^{Note 3} 2.4 V ≤ AVREFP ≤ 5.5 V			±2.5	LSB
Differential linearity error ^{Note 1}	DLE	10-bit resolution AVREFP = VDD ^{Note 3} 2.4 V ≤ AVREFP ≤ 5.5 V			±1.5	LSB
Analog input voltage	VAIN	ANI2 to ANI7	0		AVREFP	V
		Internal reference voltage		VBGR ^{Note 4}		V
		Temperature sensor output voltage		VTMP25 ^{Note 4}		V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When reference voltage (+) = VDD and reference voltage (-) = VSS, the maximum values are as follows.

Overall error: Add ±10 LSB to the maximum value when VDD = AVREFP.

Zero-scale/full-scale error: Add ±0.05%FSR to the maximum value when VDD = AVREFP.

Integral linearity error and differential linearity error: Add ±0.5 LSB to the maximum value when VDD = AVREFP.

Note 4. See **34.6.2 Characteristics of the temperature sensor and internal reference voltage.**

- (2) Reference voltage (+) = $AV_{REFP}/ANI0$ ($ADREFP1 = 0$, $ADREFP0 = 1$), reference voltage (-) = $AV_{REFM}/ANI1$ ($ADREFM = 1$), conversion target: ANI16 to ANI19

(TA = -40 to +105°C, $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.4\text{ V} \leq AV_{REFP} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$,
reference voltage (+) = AV_{REFP} , reference voltage (-) = $AV_{REFM} = 0\text{ V}$)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		10	Bit
Overall error ^{Note 1}	AINL	10-bit resolution $AV_{REFP} = V_{DD}$ ^{Note 3} $2.4\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$		1.2	±5.0	LSB
Conversion time	tCONV	10-bit resolution Conversion target: ANI16 to ANI19 $3.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.125		39	μs
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.1875		39	μs
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	μs
Zero-scale error ^{Notes 1, 2}	EZS	10-bit resolution $AV_{REFP} = V_{DD}$ ^{Note 3} $2.4\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			±0.35	%FSR
Full-scale error ^{Notes 1, 2}	EFS	10-bit resolution $AV_{REFP} = V_{DD}$ ^{Note 3} $2.4\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			±0.35	%FSR
Integral linearity error ^{Note 1}	ILE	10-bit resolution $AV_{REFP} = V_{DD}$ ^{Note 3} $2.4\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			±3.5	LSB
Differential linearity error ^{Note 1}	DLE	10-bit resolution $AV_{REFP} = V_{DD}$ ^{Note 3} $2.4\text{ V} \leq AV_{REFP} \leq 5.5\text{ V}$			±2.0	LSB
Analog input voltage	VAIN	ANI16 to ANI19	0		AV_{REFP}	V

Note 1. This value does not include the quantization error ($\pm 1/2$ LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When $AV_{REFP} < V_{DD}$, the maximum values are as follows.

- Overall error: Add ± 4.0 LSB to the maximum value when $V_{DD} = AV_{REFP}$.
- Zero-scale/full-scale error: Add $\pm 0.20\%$ FSR to the maximum value when $V_{DD} = AV_{REFP}$.
- Integral linearity error/differential linearity error: Add ± 2.0 LSB to the maximum value when $V_{DD} = AV_{REFP}$.

- (3) Reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = V_{SS} (ADREFM = 0), conversion target: ANI0 to ANI7, and ANI16 to ANI19, internal reference voltage, temperature sensor output voltage

(TA = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, reference voltage (+) = V_{DD}, reference voltage (-) = V_{SS})

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES			8		10	Bit
Overall error ^{Note 1}	AINL	10-bit resolution	2.4 V ≤ AV _{REFP} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t _{CONV}	10-bit resolution Conversion target: ANI0 to ANI7, ANI16 to ANI19	3.6 V ≤ V _{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V _{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V _{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Conversion target: Internal reference voltage, temperature sensor output voltage in HS mode	3.6 V ≤ V _{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V _{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V _{DD} ≤ 5.5 V	17		39	μs
Zero-scale error ^{Notes 1, 2}	E _{ZS}	10-bit resolution	2.4 V ≤ V _{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error ^{Notes 1, 2}	E _{FS}	10-bit resolution	2.4 V ≤ V _{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error ^{Note 1}	I _{LE}	10-bit resolution	2.4 V ≤ AV _{REFP} ≤ 5.5 V			±4.0	LSB
Differential linearity error ^{Note 1}	D _{LE}	10-bit resolution	2.4 V ≤ AV _{REFP} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V _{AIN}	ANI0 to ANI7		0		V _{DD}	V
		ANI16 to ANI19		0		V _{DD}	V
		Internal reference voltage		V _{BGR} ^{Note 3}			V
		Temperature sensor output voltage		V _{TMPS25} ^{Note 3}			V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. See 34.6.2 Characteristics of the temperature sensor and internal reference voltage.

- (4) Reference voltage (+) = V_{DD} (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = $AV_{REFM}/ANI1$ (ADREFM = 1), conversion target: ANI0, ANI2 to ANI7, ANI16 to ANI19

(TA = –40 to +105°C, $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$, reference voltage (+) = V_{BGR} ^{Note 3}, reference voltage (–) = AV_{REFM} ^{Note 4} = 0 V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES			8			Bit
Conversion time	tCONV	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	17		39	μs
Zero-scale error ^{Notes 1, 2}	Ezs	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			±0.60	%FSR
Integral linearity error ^{Note 1}	ILE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			±2.0	LSB
Differential linearity error ^{Note 1}	DLE	8-bit resolution	$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			±1.0	LSB
Analog input voltage	VAIN			0		V_{BGR} ^{Note 3}	V

Note 1. This value does not include the quantization error ($\pm 1/2$ LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. See 34.6.2 Characteristics of the temperature sensor and internal reference voltage.

