

RL78/L23

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

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/L23 and design and develop application systems and programs for these devices. The target products are as follows.

- 44-pin: R7F100LFx (x = E, G, J, L)
- 48-pin: R7F100LGx (x = E, G, J, L)
- 52-pin: R7F100LJx (x = E, G, J, L)
- 64-pin: R7F100LLx (x = E, G, J, L)
- 80-pin: R7F100LMx (x = G, J, L)
- 100-pin: R7F100LPx (x = G, J, L)

Purpose

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

Organization

The RL78/L23 manual is separated into two parts: this manual and User's Manual: Software (common to the RL78 family).

**RL78/L23
User's Manual:
Hardware
(This Manual)**

- Pin functions
- Internal block functions
- Interrupts
- Other on-chip peripheral functions
- Electrical characteristics

**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/L23 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 numbers are represented as xxxx or xxxxB, where each x is 0 or 1.

Decimal numbers are represented as xxxx, where each x is a numeral from 0 to 9.

Hexadecimal numbers are represented as xxxxH, where each x is a number from 0 to 9 or a letter from A to F.

Related Documents

Documents Related to Devices

Document Name	Document No.
RL78/L23 User's Manual: Hardware	This manual
RL78 Family User's Manual: Software	R01US0015E

Documents (User's Manual) 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	44-pin products	7
1.3.2	48-pin products	11
1.3.3	52-pin products	15
1.3.4	64-pin products	19
1.3.5	80-pin products	24
1.3.6	100-pin products	29
1.4	Pin Identification	41
1.5	Block Diagram	42
1.6	Outline of Functions	43
2.	Pin Functions	47
2.1	Functions of Port Pins	47
2.1.1	44-pin products	48
2.1.2	48-pin products	51
2.1.3	52-pin products	54
2.1.4	64-pin products	57
2.1.5	80-pin products	61
2.1.6	100-pin products	65
2.2	Pin Functions Other than Port Pin Functions	71
2.2.1	Functions for each product	71
2.2.2	Description of pin functions	79
2.2.3	VBAT _n pins (n = 0, 1)	81
2.3	Connection of Unused Pins	84
2.4	Block Diagrams of Pins	86
3.	CPU Architecture	139
3.1	Memory Space	140
3.1.1	Internal program memory space	147
3.1.2	Mirror area	152
3.1.3	Internal data memory space	154
3.1.4	Special function register (SFR) area	155
3.1.5	Extended special function register (2nd SFR: 2nd special function register) area	155
3.1.6	Data memory addressing	156
3.2	Processor Registers	157
3.2.1	Control registers	157
3.2.2	General-purpose registers	159
3.2.3	ES and CS registers	160
3.2.4	Special function registers (SFRs)	161
3.2.5	Extended special function registers (2nd SFRs)	165
3.3	Instruction Address Addressing	185
3.3.1	Relative addressing	185

3.3.2	Immediate addressing	185
3.3.3	Table indirect addressing	186
3.3.4	Register indirect addressing	187
3.4	Addressing for Processing Data Addresses	188
3.4.1	Implied addressing	188
3.4.2	Register addressing	189
3.4.3	Direct addressing	190
3.4.4	Short direct addressing	191
3.4.5	SFR addressing	192
3.4.6	Register indirect addressing	193
3.4.7	Based addressing	194
3.4.8	Based indexed addressing	197
3.4.9	Stack addressing	198
4.	Port Functions	201
4.1	Port Functions	201
4.2	Port Configuration	201
4.2.1	Port 0	202
4.2.2	Port 1	202
4.2.3	Port 2	203
4.2.4	Port 3	203
4.2.5	Port 4	204
4.2.6	Port 5	204
4.2.7	Port 6	205
4.2.8	Port 7	205
4.2.9	Port 8	206
4.2.10	Port 9	206
4.2.11	Port 12	207
4.2.12	Port 13	207
4.2.13	Port 14	208
4.3	Registers for Controlling the Port Function	209
4.3.1	Port mode registers (PMxx)	213
4.3.2	Port registers (Pxx)	214
4.3.3	Pull-up resistor option registers (PUxx)	215
4.3.4	Port input mode registers (PIMxx)	216
4.3.5	Port output mode registers (POMxx)	217
4.3.6	Port digital input disable registers (PDIDISxx)	218
4.3.7	Port mode control A registers (PMCAxx)	219
4.3.8	Port mode control T registers (PMCTxx)	220
4.3.9	Port mode control E registers (PMCExx)	221
4.3.10	Peripheral I/O redirection registers (PIORx)	222
4.3.11	Output current control enable register (CCDE)	228
4.3.12	Output current select registers (CCSx)	230
4.3.13	40-mA port output control register (PTDC)	231
4.3.14	Port function output enable registers (PFOEx)	233
4.3.15	Port mode select register (PMS)	234
4.3.16	LCD port function registers (PFSEGx)	235
4.3.17	LCD input switch control register (ISCLCD)	236

4.4	Port Function Operations	237
4.4.1	Writing to I/O port	237
4.4.2	Reading from I/O port	237
4.4.3	Operations on I/O port	237
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	238
4.5	Register Settings When Using Alternate Function	241
4.5.1	Basic concept when using alternate function	241
4.5.2	Register settings for alternate function whose output function is not used	243
4.5.3	Register settings and port pin state	245
4.5.4	Examples of register settings for port and alternate functions	245
4.6	Cautions When Using Port Function	291
4.6.1	Cautions on 1-bit manipulation instruction for port register m (Pm)	291
4.6.2	Notes on specifying the pin settings	292
5.	Operation State Control	293
5.1	Configuration of Operation State Control	293
5.2	Registers for Controlling the Operation State Control	295
5.2.1	Flash operating mode select register (FLMODE)	295
5.2.2	Flash operating mode protect register (FLMWRP)	297
5.3	Initial Setting of Flash Operation Modes	298
5.4	Transitions between Flash Operation Modes	299
5.5	Details of Flash Operation Modes	300
5.5.1	Details of HS (high-speed main) mode	300
5.5.2	Details of LS (low-speed main) mode	302
5.5.3	Details of LP (low-power main) mode	304
5.5.4	Details on SUB mode	305
6.	Clock Generator	306
6.1	Functions of Clock Generator	306
6.2	Configuration of Clock Generator	308
6.3	Registers for Controlling the Clock Generator	311
6.3.1	Clock operation mode control register (CMC)	312
6.3.2	Main oscillator oscillation mode select register (MODRV)	314
6.3.3	System clock control register (CKC)	315
6.3.4	Clock operation status control register (CSC)	317
6.3.5	Oscillation stabilization time counter status register (OSTC)	318
6.3.6	Oscillation stabilization time select register (OSTS)	320
6.3.7	Peripheral enable registers 0 to 2 (PER0 to PER2)	322
6.3.8	Subsystem clock supply mode control register (OSMC)	329
6.3.9	Subsystem clock select register (CKSEL)	331
6.3.10	High-speed on-chip oscillator frequency select register (HOCODIV)	332
6.3.11	Middle-speed on-chip oscillator frequency select register (MOCODIV)	333
6.3.12	High-speed system clock division register (MOSCDIV)	334
6.3.13	High-speed on-chip oscillator trimming register (HIOTRM)	335
6.3.14	Middle-speed on-chip oscillator trimming register (MIOTRM)	336
6.3.15	Low-speed on-chip oscillator trimming register (LIOTRM)	337
6.3.16	Standby mode release setting register (WKUPMD)	338
6.3.17	XT1 oscillator margin checking control register (SOMRG)	339

6.4	System Clock Oscillator	340
6.4.1	X1 oscillator	340
6.4.2	XT1 oscillator	340
6.4.3	High-speed on-chip oscillator	344
6.4.4	Middle-speed on-chip oscillator	344
6.4.5	Low-speed on-chip oscillator	344
6.5	Operations of the Clock Generator	345
6.6	Controlling Clocks	347
6.6.1	Example of setting the high-speed on-chip oscillator	347
6.6.2	Example of setting the X1 oscillator clock	349
6.6.3	Example of setting the XT1 oscillator clock	350
6.6.4	State transitions of the CPU clock	351
6.6.5	Conditions before changing the CPU clock and processing after changing the CPU clock	355
6.6.6	Time required for switchover of the CPU clock and main system clock	359
6.6.7	Conditions before clock oscillation is stopped	361
6.7	Resonator and Oscillator Constants	362
7.	Timer Array Unit (TAU)	363
7.1	Functions of Timer Array Unit	364
7.1.1	Independent channel operation function	364
7.1.2	Simultaneous channel operation function	366
7.1.3	8-bit timer operation function (channels 1 and 3 only)	368
7.1.4	LIN-bus supporting function (channel 7 of unit 0 only)	368
7.2	Configuration of Timer Array Unit	369
7.2.1	Timer counter register mn (TCRmn)	375
7.2.2	Timer data register mn (TDRmn)	377
7.3	Registers for Controlling the Timer Array Unit	378
7.3.1	Peripheral enable register 0 (PER0)	379
7.3.2	Peripheral reset control register 0 (PRR0)	380
7.3.3	Timer clock select register m (TPSm)	381
7.3.4	Timer mode register mn (TMRmn)	385
7.3.5	Timer status register mn (TSRmn)	389
7.3.6	Timer channel enable status register m (TEm)	390
7.3.7	Timer channel start register m (TSM)	391
7.3.8	Timer channel stop register m (TTm)	393
7.3.9	Timer input select register 0 (TIS0)	394
7.3.10	Timer input select register 1 (TIS1)	395
7.3.11	Timer output select register (TOS)	395
7.3.12	Timer output enable register m (TOEm)	396
7.3.13	Timer output register m (TOM)	397
7.3.14	Timer output level register m (TOLm)	398
7.3.15	Timer output mode register m (TOMm)	399
7.3.16	Input switch control register (ISC)	400
7.3.17	Noise filter enable register 1 (NFEN1)	401
7.3.18	Registers for controlling the port functions multiplexed with the inputs and outputs of timers	403
7.4	Basic Rules of Timer Array Unit	404
7.4.1	Basic rules of simultaneous channel operation function	404

7.4.2	Basic rules of 8-bit timer operation function (channels 1 and 3 only)	406
7.5	Operations of Counters	407
7.5.1	Counter clock (f _{TCLK})	407
7.5.2	Timing of the start of counting	409
7.5.3	Operations of Counters	410
7.6	Channel Output (TOMn Pin) Control	415
7.6.1	TOMn pin output circuit configuration	415
7.6.2	TOMn pin output setting	416
7.6.3	Cautions on channel output operation	417
7.6.4	Collective manipulation of TOMn bit	423
7.6.5	Timer interrupts and TOMn outputs when counting is started	424
7.7	Timer Input (TImn) Control	425
7.7.1	TImn input circuit configuration	425
7.7.2	Noise filter	425
7.7.3	Cautions on channel input operation	426
7.8	Independent Channel Operation Function of Timer Array Unit	427
7.8.1	Operation as an interval timer or for square wave output	427
7.8.2	Operation as an external event counter	431
7.8.3	Operation as a frequency divider (channel 0 only)	435
7.8.4	Operation for input pulse interval measurement	440
7.8.5	Operation for input signal high-/low-level width measurement	445
7.8.6	Operation as a delay counter	449
7.9	Simultaneous Channel Operation Function of Timer Array Unit	453
7.9.1	Operation for the one-shot pulse output function	453
7.9.2	Operation for the PWM function	460
7.9.3	Operation for the multiple PWM output function	467
7.9.4	Remote-control output function	475
7.10	Cautions When Using Timer Array Unit	478
7.10.1	Cautions when using timer output	478
7.10.2	Point for caution when a timer output is to be used as an event input for the ELCL	478
8.	Timer RJ	479
8.1	Functions of Timer RJn	479
8.2	Configuration of Timer RJn	480
8.3	Registers for Controlling Timer RJ	481
8.3.1	Peripheral enable register 2 (PER2)	482
8.3.2	Peripheral reset control register 2 (PRR2)	483
8.3.3	Timer RJ counter registers n (TRJn) (n = 0, 1)	484
8.3.4	Timer RJ control registers n (TRJCRn) (n = 0, 1)	485
8.3.5	Timer RJ I/O control registers n (TRJIOCn) (n = 0, 1)	487
8.3.6	Timer RJ mode registers n (TRJMRn) (n = 0, 1)	489
8.3.7	Timer RJ event pin select registers n (TRJISRn) (n = 0, 1)	490
8.3.8	Registers for controlling the port functions multiplexed with the inputs and outputs of timer RJ	491
8.4	Timer RJn Operation	492
8.4.1	Reload register and counter rewrite operation	492
8.4.2	Timer mode	493
8.4.3	Pulse output mode	494
8.4.4	Event counter mode	495

8.4.5	Pulse width measurement mode	497
8.4.6	Pulse period measurement mode	498
8.4.7	Coordination with the logic and event link controller (ELCL)	500
8.4.8	Output settings for each mode	501
8.5	Points for Caution when Timer R _{Jn} is to be Used	502
8.5.1	Control over starting and stopping of the counter	502
8.5.2	Access to flags (bits TEDGF _n and TUNDF _n in TRJCR _n register)	502
8.5.3	Access to counter register	502
8.5.4	When changing mode	503
8.5.5	Procedure for setting pins TRJOn and TRJIO _n	503
8.5.6	When timer R _{Jn} is not used	503
8.5.7	When timer R _{Jn} operating clock is stopped	503
8.5.8	Procedures for enabling and disabling operation in the event counter mode while the CPU is in the STOP mode	504
8.5.9	Functional restriction in STOP mode (event counter mode only)	504
8.5.10	When counting is forcibly stopped by the TSTOP _n bit	504
8.5.11	Digital filter	504
8.5.12	When selecting f _{IL} as the source for counting	504
9.	16-bit Timers KB40, KB41, and KB42	505
9.1	Functions of 16-bit Timers KB40, KB41, and KB42	505
9.2	Configuration of 16-bit Timers KB40, KB41, and KB42	507
9.3	Registers for Controlling 16-bit Timers KB40, KB41, and KB42	509
9.3.1	Peripheral enable register 2 (PER2)	510
9.3.2	Peripheral reset control register 2 (PRR2)	511
9.3.3	Timer clock select register 2 (TPS2)	512
9.3.4	16-bit timer KB compare register nm (TKBCR _{nm}) (n = 0 to 2, m = 0 to 3)	513
9.3.5	16-bit timer KB trigger compare register n (TKBTGCR _n) (n = 0 to 2)	513
9.3.6	16-bit timer KB operation control register n0 (TKBCTL _{n0}) (n = 0 to 2)	514
9.3.7	16-bit timer KB operation control register n1 (TKBCTL _{n1}) (n = 0 to 2)	516
9.3.8	16-bit timer KB operation control register n2 (TKBCTL _{n2}) (n = 0 to 2)	517
9.3.9	16-bit timer KB output control register n0 (TKBIOC _{n0}) (n = 0 to 2)	518
9.3.10	16-bit timer KB output control register n1 (TKBIOC _{n1}) (n = 0 to 2)	519
9.3.11	16-bit timer KB flag register n (TKBFLG _n) (n = 0 to 2)	520
9.3.12	16-bit timer KB trigger register n (TKBTRG _n) (n = 0 to 2)	522
9.3.13	16-bit timer KB flag clear trigger register n (TKBCLR _n) (n = 0 to 2)	523
9.3.14	16-bit timer KB dithering count register np (TKBDNR _{np}) (n = 0 to 2; p = 0, 1)	524
9.3.15	16-bit timer KB compare 1L & dithering count register n0 (TKBCRLD _{n0}) (n = 0 to 2)	526
9.3.16	16-bit timer KB compare 3L & dithering count register n1 (TKBCRLD _{n1}) (n = 0 to 2)	526
9.3.17	16-bit timer KB smooth start initial duty register np (TKBSIR _{np}) (n = 0 to 2; p = 0, 1)	527
9.3.18	16-bit timer KB smooth start step width register np (TKBSSR _{np}) (n = 0 to 2; p = 0, 1)	527
9.3.19	16-bit timer KB maximum frequency limit setting register n (TKBMFR _n) (n = 0 to 2)	528
9.3.20	16-bit timer KB skipping control register n (TKBTCTL _n) (n = 0 to 2)	529
9.3.21	16-bit timer KB times-of-skipping setting register n (TKBTCMP _n) (n = 0 to 2)	530
9.3.22	External interrupt control register x (INTPCTL _x) (x = 0 to 7)	531
9.3.23	Registers for controlling the port functions multiplexed with the outputs of 16-bit timers KB40, KB41, and KB42	533
9.4	Operations of 16-bit Timers KB40, KB41, and KB42	534
9.4.1	Basic counter operation	537