Note 4. When reference voltage (–) = V_{SS} , the maximum values are as follows.

- Zero-scale error: Add $\pm 0.35\%$ FSR to the maximum value when reference voltage (–) = AV_{REFM} .
- Integral linearity error: Add ± 0.5 LSB to the maximum value when reference voltage (–) = AV_{REFM} .
- Differential linearity error: Add ± 0.2 LSB to the maximum value when reference voltage (–) = AV_{REFM} .

34.6.2 Characteristics of the temperature sensor and internal reference voltage

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

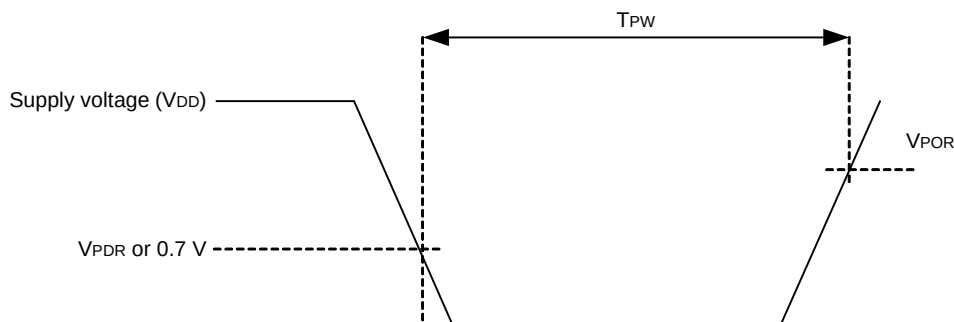
Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Temperature sensor output voltage	VTMPS25	Setting ADS register = 80H, TA = +25°C		1.05		V
Internal reference voltage	VBGR	Setting ADS register = 81H	1.42	1.48	1.54	V
Temperature coefficient	FVTMPS	Temperature dependency of the temperature sensor voltage		-3.3		mV/°C
Operation stabilization wait time	tAMP		5			μs

34.6.3 Characteristics of the POR circuit

(TA = -40 to +105°C, VSS = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	VPOR, VPDR		1.43	1.50	1.57	V
Minimum pulse width ^{Note}	TPW		300			μs

Note This width is the minimum time required for a POR reset when VDD falls below VPDR. This width is also the minimum time required for a POR reset from when VDD falls below 0.7 V to when VDD exceeds VPOR in the STOP mode or while the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



34.6.4 Characteristics of the LVD circuit

(1) LVD0 Detection Voltage in the Reset Mode and Interrupt Mode

(TA = -40 to +105°C, VPDR ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	VLVD00	The power supply voltage is rising.	3.84	3.96	4.08	V
			The power supply voltage is falling.	3.76	3.88	4.00	V
		VLVD01	The power supply voltage is rising.	2.88	2.97	3.06	V
			The power supply voltage is falling.	2.82	2.91	3.00	V
		VLVD02	The power supply voltage is rising.	2.59	2.67	2.75	V
			The power supply voltage is falling.	2.54	2.62	2.70	V
		VLVD03	The power supply voltage is rising.	2.31	2.38	2.45	V
			The power supply voltage is falling.	2.26	2.33	2.40	V
		VLVD04	The power supply voltage is rising.	1.84	1.90	1.95	V
			The power supply voltage is falling.	1.80	1.86	1.91	V
		VLVD05	The power supply voltage is rising.	1.64	1.69	1.74	V
			The power supply voltage is falling.	1.60	1.65	1.70	V
Minimum pulse width		tLW		500			μs
Detection delay time						500	μs

(2) LVD1 Detection Voltage of Reset Mode and Interrupt Mode

(TA = -40 to +105°C, VPDR ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	VLVD10	The power supply voltage is rising.	4.08	4.16	4.24	V
			The power supply voltage is falling.	4.00	4.08	4.16	V
		VLVD11	The power supply voltage is rising.	3.88	3.96	4.04	V
			The power supply voltage is falling.	3.80	3.88	3.96	V
		VLVD12	The power supply voltage is rising.	3.68	3.75	3.82	V
			The power supply voltage is falling.	3.60	3.67	3.74	V
		VLVD13	The power supply voltage is rising.	3.48	3.55	3.62	V
			The power supply voltage is falling.	3.40	3.47	3.54	V
		VLVD14	The power supply voltage is rising.	3.28	3.35	3.42	V
			The power supply voltage is falling.	3.20	3.27	3.34	V
		VLVD15	The power supply voltage is rising.	3.07	3.13	3.19	V
			The power supply voltage is falling.	3.00	3.06	3.12	V
		VLVD16	The power supply voltage is rising.	2.91	2.97	3.03	V
			The power supply voltage is falling.	2.85	2.91	2.97	V
		VLVD17	The power supply voltage is rising.	2.76	2.82	2.87	V
			The power supply voltage is falling.	2.70	2.76	2.81	V
		VLVD18	The power supply voltage is rising.	2.61	2.66	2.71	V
			The power supply voltage is falling.	2.55	2.60	2.65	V
		VLVD19	The power supply voltage is rising.	2.45	2.50	2.55	V
			The power supply voltage is falling.	2.40	2.45	2.50	V
		VLVD110	The power supply voltage is rising.	2.35	2.40	2.45	V
			The power supply voltage is falling.	2.30	2.35	2.40	V
		VLVD111	The power supply voltage is rising.	2.25	2.30	2.34	V
			The power supply voltage is falling.	2.20	2.25	2.29	V
		VLVD112	The power supply voltage is rising.	2.15	2.20	2.24	V
			The power supply voltage is falling.	2.10	2.15	2.19	V
		VLVD113	The power supply voltage is rising.	2.05	2.09	2.13	V
			The power supply voltage is falling.	2.00	2.04	2.08	V
		VLVD114	The power supply voltage is rising.	1.94	1.98	2.02	V
			The power supply voltage is falling.	1.90	1.94	1.98	V
		VLVD115 Note	The power supply voltage is rising.	1.84	1.88	1.91	V
			The power supply voltage is falling.	1.80	1.84	1.87	V
		VLVD116 Note	The power supply voltage is rising.	1.74	1.78	1.81	V
			The power supply voltage is falling.	1.70	1.74	1.77	V
		VLVD117 Note	The power supply voltage is rising.	1.64	1.67	1.70	V
			The power supply voltage is falling.	1.60	1.63	1.66	V
Minimum pulse width		tLW		500			μs
Detection delay time						500	μs

Note This setting can only be used when LVD0 is disabled.

34.6.5 Characteristics of the rising slope of the power supply voltage

(T_A = -40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltage rising slope	SVDD				54	V/ms

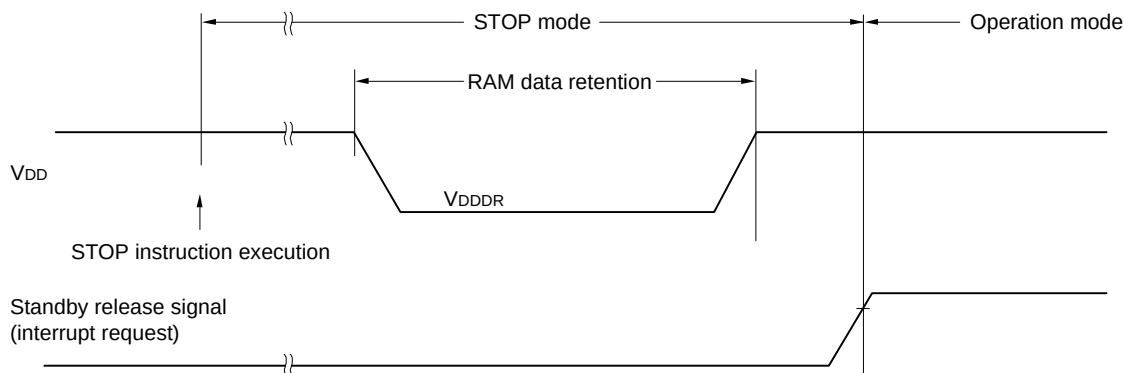
Caution Make sure to keep the internal reset state by the LVD0 circuit or an external reset until V_{DD} reaches the operating voltage range shown in AC characteristics.

34.7 Characteristics of Retention of RAM Data

(TA = -40 to +105°C, VSS = 0V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Data retention supply voltage	VDDDR		1.43 ^{Note}		5.5	V

Note This voltage depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.



34.8 Characteristics of Flash Memory Programming

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
CPU/peripheral hardware clock frequency	fCLK		1		32	MHz
Number of code flash rewrites ^{Notes 1, 2, 3}	Cenwr	Retained for 20 years TA = 85°C	1,000			Times
Number of data flash rewrites ^{Notes 1, 2, 3}		Retained for 1 year TA = 25°C		1,000,000		
		Retained for 5 years TA = 85°C	100,000			
		Retained for 20 years TA = 85°C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite.
The retaining years are until next rewrite after the rewrite.

Note 2. The listed numbers of times apply when using flash memory programmer and Renesas Electronics self-programming library.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

(1) Code flash memory

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbol	fCLK = 1 MHz			fCLK = 2 MHz, 3 MHz			4 MHz ≤ fCLK < 8 MHz			8 MHz ≤ fCLK < 32 MHz			fCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4 bytes	tp4	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	μs
Erasure time	2 Kbytes	te2K	—	10.4	312.2	—	7.7	258.5	—	6.4	231.8	—	5.8	218.4	—	5.6	214.4	ms
Blank checking time	4 bytes	tbc4	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	μs
	2 Kbytes	tbc2K	—	—	2618.9	—	—	1309.5	—	—	658.3	—	—	332.8	—	—	234.1	μs
Time taken to forcibly stop the erasure		tSED	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	μs
Security setting time		tAWSSAS	—	18.2	526.2	—	14.4	469.2	—	12.5	441.1	—	11.6	427.1	—	11.3	422.6	ms
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	μs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

(2) Data flash memory

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbol	fCLK = 1 MHz			fCLK = 2 MHz, 3 MHz			4 MHz ≤ fCLK < 8 MHz			8 MHz ≤ fCLK < 32 MHz			fCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	tp4	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	μs
Erasure time	256 bytes	te2K	—	7.8	259.2	—	6.4	232.0	—	5.8	218.5	—	5.5	211.8	—	5.4	209.7	ms
Blank checking time	1 byte	tbc4	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	μs
	256 bytes	tbc2K	—	—	1326.1	—	—	663.1	—	—	335.1	—	—	171.2	—	—	121.0	μs
Time taken to forcibly stop the erasure		tSED	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	μs
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	μs
Time until reading starts following setting DFLEN to 1		—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	μs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