9.4.2	Default level and active level	537
9.4.3	Stopping and restarting operation	543
9.4.4	Batch overwriting	546
9.4.5	Skipping of interrupt requests and A/D conversion triggers	547
9.4.6	Standalone mode (with the period controlled by the TKBCRn0 register)	550
9.4.7	Standalone mode (with the period controlled by external trigger input)	556
9.4.8	Simultaneous start/stop mode	564
9.4.9	Simultaneous start/clear mode	572
9.4.10	Interleaved power factor correction (PFC) output mode	581
9.5	Optional Functions of 16-bit Timers KB40, KB41, and KB42	604
9.5.1	A/D conversion start timing signal output	605
9.5.2	PWM output dithering	607
9.5.3	PWM output smooth start	611
9.5.4	Maximum frequency limiting	614
9.5.5	Masking of external trigger inputs while the TKBOnp output is active	623
9.5.6	PWM output for IH control	626
9.6	Forced Output Stop	628
9.6.1	Forced output stop 1 and 2	629
9.6.2	Configuration of hardware for forced output stop	631
9.6.3	Registers for controlling forced output stop	632
9.6.3.1	Forced output stop function control register np (TKBPACTLn _p) (n = 0 to 2; p = 0, 1)	633
9.6.3.2	Forced output stop function control register n2 (TKBPACTLn ₂) (n = 0 to 2)	635
9.6.3.3	Forced output stop function 1 input select register np (TKBPAHZSn _p) (n = 0 to 2; p = 0, 1)	636
9.6.3.4	Forced output stop function 2 input select register np (TKBPAFXSn _p) (n = 0 to 2, p = 0, 1)	638
9.6.3.5	Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2)	640
9.6.3.6	Forced output stop function start trigger register n (TKBPAHFSn) (n = 0 to 2)	641
9.6.3.7	Forced output stop function cancel trigger register n (TKBPAHFTn) (n = 0 to 2)	642
9.7	Operation with Forced Output Stop 1	643
9.7.1	Summary of forced output stop 1	643
9.7.2	Start and cancellation by software of forced output stop 1	644
9.7.3	Basic operation with forced output stop 1	645
9.8	Operation with Forced Output Stop 2	649
9.8.1	Summary of forced output stop 2	649
9.8.2	Basic operation with forced output stop 2	650
9.9	Notes on Using Interlocking of External Interrupts INTPx with 16-bit Timers KB40, KB41, and KB42	654
10.	Realtime Clock (RTC)	655
10.1	Functions of Realtime Clock	655
10.2	Configuration of the Realtime Clock	656
10.3	Registers for Controlling the Realtime Clock	658
10.3.1	Peripheral enable register 0 (PER0)	659
10.3.2	Subsystem clock supply mode control register (OSMC)	660
10.3.3	Realtime clock control register 0 (RTCC0)	661
10.3.4	Realtime clock control register 1 (RTCC1)	663
10.3.5	Second count register (SEC)	665

10.3.6	Minute count register (MIN)	665
10.3.7	Hour count register (HOUR)	666
10.3.8	Day count register (DAY)	668
10.3.9	Day-of-week count register (WEEK)	669
10.3.10	Month count register (MONTH)	670
10.3.11	Year count register (YEAR)	670
10.3.12	Time error correction register (SUBCUD)	671
10.3.13	Alarm minute register (ALARMWM)	672
10.3.14	Alarm hour register (ALARMWH)	672
10.3.15	Alarm day-of-week register (ALARMWW)	673
10.3.16	Registers for controlling the port functions multiplexed with the realtime clock output	674
10.4	Operations of the Realtime Clock	675
10.4.1	Starting the Realtime Clock Operation	675
10.4.2	Shifting to HALT or STOP mode after starting operation	676
10.4.3	Reading from and writing to the counters of the realtime clock	677
10.4.4	Setting alarm by the realtime clock	679
10.4.5	1 Hz output by the realtime clock	680
10.4.6	Example of time error correction by the realtime clock	681
11.	8-bit Interval Timers	686
11.1	Overview	686
11.2	I/O Pins	688
11.3	Registers for Controlling the 8-bit Interval Timers	688
11.3.1	8-bit interval timer counter registers ni (TRTni)	689
11.3.2	8-bit interval timer counter register n (TRTn)	689
11.3.3	8-bit interval timer compare registers ni (TRTCMPni)	690
11.3.4	8-bit interval timer compare register n (TRTCMPn)	690
11.3.5	8-bit interval timer control register n (TRTCRn)	691
11.3.6	8-bit interval timer frequency division register n (TRTMDn)	692
11.4	Operation	693
11.4.1	Interval timer modes	693
11.4.2	Timer operation	694
11.4.3	Start/stop timing	696
11.4.4	Timing for updating compare register values	700
11.4.5	Procedures for Setting an 8-bit Interval Timer	701
11.5	Notes on 8-bit Interval Timers	702
11.5.1	Changing settings of operating modes	702
11.5.2	Accessing compare registers	702
11.5.3	Setting procedure for the 8-bit interval timers	702
12.	32-bit Interval Timer (TML32)	703
12.1	Overview	703
12.2	Registers for Controlling the 32-bit Interval Timer	706
12.2.1	Peripheral enable register 1 (PER1)	707
12.2.2	Peripheral reset control register 1 (PRR1)	708
12.2.3	Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)	709
12.2.4	Interval timer compare register 0n (ITLCMP0n) (n = 0, 1)	709
12.2.5	Interval timer capture register 00 (ITLCAP00)	710
12.2.6	Interval timer control register (ITLCTL0)	711

12.2.7	Interval timer clock select register 0 (ITLCSEL0)	713
12.2.8	Interval timer frequency division register 0 (ITLFDIV00)	714
12.2.9	Interval timer frequency division register 1 (ITLFDIV01)	715
12.2.10	Interval timer capture control register 0 (ITLCC0)	716
12.2.11	Interval timer status register (ITLS0)	718
12.2.12	Interval timer match detection mask register (ITLMKF0)	720
12.3	Operation	721
12.3.1	Settings for counter modes	721
12.3.2	Settings for the capture mode	723
12.3.3	Timer operation	724
12.3.4	Capture operation	725
12.3.5	Interrupt	727
12.3.6	Procedures for Setting the Interval Timer	729
13.	External Signal Sampler	733
13.1	Functions of External Signal Sampler	733
13.2	Configuration of External Signal Sampler	733
13.3	Registers for Controlling the External Signal Sampler	735
13.3.1	Peripheral enable register 2 (PER2)	736
13.3.2	Peripheral reset control register 2 (PRR2)	737
13.3.3	External signal sampler control register 0 (EXSDM0)	738
13.3.4	External signal sampler control register 1 (EXSDM1)	740
13.3.5	Registers for controlling the port functions multiplexed with the inputs and outputs of the external signal sampler	741
13.4	Sampling Clock Output Operation	742
13.5	Operation of External Signal Sampler	743
13.6	Cautions	745
14.	Clock Output/Buzzer Output Controller	746
14.1	Functions of Clock Output/Buzzer Output Controller	746
14.2	Configuration of Clock Output/Buzzer Output Controller	748
14.3	Registers for Controlling the Clock Output/Buzzer Output Controller	748
14.3.1	Clock output select registers n (CKSn)	749
14.3.2	Registers for controlling the port functions multiplexed with the clock or buzzer outputs	751
14.4	Operations of the Clock Output/Buzzer Output Controller	752
14.4.1	Operation of output pins	752
14.5	Point for Caution When the Clock Output/Buzzer Output Controller Is to be Used	752
15.	Watchdog Timer (WDT)	753
15.1	Functions of Watchdog Timer	753
15.2	Configuration of Watchdog Timer	754
15.3	Register for Controlling the Watchdog Timer	755
15.3.1	Watchdog timer enable register (WDTE)	755
15.4	Operation of Watchdog Timer	756
15.4.1	Controlling operation of watchdog timer	756
15.4.2	Setting overflow time of watchdog timer	757
15.4.3	Setting window open period of watchdog timer	758
15.4.4	Setting watchdog timer interval interrupt	760

16.	A/D Converter (ADC)	761
16.1	Function of A/D Converter	761
16.2	Configuration of A/D Converter	764
16.3	Registers for Controlling the A/D Converter	766
16.3.1	Peripheral enable register 0 (PER0)	767
16.3.2	Peripheral reset control register 0 (PRR0)	768
16.3.3	A/D converter mode register 0 (ADM0)	769
16.3.4	A/D converter mode register 1 (ADM1)	781
16.3.5	A/D converter mode extension register 1 (ADM1EX)	783
16.3.6	A/D converter mode register 2 (ADM2)	784
16.3.7	12-bit/10-bit A/D conversion result register (ADCR0)	787
16.3.8	8-bit A/D conversion result register (ADCR0H)	788
16.3.9	Analog input channel specification register (ADS)	789
16.3.10	Conversion result comparison upper limit setting register (ADUL)	791
16.3.11	Conversion result comparison lower limit setting register (ADLL)	791
16.3.12	A/D test register (ADTES)	792
16.3.13	Registers for controlling the port functions multiplexed with the analog inputs of the A/D converter	793
16.4	A/D Converter Operations	794
16.5	Input Voltage and Conversion Results	796
16.6	A/D Converter Operation Modes	797
16.6.1	Software trigger no-wait mode (sequential conversion mode)	797
16.6.2	Software trigger no-wait mode (one-shot conversion mode)	798
16.6.3	Software trigger wait mode (sequential conversion mode)	799
16.6.4	Software trigger wait mode (one-shot conversion mode)	800
16.6.5	Hardware trigger no-wait mode (sequential conversion mode)	801
16.6.6	Hardware trigger no-wait mode (one-shot conversion mode)	802
16.6.7	Hardware trigger wait mode (sequential conversion mode)	803
16.6.8	Hardware trigger wait mode (one-shot conversion mode)	804
16.7	A/D Converter Setup Flowchart	805
16.7.1	Setting up software trigger no-wait mode	805
16.7.2	Setting up software trigger wait mode	806
16.7.3	Setting up hardware trigger no-wait mode	807
16.7.4	Setting up hardware trigger wait mode	808
16.7.5	Example of using the ADC when selecting the temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU, and software trigger no-wait mode and one-shot conversion mode	809
16.7.6	Setting up test mode	810
16.8	SNOOZE Mode	811
16.8.1	A/D conversion by inputting a software trigger	811
16.8.2	A/D conversion by inputting a hardware trigger	813
16.9	How to Read A/D Converter Characteristics Table	816
16.10	Points for Caution when the A/D Converter is to be Used	820
17.	D/A Converter (DAC)	824
17.1	Functions of D/A Converter	824
17.2	Configuration of D/A Converter	825
17.3	Registers for Controlling the D/A Converter	826
17.3.1	Peripheral enable register 1 (PER1)	827

17.3.2	Peripheral reset control register 1 (PRR1)	828
17.3.3	D/A converter mode register 0 (DAM0)	829
17.3.4	D/A converter mode register 1 (DAM1)	830
17.3.5	D/A conversion value setting register i (DACSi) (i = 0 to 2)	831
17.3.6	Registers for controlling the event output from the logic and event link controller	832
17.3.7	Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter	832
17.4	Operations of D/A Converter	833
17.4.1	Operation in normal mode	833
17.4.2	Operation in realtime output mode	834
17.4.3	Timing for outputting D/A conversion value	835
17.5	Points for Caution when the D/A Converter is to be Used	836
18.	Comparator (CMP)	837
18.1	Functions of Comparator	837
18.2	Configuration of Comparator	838
18.3	Registers for Controlling the Comparator	839
18.3.1	Peripheral enable register 1 (PER1)	840
18.3.2	Peripheral reset control register 1 (PRR1)	841
18.3.3	Comparator mode setting register (COMPMDR)	842
18.3.4	Comparator filter control register (COMPFIR)	844
18.3.5	Comparator output control register (COMPOCR)	846
18.3.6	Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator	847
18.4	Operation	848
18.4.1	Digital Filter in Comparator i	850
18.4.2	Comparator i interrupts (i = 0, 1)	850
18.4.3	Event signal output for the logic and event link controller (ELCL)	851
18.4.4	Comparator i output	852
18.4.5	Stopping or supplying comparator clock	852
19.	Serial Array Unit (SAU)	853
19.1	Functions of Serial Array Unit	855
19.1.1	Simplified SPIs (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31)	855
19.1.2	UART (UART0 to UART3)	856
19.1.3	Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)	857
19.2	Configuration of Serial Array Unit	858
19.2.1	Shift Register	861
19.2.2	Lower 8 or 9 bits of the serial data register mn (SDRmn)	861
19.3	Registers for Controlling the Serial Array Unit	863
19.3.1	Peripheral enable register 0 (PER0)	864
19.3.2	Peripheral reset control register 0 (PRR0)	865
19.3.3	Serial clock select register m (SPSm)	866
19.3.4	Serial mode register mn (SMRmn)	867
19.3.5	Serial communication operation setting register mn (SCRmn)	869
19.3.6	Serial data register mn (SDRmn)	871
19.3.7	Serial flag clear trigger register mn (SIRmn)	873
19.3.8	Serial status register mn (SSRmn)	874
19.3.9	Serial channel start register m (SSm)	876

19.3.10	Serial channel stop register m (STm)	877
19.3.11	Serial channel enable status register m (SEm)	878
19.3.12	Serial output enable register m (SOEm)	879
19.3.13	Serial output register m (SOM)	880
19.3.14	Serial output level register m (SOLm)	881
19.3.15	Serial standby control register m (SSCm)	883
19.3.16	Input switch control register (ISC)	885
19.3.17	Noise filter enable register 0 (NFEN0)	887
19.3.18	Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit	888
19.3.19	UART loopback select register (ULBS)	889
19.4	Operation Stop Mode	890
19.4.1	Stopping the Operation by Units	890
19.4.2	Stopping the Operation by Channels	891
19.5	Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication	892
19.5.1	Master Transmission	895
19.5.2	Master Reception	904
19.5.3	Master Transmission/Reception	914
19.5.4	Slave Transmission	924
19.5.5	Slave Reception	933
19.5.6	Slave Transmission/Reception	940
19.5.7	SNOOZE Mode Function	950
19.5.8	Calculating Transfer Clock Frequency	955
19.5.9	Procedure for Processing Errors that Occurred During Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication	957
19.6	Operation of UART (UART0 to UART3) Communication	958
19.6.1	UART Transmission	962
19.6.2	UART Reception	972
19.6.3	SNOOZE Mode	979
19.6.4	Calculating Baud Rate	986
19.6.5	Procedure for Processing Errors that Occurred During UART (UART0 to UART3) Communication	990
19.7	LIN Communication Operation	991
19.7.1	LIN Transmission	991
19.7.2	LIN Reception	994
19.8	Operation of Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) Communication	1000
19.8.1	Address Field Transmission	1003
19.8.2	Data Transmission	1009
19.8.3	Data Reception	1012
19.8.4	Stop Condition Generation	1016
19.8.5	Calculating Transfer Rate	1018
19.8.6	Procedure for Processing Errors that Occurred during Simplified I ² C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) Communication	1021
20.	Serial Interface IICA (IICA)	1022
20.1	Functions of Serial Interface IICA	1022
20.2	Configuration of Serial Interface IICA	1025
20.3	Registers for Controlling Serial Interface IICA	1028