34.9 Dedicated Flash Memory Programmer Communication (UART)

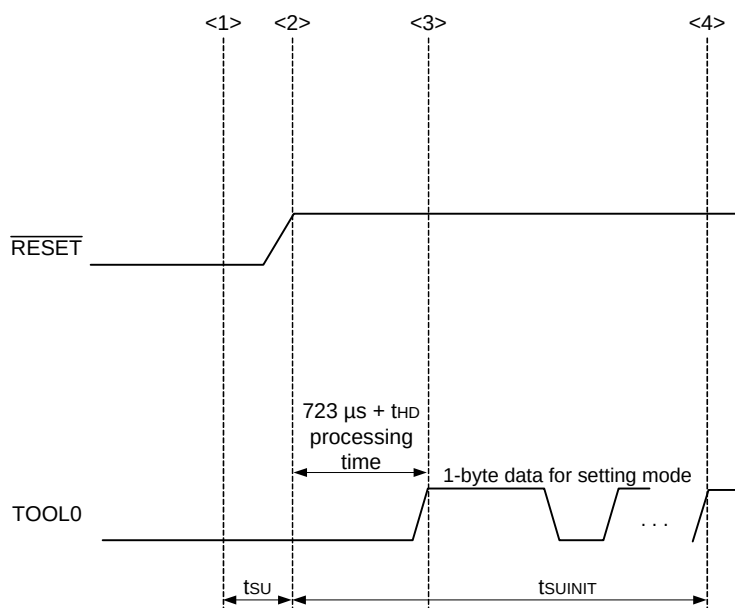
(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

34.10 Timing of Entry to Flash Memory Programming Modes

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Time to complete the communication for the initial setting after the external reset is released	tsuINIT	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	tsu	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (the processing time of the firmware to control the flash memory is not included)	tHD	POR and LVD reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released. Note that the POR and LVD reset must be released before the external reset is released.

<3> The TOOL0 pin is set to the high level.

<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark tsuINIT: The time during which the communications for the initial setting must be completed within 100 ms after the external reset is released.

tsu: Time to release the external reset after the TOOL0 pin is set to the low level

tHD: Time to hold the TOOL0 pin at the low level after the external reset is released. It does not include the processing time of the firmware to control the flash memory.

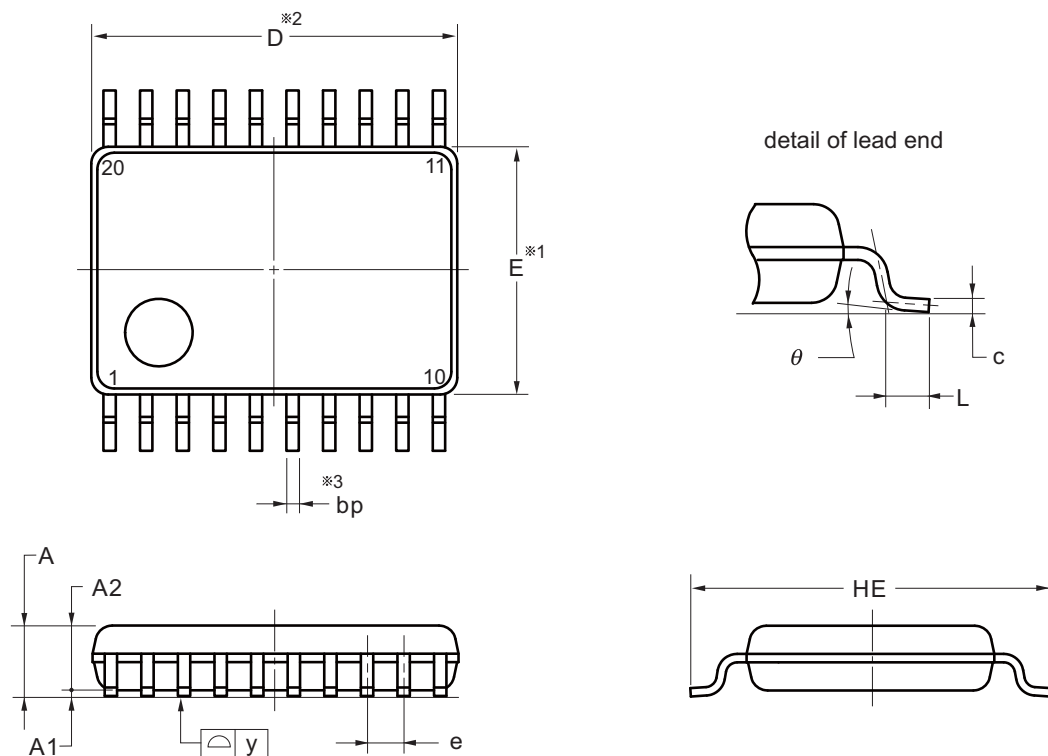
Section 35 Package Drawings

35.1 16-pin Products

The products are currently under planning. Contact a Renesas Electronics sales office for details.