20.3.1	Peripheral enable register 0 (PER0)	1029
20.3.2	Peripheral reset control register 0 (PRR0)	1030
20.3.3	IICA control register n0 (IICCTLn0)	1031
20.3.4	IICA status register n (IICSn)	1036
20.3.5	IICA flag register n (IICFn)	1040
20.3.6	IICA control register n1 (IICCTLn1)	1042
20.3.7	IICA low-level width setting register n (IICWLn)	1044
20.3.8	IICA high-level width setting register n (IICWHn)	1044
20.3.9	Registers for controlling the port functions multiplexed with the IICA inputs and outputs	1045
20.4	I ² C Bus Mode Functions	1046
20.4.1	Pin Configuration	1046
20.4.2	Setting transfer clock by using IICWLn and IICWHn registers	1047
20.5	I ² C Bus Definitions and Control Methods	1049
20.5.1	Start Conditions	1049
20.5.2	Address	1050
20.5.3	Transfer Direction Specification	1050
20.5.4	Acknowledge (ACK)	1051
20.5.5	Stop Condition	1052
20.5.6	Clock stretching	1053
20.5.7	Release from clock stretching	1055
20.5.8	Timing of Generation of the Interrupt Request Signal (INTIICAn) and Control of Clock Stretching	1056
20.5.9	Address Match Detection Method	1058
20.5.10	Error Detection	1058
20.5.11	Extension Code	1059
20.5.12	Arbitration	1060
20.5.13	Wakeup Function	1062
20.5.14	Communication Reservation	1065
20.5.15	Cautions	1069
20.5.16	Communication Operations	1070
20.5.17	Timing of I ² C Interrupt Request Signal (INTIICAn) Occurrence	1078
20.6	Timing Charts	1098
21.	Serial Interface UARTA (UARTA)	1113
21.1	Overview	1113
21.2	Registers for Controlling the Serial Interface UARTA	1115
21.2.1	Peripheral enable registers 1, 2 (PER1, PER2)	1116
21.2.2	Transmit buffer register n (TXBAn)	1117
21.2.3	Receive buffer register n (RXBAn)	1118
21.2.4	Operation mode setting register n0 (ASIMAn0)	1119
21.2.5	Operation mode setting register n1 (ASIMAn1)	1121
21.2.6	Baud rate generator control register n (BRGCAn)	1122
21.2.7	Status register n (ASISAn)	1123
21.2.8	Status clear trigger register n (ASCTAn)	1125
21.2.9	UARTA clock select register n (UTAnCK)	1126
21.2.10	UART loopback select register (ULBS)	1128
21.2.11	Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface	1129

21.3	Operation	1130
21.3.1	Operation stop mode	1130
21.3.2	UART mode	1130
21.3.3	Receive data noise filter	1144
21.3.4	Baud rate generator	1145
21.4	Points for Caution when the Serial Interface UARTA is to be Used	1154
21.4.1	Port setting for RxDAn pin	1154
21.4.2	Serial Interfaces	1154
21.4.3	Point for caution when selecting the UARTAn operation clock (fUTAn)	1154
22.	LCD Controller/Driver	1155
22.1	Functions of LCD Controller/Driver	1157
22.2	Configuration of LCD Controller/Driver	1170
22.3	Registers for Controlling the LCD Controller/Driver	1172
22.3.1	LCD mode register 0 (LCDM0)	1173
22.3.2	LCD mode register 1 (LCDM1)	1175
22.3.3	LCD mode register 2 (LCDM2)	1177
22.3.4	Subsystem clock supply mode control register (OSMC)	1178
22.3.5	LCD clock control register 0 (LCDC0)	1179
22.3.6	LCD boost level control register (VLCD)	1181
22.3.7	LCD input switch control register (ISCLCD)	1184
22.3.8	LCD port function registers (PFSEGx)	1190
22.3.9	Registers for controlling the port functions multiplexed with the common outputs and segment outputs of the LCD controller/driver	1197
22.4	LCD Display Data Registers	1198
22.5	Selection of LCD Display Register	1206
22.5.1	A-pattern area and B-pattern area data display	1207
22.5.2	Alternation of the display (Alternately displaying A-pattern and B-pattern area data)	1207
22.6	Setting the LCD Controller/Driver	1208
22.7	Operation Stop Procedure	1213
22.8	Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4	1214
22.8.1	External resistance division method	1214
22.8.2	Internal voltage boosting method	1216
22.8.3	Capacitor split method	1218
22.9	Common and Segment Signals	1220
22.10	Display Modes	1228
22.10.1	Static display example	1228
22.10.2	Two-time-slice display example	1231
22.10.3	Three-time-slice display example	1234
22.10.4	Four-time-slice display example	1238
22.10.5	Six-time-slice display example	1242
22.10.6	Eight-time-slice display example	1246
23.	Data Transfer Controller (DTC)	1250
23.1	Functions of DTC	1250
23.2	Configuration of DTC	1252
23.3	Registers for Controlling the DTC	1253
23.3.1	Allocation of DTC control data area and DTC vector table area	1254
23.3.2	Control data allocation	1255

23.3.3	Vector table	1257
23.3.4	Peripheral enable register 1 (PER1)	1260
23.3.5	DTC control register j (DTCCRj) (j = 0 to 23)	1261
23.3.6	DTC block size register j (DTBLSj) (j = 0 to 23)	1263
23.3.7	DTC transfer count register j (DTCCTj) (j = 0 to 23)	1264
23.3.8	DTC transfer count reload register j (DTRLdj) (j = 0 to 23)	1264
23.3.9	DTC source address register j (DTSARj) (j = 0 to 23)	1265
23.3.10	DTC destination address register j (DTDARj) (j = 0 to 23)	1265
23.3.11	DTC activation enable register i (DTCENi) (i = 0 to 6)	1266
23.3.12	DTC base address register (DTCBAR)	1269
23.4	DTC Operation	1270
23.4.1	Activation sources	1271
23.4.2	Normal mode	1272
23.4.3	Repeat mode	1275
23.4.4	Chain transfers	1279
23.5	Points for Caution when the DTC is to be Used	1281
23.5.1	Setting DTC control data and vector table	1281
23.5.2	Allocation of DTC control data area and DTC vector table area	1281
23.5.3	DTC pending instruction	1282
23.5.4	Operation when accessing data flash memory space	1282
23.5.5	Number of DTC execution clock cycles	1283
23.5.6	DTC response time	1284
23.5.7	DTC activation sources	1284
23.5.8	Operation in standby mode	1285
24.	Logic and Event Link Controller (ELCL)	1286
24.1	Functions of ELCL	1286
24.2	Configuration of ELCL	1286
24.3	Registers for Controlling the ELCL	1296
24.3.1	Input signal select registers n (ELISELn) (n = 0 to 11)	1297
24.3.2	Event link L1 signal select registers n (ELL1SELn) (n = 0 to 3)	1302
24.3.3	Event link L1 signal select registers n (ELL1SELn) (n = 4, 5)	1304
24.3.4	Event link L1 signal select register 6 (ELL1SEL6)	1305
24.3.5	Logic cell block L1 control register (ELL1CTL)	1306
24.3.6	Event link L1 output select registers n (ELL1LNKn) (n = 0 to 3)	1307
24.3.7	Event link L1 output select registers n (ELL1LNKn) (n = 4, 5)	1308
24.3.8	Event link L1 output select register 6 (ELL1LNK6)	1309
24.3.9	Event link L2 signal select registers n (ELL2SELn) (n = 0 to 3)	1310
24.3.10	Event link L2 signal select registers n (ELL2SELn) (n = 4, 5)	1312
24.3.11	Event link L2 signal select register 6 (ELL2SEL6)	1313
24.3.12	Logic cell block L2 control register (ELL2CTL)	1314
24.3.13	Event link L2 output select registers n (ELL2LNKn) (n = 0 to 3)	1315
24.3.14	Event link L2 output select registers n (ELL2LNKn) (n = 4, 5)	1316
24.3.15	Event link L2 output select register 6 (ELL2LNK6)	1317
24.3.16	Event link L3 signal select registers n (ELL3SELn) (n = 0 to 3)	1318
24.3.17	Event link L3 signal select registers n (ELL3SELn) (n = 4, 5)	1320
24.3.18	Event link L3 signal select register 6 (ELL3SEL6)	1321
24.3.19	Logic cell block L3 control register (ELL3CTL)	1322
24.3.20	Event link L3 output select registers n (ELL3LNKn) (n = 0 to 3)	1323

24.3.21	Event link L3 output select registers n (ELL3LNKn) (n = 4, 5)	1324
24.3.22	Event link L3 output select register 6 (ELL3LNK6)	1325
24.3.23	Output signal select registers n (ELOSELn) (n = 0 to 7)	1326
24.3.24	Output signal enable register (ELOENCTL)	1328
24.3.25	Output signal monitor register (ELOMONI)	1330
24.3.26	Registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller	1331
24.4	ELCL Operation	1332
24.5	Procedure of Setting the ELCL	1333
24.6	Points for Caution when the ELCL is to be Used	1336
25.	Interrupt Functions	1337
25.1	Types of Interrupt	1337
25.2	Interrupt Sources and Configuration	1337
25.3	Registers for Controlling the Interrupts	1344
25.3.1	Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)	1350
25.3.2	Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)	1352
25.3.3	Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)	1354
25.3.4	External interrupt rising edge enable register (EGP0), External interrupt falling edge enable register (EGN0)	1358
25.3.5	Program status word (PSW)	1360
25.3.6	Registers for controlling the port functions multiplexed with the interrupt inputs	1361
25.4	Interrupt Servicing Operations	1362
25.4.1	Maskable interrupt request acknowledgment	1362
25.4.2	Software interrupt request acknowledgment	1365
25.4.3	Multiple interrupt servicing	1365
25.4.4	Interrupt request held pending	1369
26.	Key Interrupt Function	1370
26.1	Functions of the Key Interrupt	1370
26.2	Configuration of the Key Interrupt	1370
26.3	Registers for Controlling the Key Interrupt	1372
26.3.1	Key return control register (KRCTL)	1372
26.3.2	Key return mode register 0 (KRM0)	1373
26.3.3	Key return flag register (KRF)	1374
26.3.4	Registers for controlling the port functions multiplexed with the key interrupt inputs	1374
27.	Standby Function	1375
27.1	Standby Function	1375
27.2	Registers for Controlling the Standby Function	1376
27.2.1	Standby mode release setting register (WKUPMD)	1377
27.2.2	Memory power reduction control register (PSMCR)	1378
27.3	Standby Function Operation	1380
27.3.1	HALT mode	1380
27.3.2	STOP mode	1386
27.3.3	SNOOZE mode	1392

28.	Reset Function	1396
28.1	Timing of Reset Operation	1398
28.2	Registers for Controlling the Reset Function	1402
28.2.1	Reset control flag register (RESF)	1402
28.2.2	Power-on-reset status register (PORSR)	1405
28.2.3	Peripheral reset control register 0 (PRR0)	1406
28.2.4	Peripheral reset control register 1 (PRR1)	1407
28.2.5	Peripheral reset control register 2 (PRR2)	1408
29.	Power-on-reset Circuit (POR)	1409
29.1	Functions of Power-on-reset Circuit	1409
29.2	Configuration of Power-on-reset Circuit	1410
29.3	Operation of Power-on-reset Circuit	1410
30.	Voltage Detector (LVD)	1413
30.1	Functions of Voltage Detector	1413
30.2	Configuration of Voltage Detector	1414
30.3	Registers for Controlling the Voltage Detector	1415
30.3.1	Voltage detection register (LVIM)	1415
30.3.2	LVD detection flag clearing register (LVDFCLR)	1416
30.3.3	Voltage detection level register (LVIS)	1417
30.4	Operation of Voltage Detector	1419
30.4.1	When used as reset mode	1419
30.4.2	When used as interrupt mode	1422
30.5	Points for Caution when the Voltage Detector Is to be Used	1425
31.	Oscillation Stop Detector	1428
31.1	Functions of Oscillation Stop Detector	1428
31.2	Configuration of Oscillation Stop Detector	1429
31.3	Registers for Controlling the Oscillation Stop Detector	1430
31.3.1	Peripheral enable register 2 (PER2)	1430
31.3.2	Peripheral reset control register 2 (PRR2)	1431
31.3.3	Oscillation stop detection control register (OSDC)	1432
31.4	Operation of Oscillation Stop Detector	1433
31.4.1	How the oscillation stop detector operates	1433
31.5	Points for Caution when the Oscillation Stop Detector Is to be Used	1434
32.	Safety Functions	1435
32.1	Overview of Safety Functions	1435
32.2	Registers for Controlling the Safety Functions	1436
32.3	Operation of Safety Functions	1436
32.3.1	Flash memory CRC operation function (high-speed CRC)	1436
32.3.1.1	Flash memory CRC control register (CRC0CTL)	1437
32.3.1.2	Flash memory CRC operation result register (PGCRCL)	1439
32.3.2	CRC operation (general-purpose CRC)	1441
32.3.2.1	CRC input register (CRCIN)	1441
32.3.2.2	CRC data register (CRCD)	1442

32.3.3	Flash memory guard function	1443
32.3.3.1	Code flash memory guard register (GFLASH0)	1443
32.3.3.2	Data flash memory guard register (GFLASH1)	1444
32.3.3.3	Flash security area guard register (GFLASH2)	1445
32.3.4	RAM parity error detection	1446
32.3.4.1	RAM parity error control register (RPECTL)	1446
32.3.5	RAM guard function	1448
32.3.5.1	Invalid memory access detection control register (IAWCTL)	1448
32.3.6	SFR guard function	1449
32.3.6.1	Invalid memory access detection control register (IAWCTL)	1449
32.3.7	Illicit memory access detection	1450
32.3.7.1	Invalid memory access detection control register (IAWCTL)	1451
32.3.8	Guard function of invalid memory access detection control register	1452
32.3.8.1	Guard register of IAWCTL register (GIAWCTL)	1452
32.3.9	Frequency detection	1453
32.3.9.1	Timer input select register 0 (TIS0)	1454
32.3.10	Testing of the A/D converter	1455
32.3.10.1	A/D test register (ADTES)	1457
32.3.10.2	Analog input channel specification register (ADS)	1458
32.3.11	Detection of the digital output signal level of the I/O pins	1460
32.3.11.1	Port mode select register (PMS)	1460
32.3.12	UART loopback	1461
32.3.12.1	UART loopback select register (ULBS)	1462
33.	Security Functions	1464
33.1	True Random Number Generator	1464
33.1.1	Function of the true random number generator	1464
33.1.2	Registers for controlling the true random number generator	1464
33.1.2.1	Random number seed command register 0 (TRNGSCR0)	1465
33.1.2.2	Random number seed data register (TRNGSDR)	1466
33.1.3	Operations of the true random number generator	1467
33.2	Flash Read Protection	1468
33.2.1	Function of flash read protection	1468
33.2.2	Setting of flash read protection	1468
33.2.3	Operation	1470
33.3	Unique ID	1471
33.3.1	Function of a unique ID	1471
33.3.2	ASCII codes representing the product name	1471
34.	SNOOZE Mode Sequencer (SMS)	1473
34.1	Functions of the SNOOZE Mode Sequencer	1473
34.2	Configuration of the SNOOZE Mode Sequencer	1474
34.3	Registers for Controlling the SNOOZE Mode Sequencer	1475
34.3.1	Peripheral enable register 1 (PER1)	1475
34.3.2	Peripheral reset control register 1 (PRR1)	1476
34.3.3	Sequencer instruction registers p (SMSIp) (p = 0 to 31)	1477
34.3.4	Sequencer general-purpose registers n (SMSGn) (n = 0 to 15)	1480
34.3.5	Sequencer control register (SMSC)	1482
34.3.6	Sequencer status register (SMSS)	1484