35.2 20-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	P20MA-65-NAA-1	0.1



NOTE

1. Dimensions "※1" and "※2" do not include mold flash.
2. Dimension "※3" does not include tin offset.

(UNIT:mm)	
ITEM	DIMENSIONS
D	6.50±0.10
E	4.40±0.10
HE	6.40±0.20
A	1.45 MAX.
A1	0.10±0.10
A2	1.15
e	0.65±0.12
bp	0.22 ^{+0.10} _{-0.05}
c	0.15 ^{+0.05} _{-0.02}
L	0.50±0.20
y	0.10
θ	0° to 10°

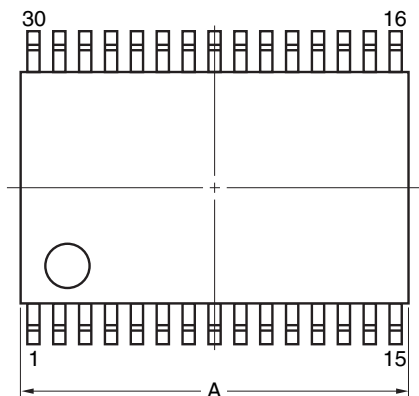
©2012 Renesas Electronics Corporation. All rights reserved.

35.4 25-pin Products

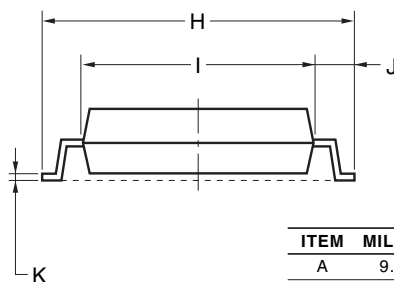
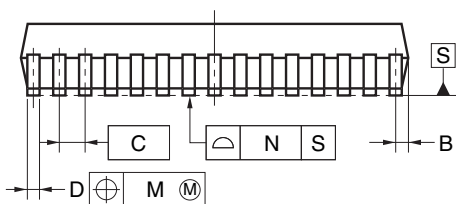
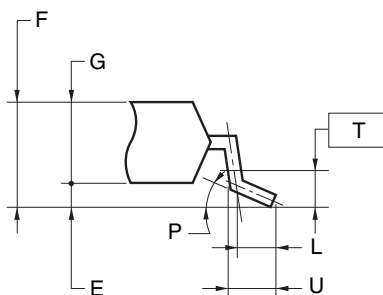
The products are currently under planning. Contact a Renesas Electronics sales office for details.

35.5 30-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



detail of lead end



NOTE

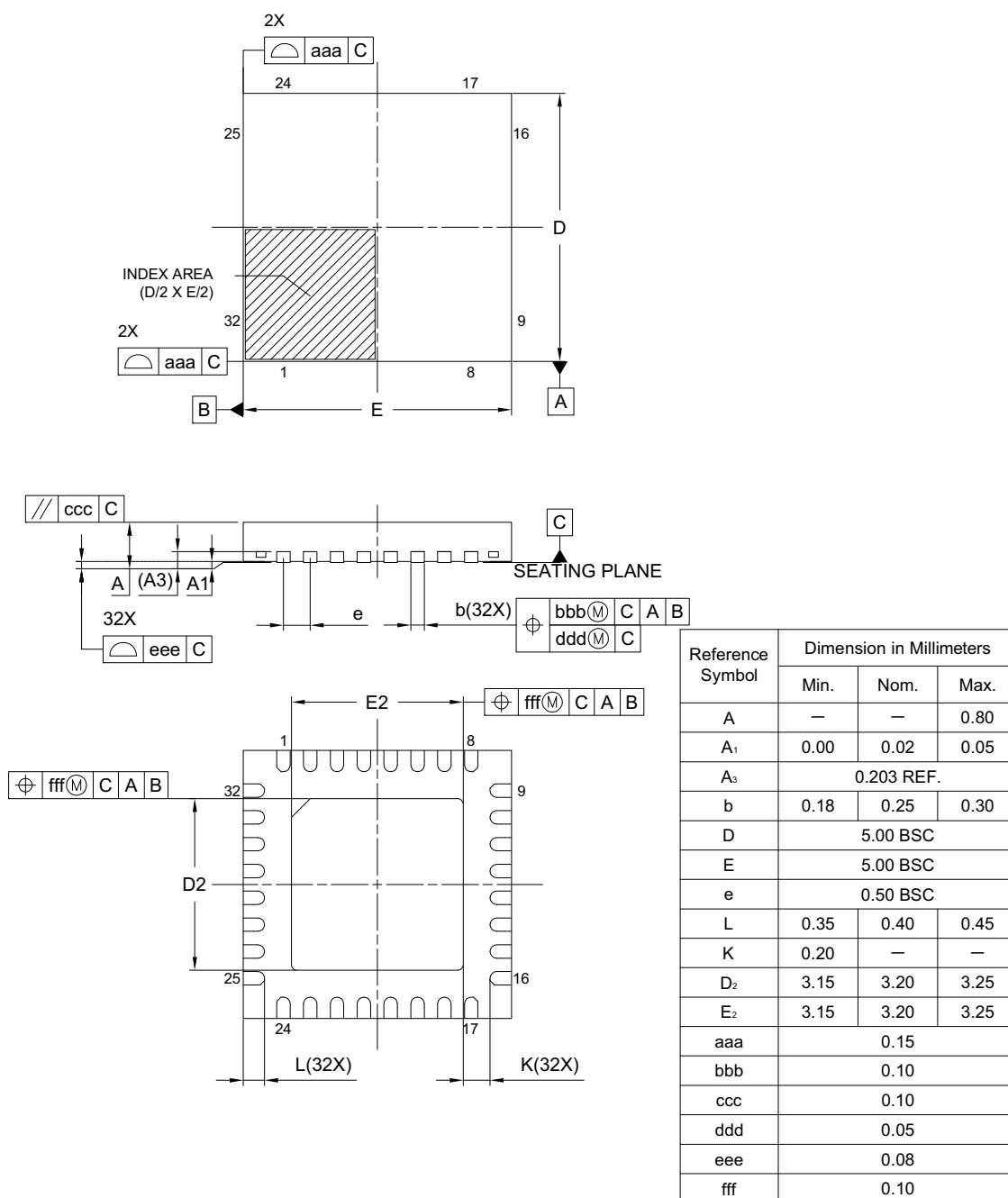
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	0.24 ^{+0.08} _{-0.07}
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25
U	0.6±0.15