34.4	Operations of the SNOOZE Mode Sequencer	1485
34.4.1	Internal operations of the SNOOZE mode sequencer	1485
34.4.2	Memory space allocated to the sequencer	1486
34.4.3	Sequencer flags	1488
34.4.4	Procedures for running the SNOOZE mode sequencer	1489
34.4.5	States of the SNOOZE mode sequencer	1491
34.5	Commands for Use in Processing by the Sequencer	1492
34.5.1	8-bit data transfer 1	1493
34.5.2	8-bit data transfer 2	1494
34.5.3	16-bit data transfer 1	1495
34.5.4	16-bit data transfer 2	1496
34.5.5	1-bit data setting	1497
34.5.6	1-bit data clearing	1498
34.5.7	1-bit data transfer	1499
34.5.8	Word addition	1500
34.5.9	Word subtraction	1501
34.5.10	Word comparison	1502
34.5.11	Logical shift right	1503
34.5.12	Branch 1 (SCY = 1)	1504
34.5.13	Branch 2 (SCY = 0)	1505
34.5.14	Branch 3 (SZ = 1)	1506
34.5.15	Branch 4 (SZ = 0)	1507
34.5.16	Wait	1508
34.5.17	Conditional wait 1 (bit = 1)	1510
34.5.18	Conditional wait 2 (bit = 0)	1511
34.5.19	Termination	1512
34.5.20	Interrupt plus termination	1513
34.5.21	DTC activation	1514
34.6	Operation in Standby Modes	1515
35.	Capacitive Sensing Unit (CTS2La)	1516
35.1	Overview	1518
35.1.1	Functional overview of the CTSU	1518
35.1.2	Measurement states	1520
35.2	Registers for Controlling the Capacitive Sensing Unit	1522
35.2.1	Peripheral enable register 1 (PER1)	1523
35.2.2	Peripheral reset control register 1 (PRR1)	1524
35.2.3	CTS control registers AL and AH (CTSUCRAL, CTSUCRAH)	1525
35.2.4	CTS control registers BL and BH (CTSUCRBL, CTSUCRBH)	1533
35.2.5	CTS measurement channel registers L and H (CTSUCMCHL, CTSUCMCHH)	1536
35.2.6	CTS channel enable control registers AL, AH, and BL (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL)	1538
35.2.7	CTS channel transmit/receive control registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL)	1539
35.2.8	CTS status register L (CTSUSRL)	1540
35.2.9	CTS sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)	1543
35.2.10	CTS sensor counter registers (CTSUSC, CTSUUC)	1546
35.2.11	CTS calibration registers L and H (CTSUDBGR0, CTSUDBGR1)	1547

35.2.12	CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)	1551
35.2.13	CTSUTRIM0, CTSUTRIM1)	1553
35.2.14	CTSUTRIM2, CTSUTRIM3)	1554
35.2.15	Registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensor	1555
35.3	Operation	1556
35.3.1	Principles of measurement	1556
35.3.2	Initial setting procedure	1559
35.3.3	Measurement methods	1560
35.3.4	Scan modes	1563
35.3.5	Multi-clock measurement	1563
35.3.6	Multiple electrode connection	1563
35.3.7	Interrupts	1564
35.4	Usage Notes on the Capacitive Sensing Unit	1566
35.4.1	Evaluation of detection by the capacitive sensing unit (CTS2La)	1566
35.4.2	TSCAP pin	1566
35.4.3	Changing the clock and port settings during measurement	1566
35.4.4	Software trigger	1566
35.4.5	External trigger	1567
35.4.6	Reading the CTSU sensor counter registers (CTSUSC and CTSUUC)	1567
35.4.7	Forced stop	1567
35.4.8	Sampling cycle when the spread spectrum is in use	1567
35.4.9	Settings of the transmission pin when the self-capacitance method is to be used	1567
35.4.10	Points to note on noise countermeasures	1568
35.4.11	Memory occupancy	1568
36.	Regulator	1569
36.1	Overview	1569
37.	Option Bytes	1570
37.1	Functions of Option Bytes	1570
37.1.1	User option bytes (000C0H to 000C2H or addresses of the area to be used in swapping)	1571
37.1.2	On-chip debug option byte (000C3H or address of the area to be used in swapping)	1572
37.2	Format of User Option Bytes	1573
37.3	Format of On-chip Debug Option Byte	1578
37.4	Setting of Option Bytes	1579
38.	Flash Memory	1580
38.1	Serial Programming Using Flash Memory Programmer	1582
38.1.1	Programming environment	1583
38.1.2	Communications mode	1583
38.2	Serial Programming Using External Device (that Incorporates UART)	1584
38.2.1	Programming environment	1584
38.2.2	Communications mode	1585
38.3	Handling of Pins on the Board	1586
38.3.1	P40/TOOL0 pin	1586
38.3.2	RESET pin	1587

38.3.3	Port pins	1588
38.3.4	REGC pin	1588
38.3.5	Power supply	1588
38.4	Programming Method	1589
38.4.1	Serial programming procedure	1589
38.4.2	Flash memory programming mode	1590
38.4.3	Selecting communications mode	1591
38.4.4	Communications commands	1592
38.5	Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values)	1594
38.6	Self-programming	1595
38.6.1	Self-programming procedure	1596
38.6.2	Registers for controlling the flash memory	1597
38.6.2.1	Flash address pointer registers H and L (FLAPH, FLAPL)	1598
38.6.2.2	Flash end address pointer registers H and L (FLSEDH, FLSEDL)	1599
38.6.2.3	Flash write buffer registers H and L (FLWH, FLWL)	1601
38.6.2.4	Flash protect command register (PFCMD)	1602
38.6.2.5	Flash status register (PFS)	1603
38.6.2.6	Flash programming mode control register (FLPMC)	1604
38.6.2.7	Flash area selection register (FLARS)	1605
38.6.2.8	Flash memory sequencer initial setting register (FSSET)	1606
38.6.2.9	Flash memory sequencer control register (FSSQ)	1608
38.6.2.10	Flash extra area sequencer control register (FSSE)	1610
38.6.2.11	Flash registers initialization register (FLRST)	1612
38.6.2.12	Flash memory sequencer status registers H and L (FSASTH, FSASTL)	1613
38.6.2.13	Flash security flag monitoring register (FLSEC)	1615
38.6.2.14	Flash FSW monitoring register E (FLFSWE)	1617
38.6.2.15	Flash FSW monitoring register S (FLFSWS)	1618
38.6.2.16	Data flash control register (DFLCTL)	1619
38.6.2.17	Interrupt vector jump enable register (VECTCTRL)	1620
38.6.2.18	Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)	1621
38.6.3	Setting the flash memory control mode	1622
38.6.4	Initializing the registers for use with the flash memory sequencer	1624
38.6.5	Setting the operating frequency of the flash memory sequencer	1624
38.6.6	Rewriting the flash memory	1625
38.6.7	Interrupts in code flash memory programming mode	1636
38.6.8	Example of executing the commands to rewrite the flash memory areas	1638
38.6.9	Notes on self-programming	1641
38.7	Boot Swapping	1642
38.8	Bank Programming	1645
38.8.1	Bank mode switching	1645
38.8.2	Bank programming	1646
38.8.3	Bank swapping	1647
38.8.4	Flash operating mode select register (FLMODE)	1648
38.8.5	Procedures for updating the program	1649
38.9	Flash Shield Window	1651
38.10	Security Settings	1652
38.11	Settings of Boot Cluster Size and Bank Swapping	1655

38.12	Data Flash Memory	1656
38.12.1	Overview of the data flash memory	1656
38.12.2	Procedure for accessing the data flash memory	1657
39.	On-chip Debugging	1658
39.1	Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/L23	1658
39.2	Connection between the External Device that Incorporates UART and RL78/L23	1659
39.3	Security Settings for On-chip Debugging	1660
39.4	Allocation of Memory Spaces to User Resources	1661
39.5	On-chip Debugging in the Bank Programming Mode	1661
40.	BCD Correction Circuit	1664
40.1	BCD Correction Circuit Function	1664
40.2	Register for Controlling the BCD Correction Circuit	1664
40.2.1	BCD correction result register (BCDADJ)	1664
40.3	BCD Correction Circuit Operation	1665
41.	Instruction Set	1667
41.1	Conventions Used in Operation List	1668
41.1.1	Operand identifiers and specification methods	1668
41.1.2	Description of operation column	1669
41.1.3	Description of flag operation column	1670
41.1.4	PREFIX instruction	1670
41.2	Operation List	1671
42.	Electrical Characteristics	1689
42.1	Absolute Maximum Ratings	1690
42.2	Characteristics of the Oscillators	1692
42.2.1	Characteristics of the X1 oscillator	1692
42.2.2	Characteristics of the XT1 oscillator	1692
42.2.3	Characteristics of the On-chip Oscillators	1693
42.3	DC Characteristics	1694
42.3.1	Pin characteristics	1694
42.3.2	Supply current characteristics	1702
42.4	AC Characteristics	1713
42.5	Characteristics of the Peripheral Functions	1719
42.5.1	Serial array unit	1719
42.5.2	Serial interface UARTA	1748
42.5.3	Serial interface IICA	1749
42.6	Analog Characteristics	1752
42.6.1	A/D converter characteristics	1752
42.6.2	Temperature sensor/internal reference voltage characteristics	1757
42.6.3	D/A converter characteristics	1757
42.6.4	Comparator characteristics	1758
42.6.5	POR circuit characteristics	1759
42.6.6	LVD circuit characteristics	1760
42.6.7	Power supply voltage rising slope characteristics	1762
42.6.8	LCD characteristics	1763

42.7	RAM Data Retention Characteristics	1769
42.8	Flash Memory Programming Characteristics	1769
42.9	Dedicated Flash Memory Programmer Communication (UART)	1771
42.10	Timing of Entry to Flash Memory Programming Modes	1771
43.	Package Drawings	1772
43.1	44-pin Products	1772
43.2	48-pin Products	1773
43.3	52-pin Products	1775
43.4	64-pin Products	1776
43.5	80-pin Products	1778
43.6	100-pin Products	1780
APPENDIX A	Revision History	1782
A.1	Major Revisions in This Edition	1782
A.2	Revision History of Preceding Editions	1784

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
- Minimum instruction execution time: Can be changed from high speed (0.03125 μ s @ 32 MHz operation with the high-speed on-chip oscillator clock) to ultra-low speed (30.5 μ s @ 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: 16 to 32 KB

Code flash memory

- Code flash memory: 64 to 128 KB (single bank)
256 KB (128 KB $\times$ 2 banks or a single 256-KB bank)
512 KB (256 KB $\times$ 2 banks or a single 512-KB bank)
- Block size: 2 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debugging
- Self-programming (with boot swapping and flash shield window)
- Bank programming: Enables updating of the program in one bank while the user program is running from the other bank.

Data flash memory

- Data flash memory: 8 KB
- Background 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

- Select from 64 MHz, 48 MHz, 32 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, or 1 MHz
- High accuracy: $\pm 1.0\%$ (VDD = 1.6 to 5.5 V, TA = -40 to +105°C)

Middle-speed on-chip oscillator

- Select from 4 MHz, 2 MHz, or 1 MHz (with adjustability)

Low-speed on-chip oscillator

- 32.768 kHz (typ.) (with adjustability)

Operating ambient temperature

- $T_A = -40$ to $+105^{\circ}\text{C}$ (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/L23 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.

Logic and event link controller (ELCL)

- Event signals can be set up between specified peripheral functions.
- The signals can be generated by the input of multiple event signals to the logic circuit.
- Flip-flop circuits are incorporated to handle setting and resetting functions.

Serial interface

- Simplified SPI (CSINote): 5 to 8 channels
- UART/UART (LIN-bus supported)/UARTA: 5 to 8 channels
- I²C/Simplified I²C: 8 to 10 channels

Timers

- 16-bit timer: 8 channels (with the output of remote control signals)
- 16-bit timer RJ: 1 or 2 channels
- 16-bit timers KB40, KB41, and KB42 (IH): 1 to 3 channels (with PWM output for IH)
- 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
- 8-bit interval timer: 4 to 8 units
- 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)
- External signal sampler: 1 channel
- Oscillation stop detector: 1 channel

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.

LCD controller/driver

- Internal boosting, capacitive division, and external resistor division are switchable as methods of generating the driving voltage.
- Number of segment signal outputs: 19 to 56
- Number of common signal outputs: 4 to 8

A/D converter

- 8-/10-/12-bit resolution
- Analog input: 8 to 13 channels
- Internal reference voltage (1.48 V (typ.)) and temperature sensor

D/A converter

- 8-bit resolution
- Analog output: 3 channels (one of the three interfaces is only for the internal reference voltage for the comparator)
- Output voltage: 0 V to V_{DD}
- Realtime output function

Comparator

- 1 or 2 channels
- Operating modes: Comparator high-speed mode and comparator low-speed mode
- The external reference voltage and the internal reference voltage or D/A converter output are selectable as the reference voltage.

Capacitive sensing unit

- Operating voltage condition: V_{DD} = 1.8 to 5.5 V
- Self-capacitance method: A single pin configures a single key, supporting up to 36 keys
- Mutual capacitance method: Matrix configuration with 8 × 8 pins, supporting up to 64 keys

Input/output port pins

- Number of port pins: 40 to 95
 - N-ch open-drain I/O (withstand voltage of 6 V): 2 to 4,
 - N-ch open-drain I/O (V_{DD} withstand voltage): 24 to 57,
 - Controlled current drive pins: 4 to 8
- 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 V, 2.5 V, or 3 V)

Others

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

Remark The functions mounted depend on the product. For details, see **1.6 Outline of Functions**.

O ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/L23					
			44 pins	48 pins	52 pins	64 pins	80 pins	100 pins
512 KB	8 KB	32 KB	R7F100LFL	R7F100LGL	R7F100LJL	R7F100LLL	R7F100LML	R7F100LPL
256 KB	8 KB	32 KB	R7F100LFJ	R7F100LGJ	R7F100LJJ	R7F100LLJ	R7F100LMJ	R7F100LPJ
128 KB	8 KB	16 KB	R7F100LFG	R7F100LGG	R7F100LJG	R7F100LLG	R7F100LMG	R7F100LPG
64 KB	8 KB	16 KB	R7F100LFE	R7F100LGE	R7F100LJE	R7F100LLE	—	—

1.2 List of Part Numbers

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/L23

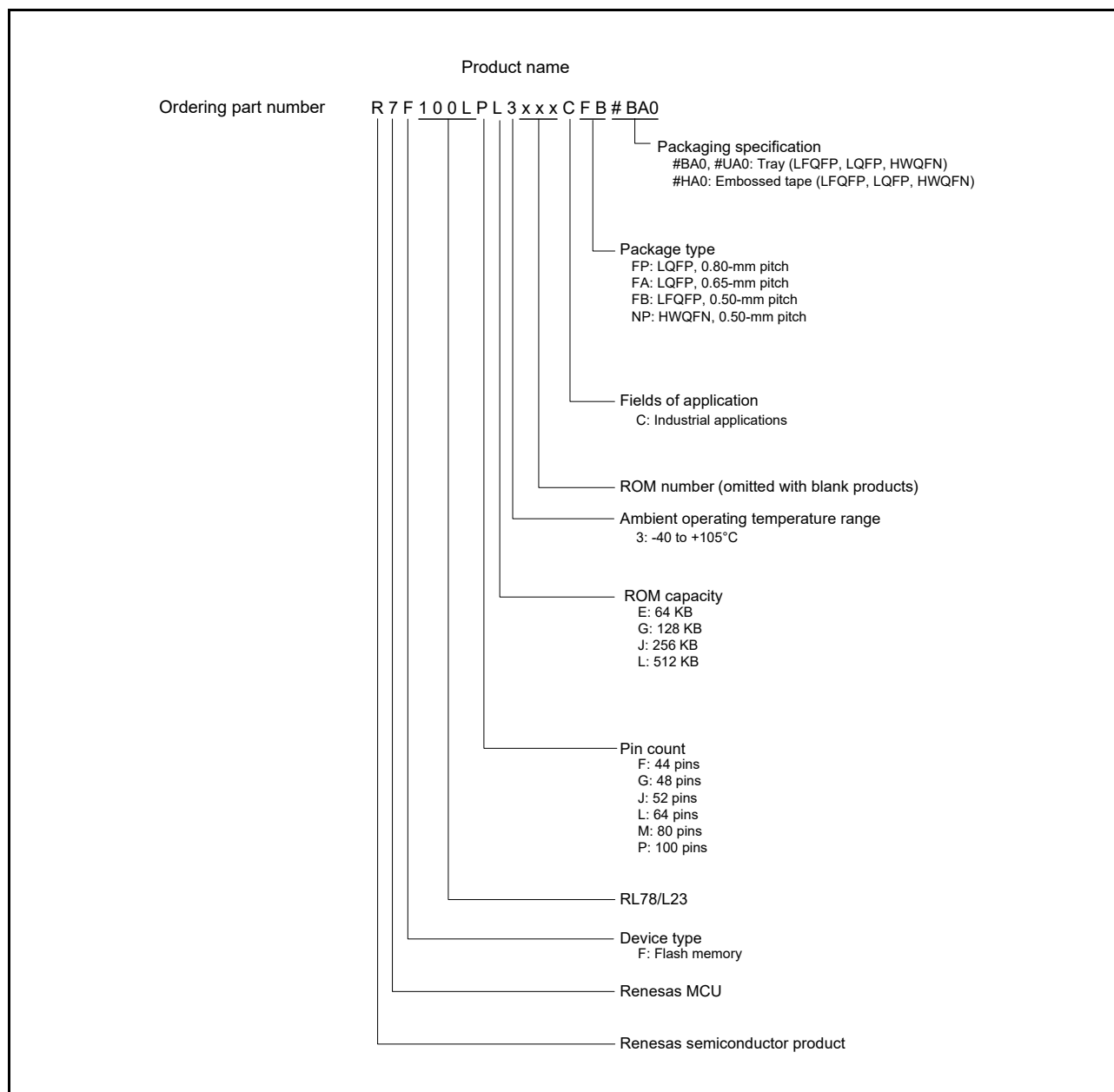


Table 1 - 1 List of Ordering Part Numbers

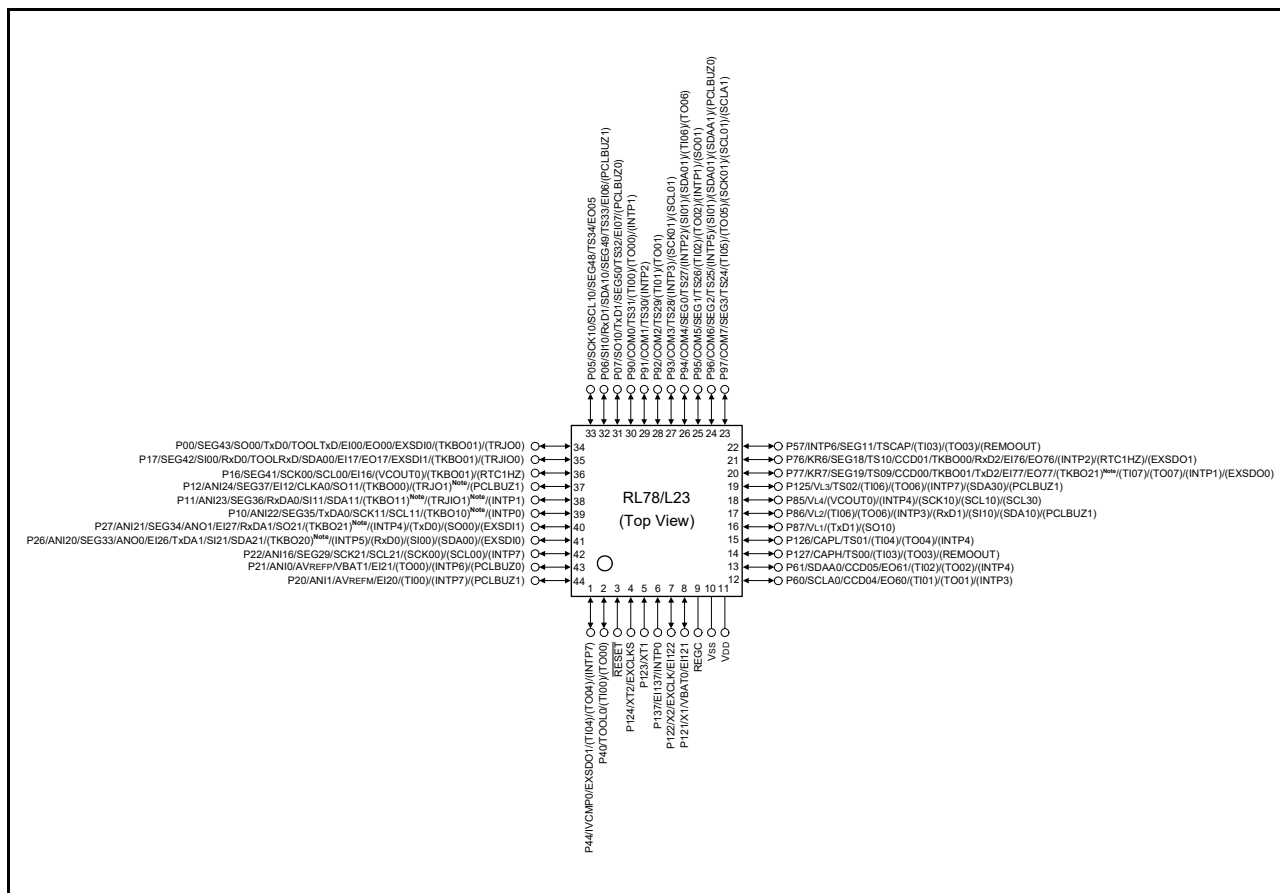
Pin count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
44	44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)	C	R7F100LFL3CFP, R7F100LFJ3CFP, R7F100LFG3CFP, R7F100LFE3CFP	#BA0, #UA0, #HA0	PLQP0044GC-A
48	48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)	C	R7F100LGL3CFB, R7F100LGJ3CFB, R7F100LGG3CFB, R7F100LGE3CFB	#BA0, #UA0, #HA0	PLQP0048KB-B
	48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)	C	R7F100LGL3CNP, R7F100LGJ3CNP, R7F100LGG3CNP, R7F100LGE3CNP	#BA0, #UA0, #HA0	PWQN0048KC-A
52	52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)	C	R7F100LJL3CFA, R7F100LJJ3CFA, R7F100LJG3CFA, R7F100LJE3CFA	#BA0, #UA0, #HA0	PLQP0052JA-A
64	64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)	C	R7F100LLL3CFA, R7F100LLJ3CFA, R7F100LLG3CFA, R7F100LLE3CFA	#BA0, #UA0, #HA0	PLQP0064JB-A
	64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)	C	R7F100LLL3CFB, R7F100LLJ3CFB, R7F100LLG3CFB, R7F100LLE3CFB	#BA0, #UA0, #HA0	PLQP0064KB-C
80	80-pin plastic LQFP (14 × 14 mm, 0.65-mm pitch)	C	R7F100LML3CFA, R7F100LMJ3CFA, R7F100LMG3CFA	#BA0, #UA0, #HA0	PLQP0080JA-B
	80-pin plastic LFQFP (12 × 12 mm, 0.50-mm pitch)	C	R7F100LML3CFB, R7F100LMJ3CFB, R7F100LMG3CFB	#BA0, #UA0, #HA0	PLQP0080KB-B
100	100-pin plastic LFQFP (14 × 14 mm, 0.50-mm pitch)	C	R7F100LPL3CFB, R7F100LPJ3CFB, R7F100LPG3CFB	#BA0, #UA0, #HA0	PLQP0100KB-B
	100-pin plastic LQFP (14 × 20 mm, 0.65-mm pitch)	C	R7F100LPL3CFA, R7F100LPJ3CFA, R7F100LPG3CFA	#BA0, #UA0, #HA0	PLQP0100JC-A

Note For the fields of application, see **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/L23**.

1.3 Pin Configuration (Top View)

1.3.1 44-pin products

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



Note Not present in products with 128 or fewer Kbytes of code flash memory.

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. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 2 Multiplexed Pin Functions of the 44-pin Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	44LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	—	—	—
2	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	—	—	—
3	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
4	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
7	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	Vdd	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	SCLA0	—
13	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	SDAA0	—
14	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
15	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—
16	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—
17	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	—	(RxSD1)/ (SI10)/ (SDA10)	—
18	P85	—	—	—	—	—	VCOU0)	VL4	(INTP4)	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCL30)	—
19	P125	—	—	(PCLBUZ1)	—	—	—	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	—	(SDA30)	—
20	P77	CCD00	EI77/ EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/ (TO07)	—	TKBO01/ (TKBO21) Note	(EXSD00)	—	—	TxD2	—
21	P76	CCD01	EI76/ EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	(EXSD01)	(RTC1HZ)	—	RxD2	—
22	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
23	P97	—	—	—	—	—	—	SEG3/ COM7	—	—	TS24	(TI05)/ (TO05)	—	—	—	—	—	(SCK01)/ (SCL01)	(SCLA1)

Table 1 - 2 Multiplexed Pin Functions of the 44-pin Products (2/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	44LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
24	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/(SDA01)	SDAA1	—
25	P95	—	—	—	—	—	—	SEG1/COM5	(INTP1)	—	TS26	(TI02)/(TO02)	—	—	—	—	(SO01)	—	—
26	P94	—	—	—	—	—	—	SEG0/COM4	(INTP2)	—	TS27	(TI06)/(TO06)	—	—	—	—	(SI01)/(SDA01)	—	—
27	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/(SCL01)	—	—
28	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/(TO01)	—	—	—	—	—	—	—
29	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
30	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/(TO00)	—	—	—	—	—	—	—
31	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/TxD1	—	—
32	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/RxD1/SDA10	—	—
33	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/SCL10	—	—
34	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJ00)	(TKB001)	EXSDI0	—	SO00/TxD0	—	—
35	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJ100)	(TKB001)	EXSDI1	—	SI00/RxD0/SDA00	—	—
36	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKB001)	—	(RTC1Hz)	SCK00/SCL00	—	—
37	P12	—	EI12	(PCLBUZ1)	ANI24	—	—	SEG37	—	—	—	—	(TRJ01) Note	(TKB000)	—	—	SO11	—	CLKA0
38	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJ101) Note	(TKB011) Note	—	—	SI11/SDA11	—	RxDA0
39	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKB010) Note	—	—	SCK11/SCL11	—	TxDA0
40	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKB021) Note	(EXSDI1)	—	(TxD0)/(SO00)/SO21	—	RxDA1
41	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKB020) Note	(EXSDI0)	—	SI21/SDA21/(RxD0)/(SI00)/(SDA00)	—	TxDA1
42	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/(SCL00)/SCK21/SCL21	—	—
43	P21	—	EI21	VBAT1/(PCLBUZ0)	ANI0/AVREFP	—	—	—	(INTP6)	—	—	—	(TO00)	—	—	—	—	—	—

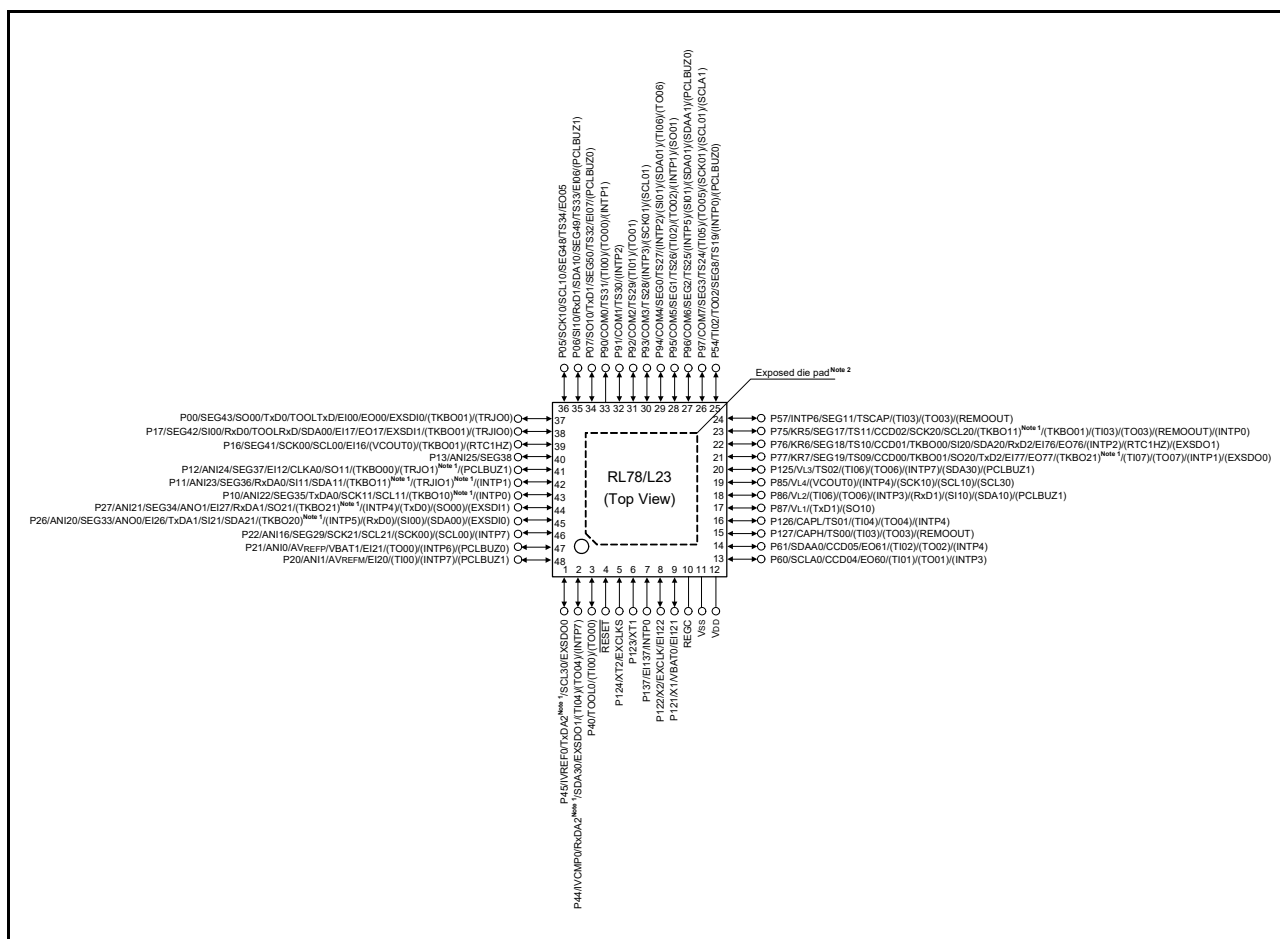
Table 1 - 2 Multiplexed Pin Functions of the 44-pin Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces			
	44LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (C1SU2La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)
44	P20	—		EI20	(PCLBUZ1)	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.3.2 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 1. Not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. 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. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Remark 3. For products in the HWQFN package, solder the exposed die pad to the printed circuit board.

We recommend leaving the destination for mounting of the exposed die pad electrically open-circuit.

Table 1 - 3 Multiplexed Pin Functions of the 48-pin Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	48LFQFP, 48HWQFN	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	SCL30	—	TxD2A2 Note
2	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/(TO04)	—	—	EXSD01	—	SDA30	—	RxD2A2 Note
3	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/(TO00)	—	—	—	—	—	—	—
4	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
5	P124	—	—	XT2/EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
8	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/(TO01)	—	—	—	—	—	SCLA0	—
14	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/(TO02)	—	—	—	—	—	SDAA0	—
15	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/(TO03)/(REMO OUT)	—	—	—	—	—	—	—
16	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/(TO04)	—	—	—	—	—	—	—
17	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/(SO10)	—	—
18	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/(TO06)	—	—	—	—	(RxD1)/(SI10)/(SDA10)	—	—
19	P85	—	—	—	—	—	(VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	(SCK10)/(SCL10)/(SCL30)	—	—
20	P125	—	—	(PCLBUZ1)	—	—	—	VL3	(INTP7)	—	TS02	(TI06)/(TO06)	—	—	—	—	(SDA30)	—	—
21	P77	CCD00	EI77/EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/(TO07)	—	TKBO01/(TKBO21) Note	(EXSD00)	—	SO20/TxD2	—	—
22	P76	CCD01	EI76/EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	(EXSD01)	(RTC1HZ)	SI20/SDA20/RxD2	—	—

Table 1 - 3 Multiplexed Pin Functions of the 48-pin Products (2/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	48LFQFP, 48HWQFN	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
23	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/ (TO03)/ (REMO OUT)	—	TKB011) Note/ TKB001)	—	—	SCK20/ SCL20	—	—
24	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
25	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/ TO02	—	—	—	—	—	—	—
26	P97	—	—	—	—	—	—	SEG3/ COM7	—	—	TS24	(TI05)/ (TO05)	—	—	—	—	(SCK01)/ (SCL01)	(SCLA1)	—
27	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/ COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/ (SDA01)	(SDAA1)	—
28	P95	—	—	—	—	—	—	SEG1/ COM5	(INTP1)	—	TS26	(TI02)/ (TO02)	—	—	—	—	(SO01)	—	—
29	P94	—	—	—	—	—	—	SEG0/ COM4	(INTP2)	—	TS27	(TI06)/ (TO06)	—	—	—	—	(SI01)/ (SDA01)	—	—
30	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/ (SCL01)	—	—
31	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/ (TO01)	—	—	—	—	—	—	—
32	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
33	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/ (TO00)	—	—	—	—	—	—	—
34	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/ TxD1	—	—
35	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/ Rx/D1/ SDA10	—	—
36	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/ SCL10	—	—
37	P00	—	EI00/ EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJ00)	(TKB001)	EXSDI0	—	SO00/ Tx/D0	—	—
38	P17	—	EI17/ EO17	TOOLRx/D	—	—	—	SEG42	—	—	—	—	(TRJ100)	(TKB001)	EXSDI1	—	SI00/ Rx/D0/ SDA00	—	—
39	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKB001)	—	(RTC1HZ)	SCK00/ SCL00	—	—
40	P13	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	—
41	P12	—	EI12	(PCLBUZ1)	ANI24	—	—	SEG37	—	—	—	—	(TRJ01) Note	(TKB000)	—	—	SO11	—	CLKA0
42	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJ101) Note	(TKB011) Note	—	—	SI11/ SDA11	—	RxDA0

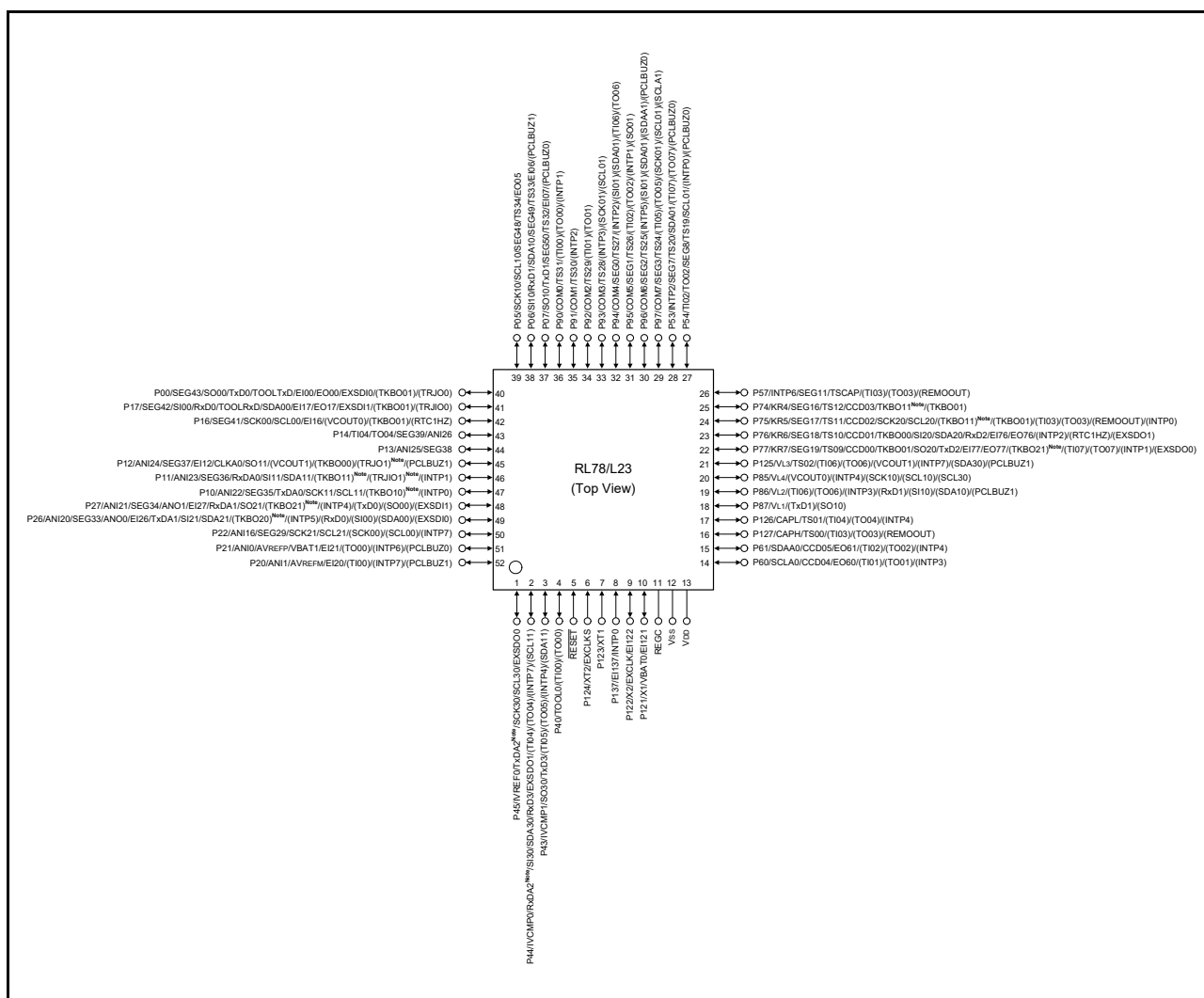
Table 1 - 3 Multiplexed Pin Functions of the 48-pin Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI			Timers					Communications Interfaces		
	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)
43	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKBO10) Note	—	—	SCK11/ SCL11	—	TxDA0
44	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKBO21) Note	(EXSDI1)	—	(TxD0)/ (SO00)/ SO21	—	RxDA1
45	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKBO20) Note	(EXSDI0)	—	SI21/ SDA21/ (RxD0)/ (SI00)/ (SDA00)	—	TxDA1
46	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/ (SCL00)/ SCK21/ SCL21	—	—
47	P21	—	EI21	VBAT1/ (PCLBUZ0)	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
48	P20	—	EI20	(PCLBUZ1)	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.3.3 52-pin products

- 52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

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. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 4 Multiplexed Pin Functions of the 52-pin Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	52LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS2U2La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	SCK30/ SCL30	—	TxDA2 Note
2	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	SI30/ SDA30/ (SCL11)/ RxD3	—	RxDA2 Note
3	P43	—	—	—	—	—	IVCMP1	—	(INTP4)	—	—	(TI05)/ (TO05)	—	—	—	—	SO30/ (SDA11)/ TxD3	—	—
4	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	—	—	—
5	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
6	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
9	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	SCLA0	—
15	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	SDAA0	—
16	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
17	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—
18	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—
19	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	(RxD1)/ (SI10)/ (SDA10)	—	—
20	P85	—	—	—	—	—	(VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCL30)	—	—
21	P125	—	—	(PCLBUZ1)	—	—	(VCOUT1)	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	(SDA30)	—	—
22	P77	CCD00	EI77/ EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/ (TO07)	—	TKBO01/ (TKBO21) Note	EXSD00	—	SO20/ TxD2	—	—

Table 1 - 4 Multiplexed Pin Functions of the 52-pin Products (2/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	52LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
23	P76	CCD01	EI76/EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	EXSD01	(RTC1HZ)	SI20/SDA20/RxD2	—	—
24	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/(TO03)/(REMO OUT)	—	(TKBO11) Note/ (TKBO01)	—	—	SCK20/SCL20	—	—
25	P74	CCD03	—	—	—	—	—	SEG16	—	KR4	TS12	—	—	TKBO11 Note/ (TKBO01)	—	—	—	—	—
26	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/(TO03)/(REMO OUT)	—	—	—	—	—	—	—
27	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/TO02	—	—	—	—	SCL01	—	—
28	P53	—	—	(PCLBUZ0)	—	—	—	SEG7	INTP2	—	TS20	(TI07)/(TO07)	—	—	—	—	SDA01	—	—
29	P97	—	—	—	—	—	—	SEG3/COM7	—	—	TS24	(TI05)/(TO05)	—	—	—	—	(SCK01)/(SCL01)	(SCLA1)	—
30	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/(SDA01)	(SDAA1)	—
31	P95	—	—	—	—	—	—	SEG1/COM5	(INTP1)	—	TS26	(TI02)/(TO02)	—	—	—	—	(SO01)	—	—
32	P94	—	—	—	—	—	—	SEG0/COM4	(INTP2)	—	TS27	(TI06)/(TO06)	—	—	—	—	(SI01)/(SDA01)	—	—
33	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/(SCL01)	—	—
34	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/(TO01)	—	—	—	—	—	—	—
35	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
36	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/(TO00)	—	—	—	—	—	—	—
37	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/TxD1	—	—
38	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/RxD1/SDA10	—	—
39	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/SCL10	—	—
40	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJ00)	(TKBO01)	EXSDI0	—	SO00/TxD0	—	—
41	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJ100)	(TKBO01)	EXSDI1	—	SI00/RxD0/SDA00	—	—

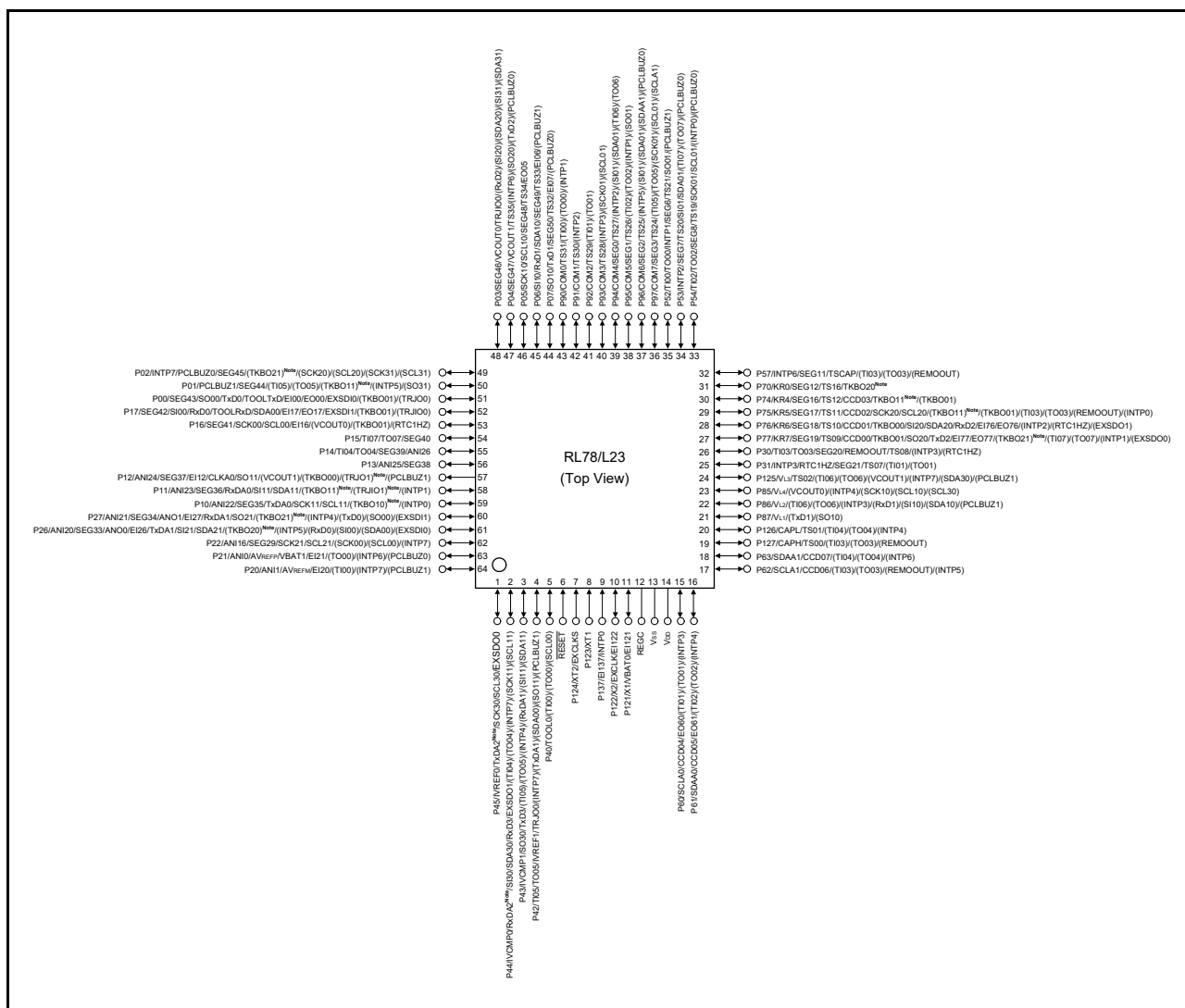
Table 1 - 4 Multiplexed Pin Functions of the 52-pin Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI			Timers					Communications Interfaces		
	52LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
42	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKBO01)	—	(RTC1HZ)	SCK00/ SCL00	—	—
43	P14	—	—	—	ANI26	—	—	SEG39	—	—	—	TI04/ TO04	—	—	—	—	—	—	—
44	P13	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	—
45	P12	—	EI12	(PCLBUZ1)	ANI24	—	(VCOUT1)	SEG37	—	—	—	—	(TRJO1) Note	(TKBO00)	—	—	SO11	—	CLKA0
46	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJO1) Note	(TKBO11) Note	—	—	SI11/ SDA11	—	RxDA0
47	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKBO10) Note	—	—	SCK11/ SCL11	—	TxDA0
48	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKBO21) Note	(EXSD1)	—	(TxD0)/ (SO00)/ SO21	—	RxDA1
49	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKBO20) Note	(EXSD0)	—	SI21/ SDA21/ (RxD0)/ (SI00)/ (SDA00)	—	TxDA1
50	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/ (SCL00)/ SCK21/ SCL21	—	—
51	P21	—	EI21	VBAT1/ (PCLBUZ0)	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
52	P20	—	EI20	(PCLBUZ1)	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.3.4 64-pin products