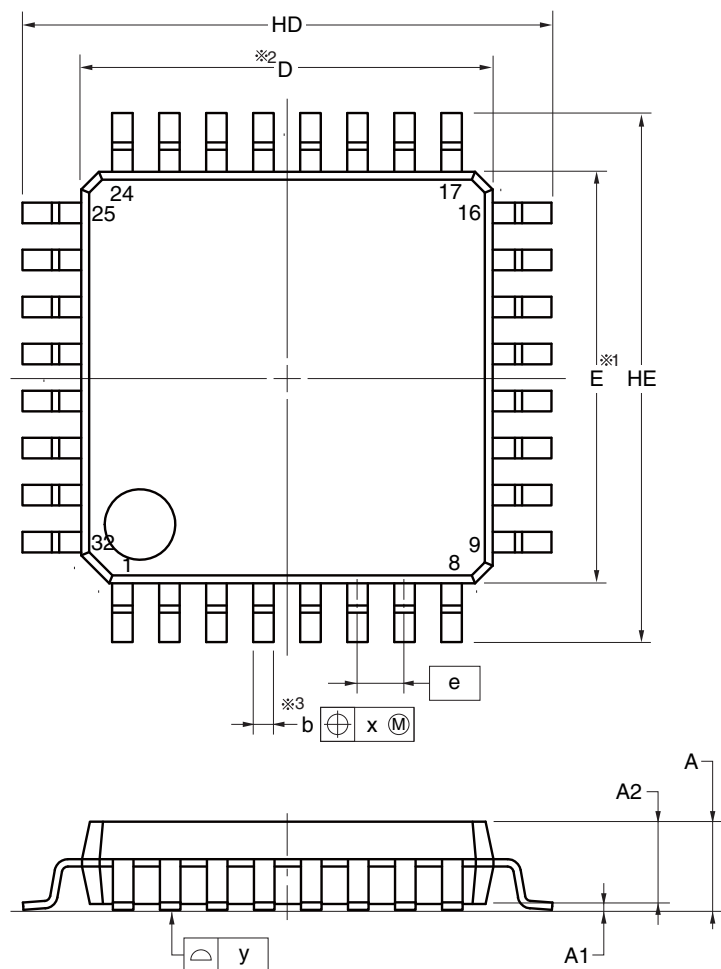
©2012 Renesas Electronics Corporation. All rights reserved.

35.6 32-pin Products

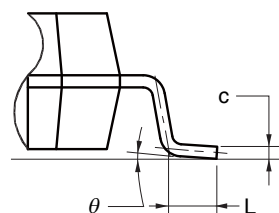
JEITA Package code	RENESAS code	MASS (TYP.) [g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06



JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

NOTE

1. Dimensions "※1" and "※2" do not include mold flash.
2. Dimension "※3" does not include trim offset.

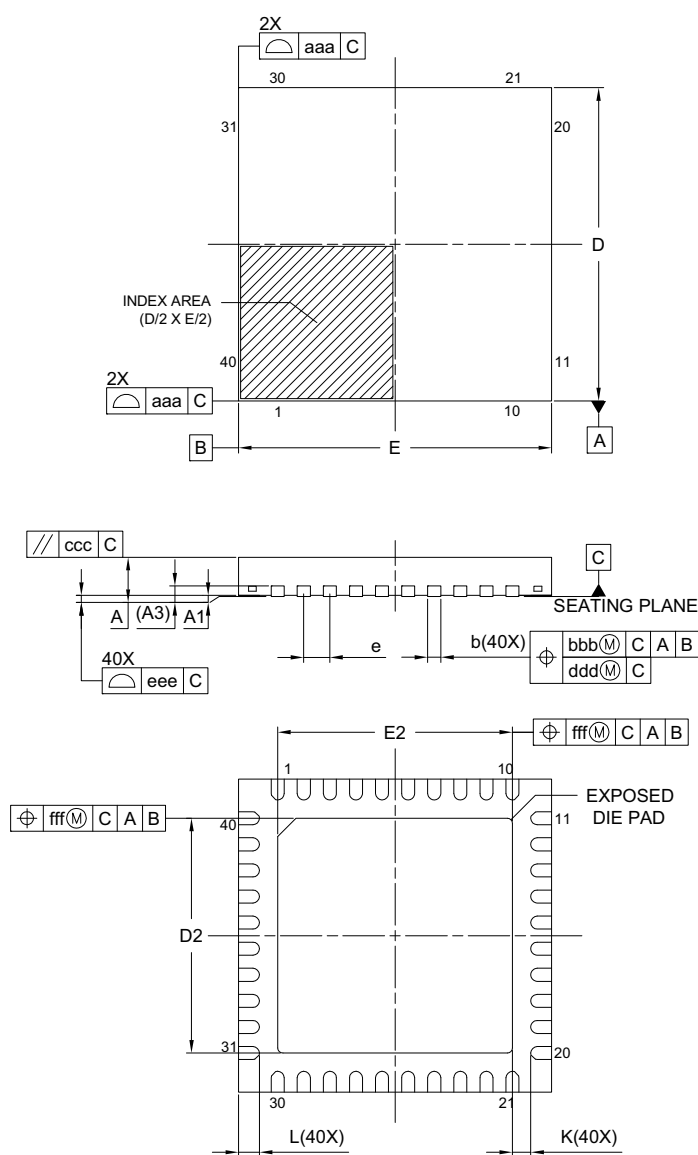
© 2012 Renesas Electronics Corporation. All rights reserved.

35.7 36-pin Products

The products are currently under planning. Contact a Renesas Electronics sales office for details.

35.8 40-pin Products

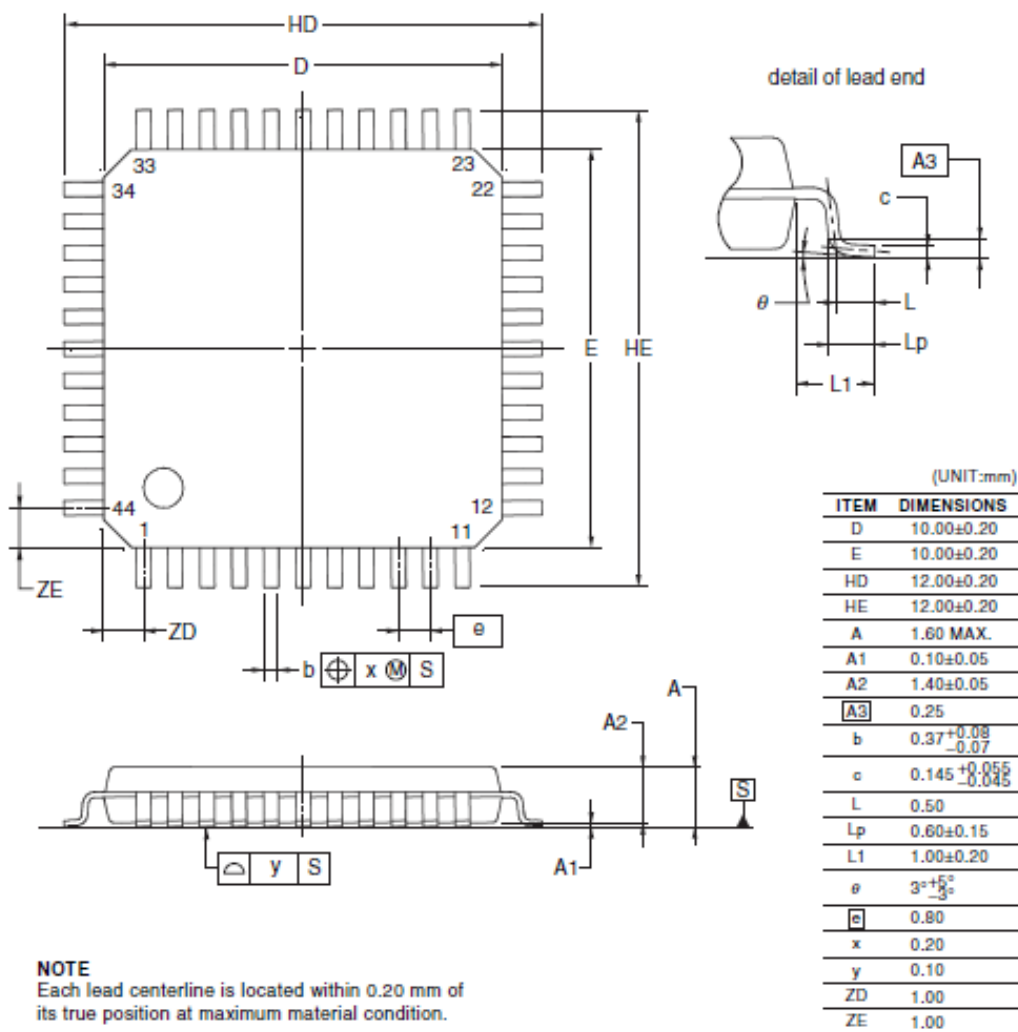
JEITA Package code	RENESAS code	MASS (TYP.) [g]
P-HWQFN040-6x6-0.50	PWQN0040KD-A	0.08



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A ₁	0.00	0.02	0.05
A ₃	0.203 REF.		
b	0.18	0.25	0.30
D	6.00 BSC		
E	6.00 BSC		
e	0.50 BSC		
L	0.30	0.40	0.50
K	0.20	—	—
D ₂	4.45	4.50	4.55
E ₂	4.45	4.50	4.55
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