- 64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)
- 64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

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. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 5 Multiplexed Pin Functions of the 64-pin Products (1/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	64LQFP, 64LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS2U2La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	SCK30/ SCL30	—	TxDA2 Note
2	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	SI30/ SDA30/ (SCL11)/ (SCK11)/ RxD3	—	RxDA2 Note
3	P43	—	—	—	—	—	IVCMP1	—	(INTP4)	—	—	(TI05)/ (TO05)	—	—	—	—	SO30/ (SI11)/ (SDA11)/ TxD3	—	(RxDA1)
4	P42	—	—	(PCLBUZ1)	—	—	IVREF1	—	(INTP7)	—	—	TI05/ TO05	TRJO0	—	—	—	(SDA00)/ (SO11)	—	(TxD1)
5	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	(SCL00)	—	—
6	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
7	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
10	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	SCLA0	—
16	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	SDAA0	—
17	P62	CCD06	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	SCLA1	—
18	P63	CCD07	—	—	—	—	—	—	(INTP6)	—	—	(TI04)/ (TO04)	—	—	—	—	—	SDAA1	—
19	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
20	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—
21	P87	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—
22	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	(RxD1)/ (SI10)/ (SDA10)	—	—

Table 1 - 5 Multiplexed Pin Functions of the 64-pin Products (2/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	64LQFP, 64LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
23	P85	—	—	—	—	—	VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCL30)	—	—
24	P125	—	—	(PCLBUZ1)	—	—	VCOUT1)	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	(SDA30)	—	—
25	P31	—	—	—	—	—	—	SEG21	INTP3	—	TS07	(TI01)/ (TO01)	—	—	—	RTC1HZ	—	—	—
26	P30	—	—	—	—	—	—	SEG20	(INTP3)	—	TS08	TI03/ TO03/ REMOO UT	—	—	—	(RTC1HZ)	—	—	—
27	P77	CCD00	EI77/ EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/ (TO07)	—	TKBO01/ (TKBO21) Note	(EXSD00)	—	SO20/ TxD2	—	—
28	P76	CCD01	EI76/ EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	(EXSD01)	(RTC1HZ)	SI20/ SDA20/ RxD2	—	—
29	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/ (TO03)/ (REMO OUT)	—	TKBO11) Note/ (TKBO01)	—	—	SCK20/ SCL20	—	—
30	P74	CCD03	—	—	—	—	—	SEG16	—	KR4	TS12	—	—	TKBO11 Note/ (TKBO01)	—	—	—	—	—
31	P70	—	—	—	—	—	—	SEG12	—	KR0	TS16	—	—	TKBO20 Note	—	—	—	—	—
32	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
33	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/ TO02	—	—	—	—	SCL01/ SCK01	—	—
34	P53	—	—	(PCLBUZ0)	—	—	—	SEG7	INTP2	—	TS20	(TI07)/ (TO07)	—	—	—	—	SI01/ SDA01	—	—
35	P52	—	—	(PCLBUZ1)	—	—	—	SEG6	INTP1	—	TS21	TI00/ TO00	—	—	—	—	SO01	—	—
36	P97	—	—	—	—	—	—	SEG3/ COM7	—	—	TS24	(TI05)/ (TO05)	—	—	—	—	(SCK01)/ (SCL01)	(SCLA1)	—
37	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/ COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/ (SDA01)	(SDAA1)	—
38	P95	—	—	—	—	—	—	SEG1/ COM5	(INTP1)	—	TS26	(TI02)/ (TO02)	—	—	—	—	(SO01)	—	—
39	P94	—	—	—	—	—	—	SEG0/ COM4	(INTP2)	—	TS27	(TI06)/ (TO06)	—	—	—	—	(SI01)/ (SDA01)	—	—
40	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/ (SCL01)	—	—
41	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/ (TO01)	—	—	—	—	—	—	—

Table 1 - 5 Multiplexed Pin Functions of the 64-pin Products (3/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces			
	64LQFP, 64LFOFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)
42	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—	
43	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/(TO00)	—	—	—	—	—	—	—	
44	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/TxD1	—	—	
45	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/RxD1/SDA10	—	—	
46	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/SCL10	—	—	
47	P04	—	—	(PCLBUZ0)	—	—	VCOUT1	SEG47	(INTP6)	—	TS35	—	—	—	—	—	(TxD2)/(SO20)	—	—	
48	P03	—	—	—	—	—	VCOUT0	SEG46	—	—	—	—	TRJIO0	—	—	—	(SI31)/(SDA31)/(RxD2)/(SI20)/(SDA20)	—	—	
49	P02	—	—	PCLBUZ0	—	—	—	SEG45	INTP7	—	—	—	—	(TKBO21) Note	—	—	(SCK31)/(SCL31)/(SCK20)/(SCL20)	—	—	
50	P01	—	—	PCLBUZ1	—	—	—	SEG44	(INTP5)	—	—	(TI05)/(TO05)	—	(TKBO11) Note	—	—	(SO31)	—	—	
51	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJIO0)	(TKBO01)	EXSDI0	—	SO00/TxD0	—	—	
52	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJIO0)	(TKBO01)	EXSDI1	—	SI00/RxD0/SDA00	—	—	
53	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKBO01)	—	(RTC1Hz)	SCK00/SCL00	—	—	
54	P15	—	—	—	—	—	—	SEG40	—	—	—	TI07/TO07	—	—	—	—	—	—	—	
55	P14	—	—	—	—	ANI26	—	—	SEG39	—	—	—	TI04/TO04	—	—	—	—	—	—	
56	P13	—	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	
57	P12	—	EI12	(PCLBUZ1)	—	ANI24	—	(VCOUT1)	SEG37	—	—	—	—	(TRJIO1) Note	(TKBO00)	—	—	SO11	—	CLKA0
58	P11	—	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJIO1) Note	(TKBO11) Note	—	—	SI11/SDA11	—	RxDA0
59	P10	—	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKBO10) Note	—	—	SCK11/SCL11	—	TxDA0
60	P27	—	EI27	—	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKBO21) Note	(EXSDI1)	—	(TxD0)/(SO00)/SO21	—	RxDA1

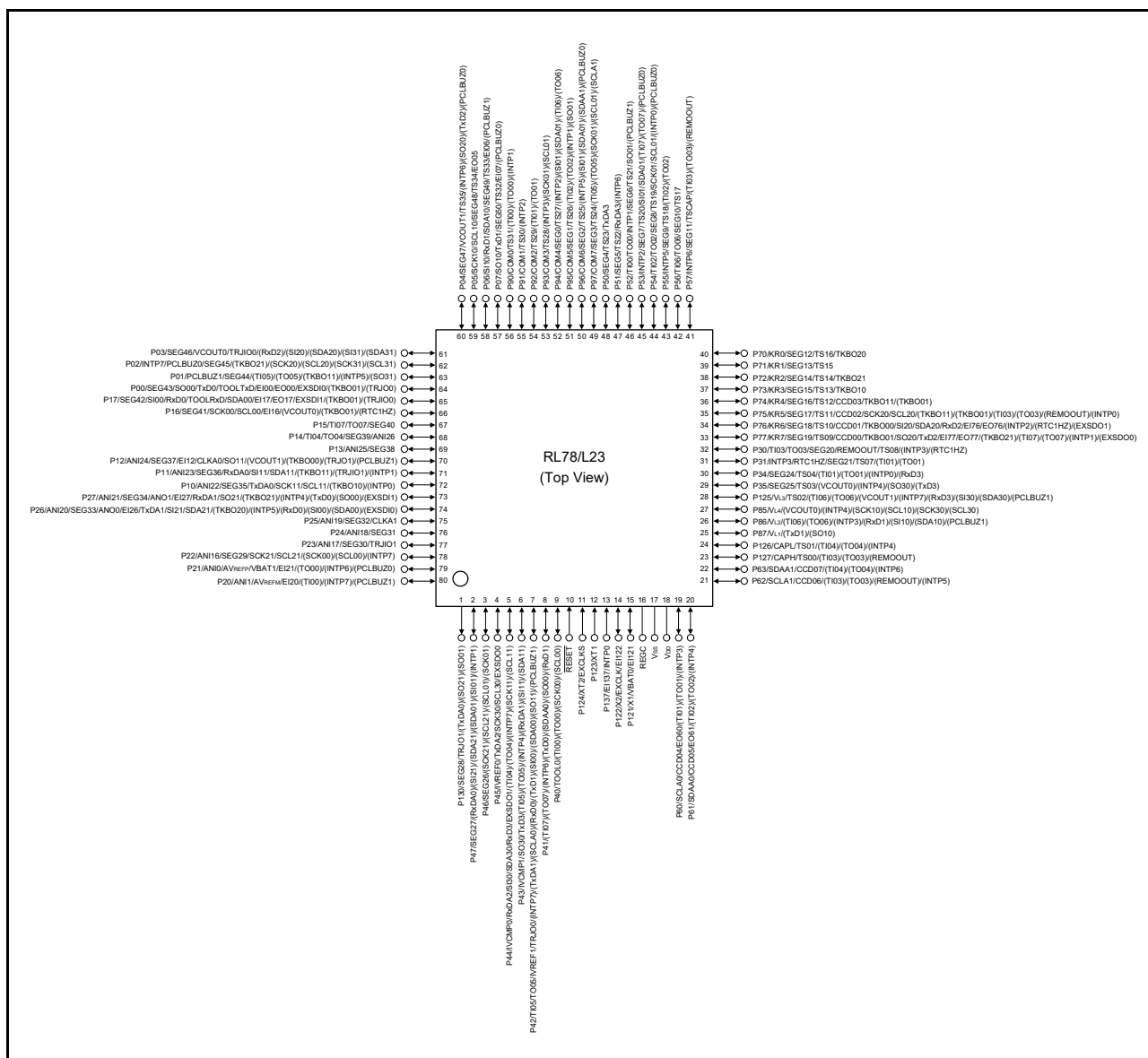
Table 1 - 5 Multiplexed Pin Functions of the 64-pin Products (4/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	64LQFP, 64LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
61	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKBO20) Note	(EXSDI0)	—	SI21/ SDA21/ (RxD0)/ (SI00)/ (SDA00)	—	TxDA1
62	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/ (SCL00)/ SCK21/ SCL21	—	—
63	P21	—	EI21	VBAT1/ (PCLBUZ0)	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
64	P20	—	EI20	(PCLBUZ1)	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.3.5 80-pin products

- 80-pin plastic LQFP (14 × 14 mm, 0.65-mm pitch)
- 80-pin plastic LFQFP (12 × 12 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. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 6 Multiplexed Pin Functions of the 80-pin Products (1/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	80LQFP, 80LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P130	—	—	—	—	—	—	SEG28	—	—	—	—	TRJO1	—	—	—	(SO21)/ (SO01)	—	(TxDA0)
2	P47	—	—	—	—	—	—	SEG27	(INTP1)	—	—	—	—	—	—	—	(SI01)/ (SI21)/ (SDA01)/ (SDA21)	—	(RxDA0)
3	P46	—	—	—	—	—	—	SEG26	—	—	—	—	—	—	—	—	(SCL01)/ (SCK01)/ (SCK21)/ (SCL21)	—	—
4	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	SCK30/ SCL30	—	TxDA2
5	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	SI30/ SDA30/ (SCL11)/ (SCK11)/ RxD3	—	RxDA2
6	P43	—	—	—	—	—	IVCMP1	—	(INTP4)	—	—	(TI05)/ (TO05)	—	—	—	—	SO30/ (SI11)/ (SDA11)/ TxD3	—	(RxDA1)
7	P42	—	—	(PCLBUZ1)	—	—	IVREF1	—	(INTP7)	—	—	TI05/ TO05	TRJO0	—	—	—	(SI00)/ (RxD0)/ (SDA00)/ (SO11)/ (TxD1)	(SCLA0)	(TxDA1)
8	P41	—	—	—	—	—	—	—	(INTP6)	—	—	(TI07)/ (TO07)	—	—	—	—	(SO00)/ (TxD0)/ (RxD1)	(SDAA0)	—
9	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	(SCL00)/ (SCK00)	—	—
10	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
14	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
17	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
18	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
19	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	SCLA0	—
20	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	SDAA0	—
21	P62	CCD06	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	SCLA1	—
22	P63	CCD07	—	—	—	—	—	—	(INTP6)	—	—	(TI04)/ (TO04)	—	—	—	—	—	SDAA1	—