35.9 44-pin Products

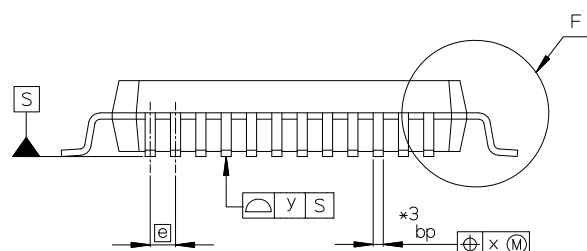
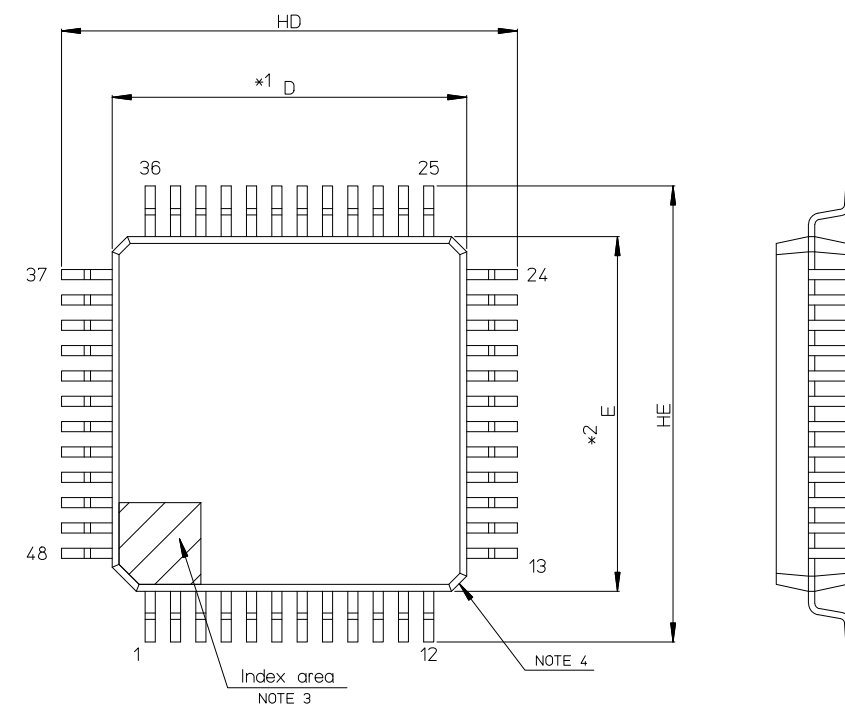
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP44-10x10-0.80	PLQP0044GC-A	P44GB-80-UES-2	0.36



©2012 Renesas Electronics Corporation. All rights reserved.

35.10 48-pin Products

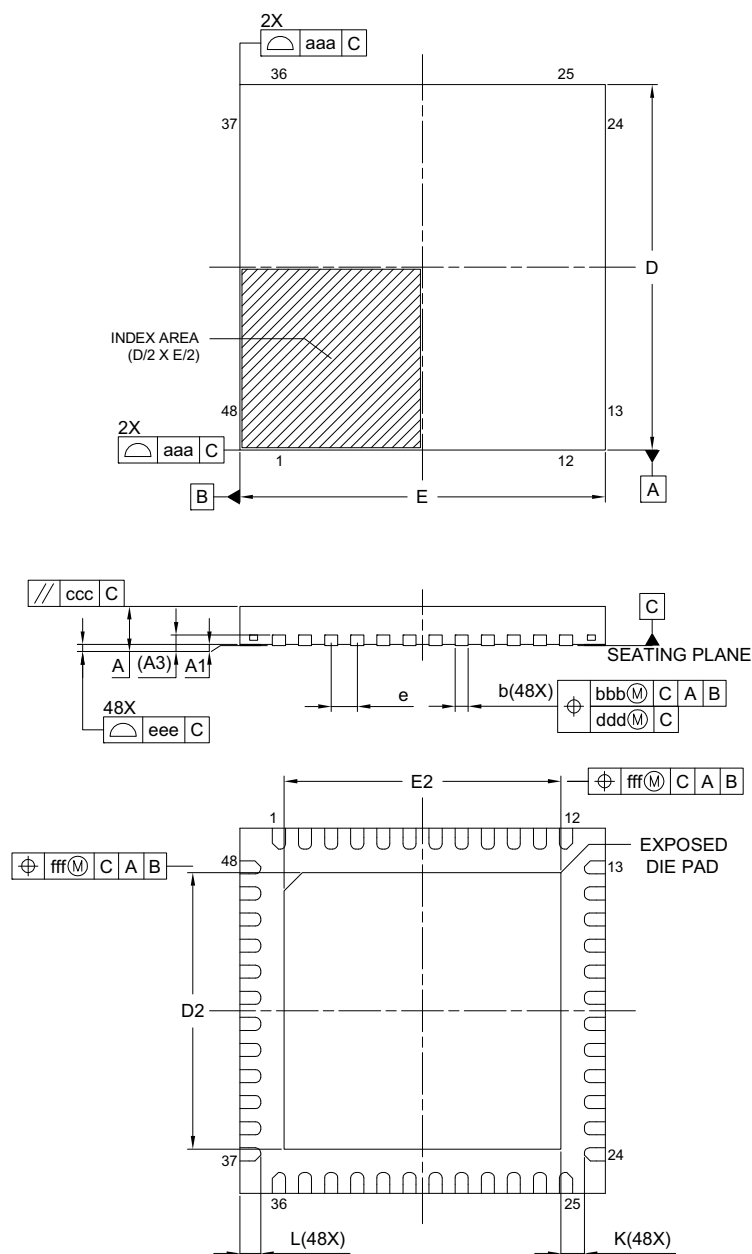
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2g



- 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	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	—	1.4	—
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	—	—	1.7
A1	0.05	—	0.15
bp	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
[e]	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Lp	0.45	0.6	0.75
L1	—	1.0	—

JEITA Package code	RENESAS code	MASS (TYP.) [g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A ₁	0.00	0.02	0.05
A ₃	0.203 REF.		
b	0.20	0.25	0.30
D	7.00 BSC		
E	7.00 BSC		
e	0.50 BSC		
L	0.30	0.40	0.50
K	0.20	—	—
D ₂	5.25	5.30	5.35
E ₂	5.25	5.30	5.35
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Appendix A Revision History

A.1 Major Revisions in This Edition

Edition	Description	Section
Rev.0.50	First edition issued	All

RL78/G22 User's Manual: Hardware

Publication Date: Rev.0.50 Feb 10, 2022

Published by: Renesas Electronics Corporation

RL78/G22