Table 1 - 6 Multiplexed Pin Functions of the 80-pin Products (2/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI's			Timers					Communications Interfaces		
	80LQFP, 80LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS2La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
23	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
24	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—
25	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—
26	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	(RxD1)/ (SI10)/ (SDA10)	—	—
27	P85	—	—	—	—	—	(VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCK30)/ (SCL30)	—	—
28	P125	—	—	(PCLBUZ1)	—	—	(VCOUT1)	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	(SI30)/ (RxD3)/ (SDA30)	—	—
29	P35	—	—	—	—	—	(VCOUT0)	SEG25	(INTP4)	—	TS03	—	—	—	—	—	(SO30)/ (TxD3)	—	—
30	P34	—	—	—	—	—	—	SEG24	(INTP0)	—	TS04	(TI01)/ (TO01)	—	—	—	—	(RxD3)	—	—
31	P31	—	—	—	—	—	—	SEG21	INTP3	—	TS07	(TI01)/ (TO01)	—	—	—	RTC1HZ	—	—	—
32	P30	—	—	—	—	—	—	SEG20	(INTP3)	—	TS08	TI03/ TO03/ REMOO UT	—	—	—	(RTC1HZ)	—	—	—
33	P77	CCD00	EI77/ EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/ (TO07)	—	TKBO01/ (TKBO21)	(EXSD00)	—	SO20/ TxD2	—	—
34	P76	CCD01	EI76/ EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	(EXSD01)	(RTC1HZ)	SI20/ SDA20/ RxD2	—	—
35	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/ (TO03)/ (REMO OUT)	—	(TKBO11)/ (TKBO01)	—	—	SCK20/ SCL20	—	—
36	P74	CCD03	—	—	—	—	—	SEG16	—	KR4	TS12	—	—	TKBO11/ (TKBO01)	—	—	—	—	—
37	P73	—	—	—	—	—	—	SEG15	—	KR3	TS13	—	—	TKBO10	—	—	—	—	—
38	P72	—	—	—	—	—	—	SEG14	—	KR2	TS14	—	—	TKBO21	—	—	—	—	—
39	P71	—	—	—	—	—	—	SEG13	—	KR1	TS15	—	—	—	—	—	—	—	—
40	P70	—	—	—	—	—	—	SEG12	—	KR0	TS16	—	—	TKBO20	—	—	—	—	—
41	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—

Table 1 - 6 Multiplexed Pin Functions of the 80-pin Products (3/4)

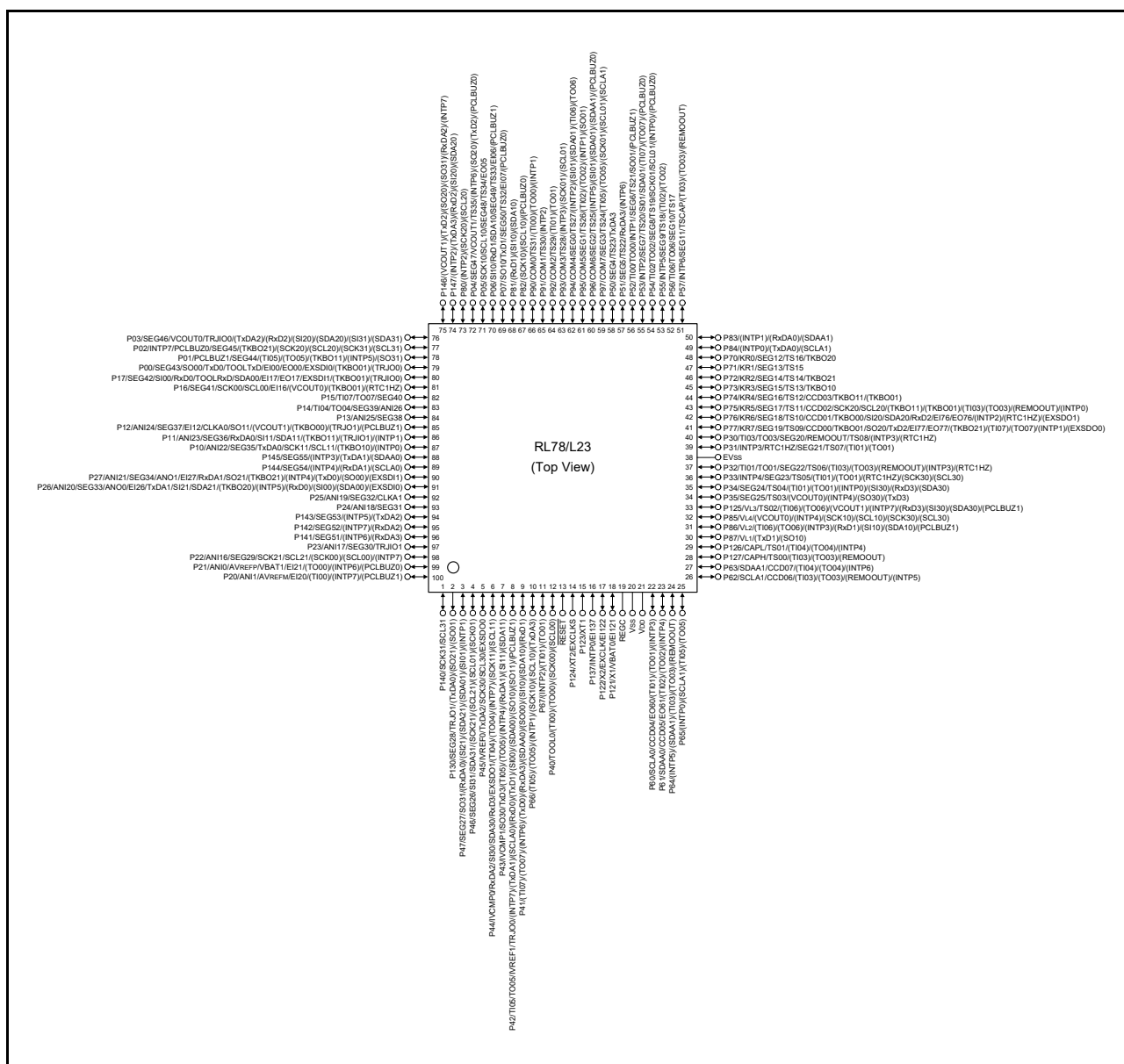
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI's			Timers					Communications Interfaces		
	80LQFP, 80LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
42	P56	—	—	—	—	—	—	SEG10	—	—	TS17	TI06/TO06	—	—	—	—	—	—	—
43	P55	—	—	—	—	—	—	SEG9	INTP5	—	TS18	(TI02)/ (TO02)	—	—	—	—	—	—	—
44	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/TO02	—	—	—	—	SCL01/ SCK01	—	—
45	P53	—	—	(PCLBUZ0)	—	—	—	SEG7	INTP2	—	TS20	(TI07)/ (TO07)	—	—	—	—	SI01/ SDA01	—	—
46	P52	—	—	(PCLBUZ1)	—	—	—	SEG6	INTP1	—	TS21	TI00/TO00	—	—	—	—	SO01	—	—
47	P51	—	—	—	—	—	—	SEG5	(INTP6)	—	TS22	—	—	—	—	—	—	—	RxDA3
48	P50	—	—	—	—	—	—	SEG4	—	—	TS23	—	—	—	—	—	—	—	TxDA3
49	P97	—	—	—	—	—	—	SEG3/ COM7	—	—	TS24	(TI05)/ (TO05)	—	—	—	—	(SCK01)/ (SCL01)	(SCLA1)	—
50	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/ COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/ (SDA01)	(SDAA1)	—
51	P95	—	—	—	—	—	—	SEG1/ COM5	(INTP1)	—	TS26	(TI02)/ (TO02)	—	—	—	—	(SO01)	—	—
52	P94	—	—	—	—	—	—	SEG0/ COM4	(INTP2)	—	TS27	(TI06)/ (TO06)	—	—	—	—	(SI01)/ (SDA01)	—	—
53	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/ (SCL01)	—	—
54	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/ (TO01)	—	—	—	—	—	—	—
55	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
56	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/ (TO00)	—	—	—	—	—	—	—
57	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/ TxD1	—	—
58	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/ RxD1/ SDA10	—	—
59	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/ SCL10	—	—
60	P04	—	—	(PCLBUZ0)	—	—	VCOUT1	SEG47	(INTP6)	—	TS35	—	—	—	—	—	(TxD2)/ (SO20)	—	—
61	P03	—	—	—	—	—	VCOUT0	SEG46	—	—	—	—	TRJIO0	—	—	—	(SI31)/ (SDA31)/ (RxD2)/ (SI20)/ (SDA20)	—	—
62	P02	—	—	PCLBUZ0	—	—	—	SEG45	INTP7	—	—	—	—	(TKBO21)	—	—	(SCK31)/ (SCL31)/ (SCK20)/ (SCL20)	—	—
63	P01	—	—	PCLBUZ1	—	—	—	SEG44	(INTP5)	—	—	(TI05)/ (TO05)	—	(TKBO11)	—	—	(SO31)	—	—

Table 1 - 6 Multiplexed Pin Functions of the 80-pin Products (4/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	80LQFP, 80LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
64	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJ00)	(TKB001)	EXSDI0	—	SO00/TxD0	—	—
65	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJ100)	(TKB001)	EXSDI1	—	SI00/RxD0/SDA00	—	—
66	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKB001)	—	(RTC1HZ)	SCK00/SCL00	—	—
67	P15	—	—	—	—	—	—	SEG40	—	—	—	TI07/TO07	—	—	—	—	—	—	—
68	P14	—	—	—	ANI26	—	—	SEG39	—	—	—	TI04/TO04	—	—	—	—	—	—	—
69	P13	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	—
70	P12	—	EI12	(PCLBUZ1)	ANI24	—	(VCOUT1)	SEG37	—	—	—	—	(TRJ01)	(TKB000)	—	—	SO11	—	CLKA0
71	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJ101)	(TKB011)	—	—	SI11/SDA11	—	RxDA0
72	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKB010)	—	—	SCK11/SCL11	—	TxDA0
73	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKB021)	(EXSDI1)	—	(TxD0)/(SO00)/SO21	—	RxDA1
74	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKB020)	(EXSDI0)	—	SI21/SDA21/(RxD0)/(SI00)/(SDA00)	—	TxDA1
75	P25	—	—	—	ANI19	—	—	SEG32	—	—	—	—	—	—	—	—	—	—	CLKA1
76	P24	—	—	—	ANI18	—	—	SEG31	—	—	—	—	—	—	—	—	—	—	—
77	P23	—	—	—	ANI17	—	—	SEG30	—	—	—	—	TRJ101	—	—	—	—	—	—
78	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/(SCL00)/SCK21/SCL21	—	—
79	P21	—	EI21	VBAT1/(PCLBUZ0)	ANI0/AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
80	P20	—	EI20	(PCLBUZ1)	ANI1/AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

1.3.6 100-pin products

- 100-pin plastic LFQFP (14 × 14 mm, 0.50-mm pitch)



Caution 1. Connect the EVSS pin to the same ground as the VSS pin.

Caution 2. 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. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to connect the VSS and EVSS pins to separate ground lines.

Remark 3. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 7 Multiplexed Pin Functions of the 100-pin Products (1/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	100LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P140	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCK31/ SCL31	—	—
2	P130	—	—	—	—	—	—	SEG28	—	—	—	—	TRJ01	—	—	—	(SO21)/ (SO01)	—	(TxDA0)
3	P47	—	—	—	—	—	—	SEG27	(INTP1)	—	—	—	—	—	—	—	SO31/ (SI01)/ (SI21)/ (SDA01)/ (SDA21)	—	(RxDA0)
4	P46	—	—	—	—	—	—	SEG26	—	—	—	—	—	—	—	—	SI31/ SDA31/ (SCL01)/ (SCK01)/ (SCK21)/ (SCL21)	—	—
5	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	SCK30/ SCL30	—	TxDA2
6	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	SI30/ SDA30/ (SCL11)/ (SCK11)/ RxD3	—	RxDA2
7	P43	—	—	—	—	—	IVCMP1	—	(INTP4)	—	—	(TI05)/ (TO05)	—	—	—	—	SO30/ (SI11)/ (SDA11)/ TxD3	—	(RxDA1)
8	P42	—	—	(PCLBUZ1)	—	—	IVREF1	—	(INTP7)	—	—	TI05/ TO05	TRJ00	—	—	—	(SI00)/ (RxD0)/ (SDA00)/ (SO11)/ (TxD1)/ (SO10)	(SCLA0)	(TxDA1)
9	P41	—	—	—	—	—	—	—	(INTP6)	—	—	(TI07)/ (TO07)	—	—	—	—	(SI10)/ (SO00)/ (TxD0)/ (SDA10)/ (RxD1)	(SDAA0)	(RxDA3)
10	P66	—	—	—	—	—	—	—	(INTP1)	—	—	(TI05)/ (TO05)	—	—	—	—	(SCK10)/ (SCL10)	—	(TxDA3)
11	P67	—	—	—	—	—	—	—	(INTP2)	—	—	(TI01)/ (TO01)	—	—	—	—	—	—	—
12	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	(SCL00)/ (SCK00)	—	—
13	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—
17	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
18	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
19	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
20	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
21	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 7 Multiplexed Pin Functions of the 100-pin Products (2/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces			
	100LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)
22	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	SCLA0	—	
23	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	SDAA0	—	
24	P64	—	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	(SDAA1)	—	
25	P65	—	—	—	—	—	—	—	(INTP0)	—	—	(TI05)/ (TO05)	—	—	—	—	—	(SCLA1)	—	
26	P62	CCD06	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	SCLA1	—	
27	P63	CCD07	—	—	—	—	—	—	(INTP6)	—	—	(TI04)/ (TO04)	—	—	—	—	—	SDAA1	—	
28	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—	
29	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—	
30	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—	
31	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	—	(RxD1)/ (SI10)/ (SDA10)	—	—
32	P85	—	—	—	—	—	(VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCK30)/ (SCL30)	—	—
33	P125	—	—	(PCLBUZ1)	—	—	(VCOUT1)	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	—	(SI30)/ (RxD3)/ (SDA30)	—	—
34	P35	—	—	—	—	—	(VCOUT0)	SEG25	(INTP4)	—	TS03	—	—	—	—	—	—	(SO30)/ (TxD3)	—	—
35	P34	—	—	—	—	—	—	SEG24	(INTP0)	—	TS04	(TI01)/ (TO01)	—	—	—	—	—	(SI30)/ (RxD3)/ (SDA30)	—	—
36	P33	—	—	—	—	—	—	SEG23	INTP4	—	TS05	(TI01)/ (TO01)	—	—	—	(RTC1HZ)	(SCK30)/ (SCL30)	—	—	
37	P32	—	—	—	—	—	—	SEG22	(INTP3)	—	TS06	TI01/ TO01/ (TI03)/ (TO03)/ (REMO OUT)	—	—	—	(RTC1HZ)	—	—	—	

Table 1 - 7 Multiplexed Pin Functions of the 100-pin Products (3/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
38	—	—	—	—	EVss	—	—	—	—	—	—	—	—	—	—	—	—	—	—
39	P31	—	—	—	—	—	—	SEG21	INTP3	—	TS07	(TI01)/ (TO01)	—	—	—	RTC1HZ	—	—	—
40	P30	—	—	—	—	—	—	SEG20	(INTP3)	—	TS08	TI03/ TO03/ REMOO UT	—	—	—	(RTC1HZ)	—	—	—
41	P77	CCD00	EI77/ EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/ (TO07)	—	TKBO01/ (TKBO21)	(EXSDO0)	—	SO20/ TxD2	—	—
42	P76	CCD01	EI76/ EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	(EXSDO1)	(RTC1HZ)	SI20/ SDA20/ RxD2	—	—
43	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/ (TO03)/ (REMO OUT)	—	TKBO11/ (TKBO01)	—	—	SCK20/ SCL20	—	—
44	P74	CCD03	—	—	—	—	—	SEG16	—	KR4	TS12	—	—	TKBO11/ (TKBO01)	—	—	—	—	—
45	P73	—	—	—	—	—	—	SEG15	—	KR3	TS13	—	—	TKBO10	—	—	—	—	—
46	P72	—	—	—	—	—	—	SEG14	—	KR2	TS14	—	—	TKBO21	—	—	—	—	—
47	P71	—	—	—	—	—	—	SEG13	—	KR1	TS15	—	—	—	—	—	—	—	—
48	P70	—	—	—	—	—	—	SEG12	—	KR0	TS16	—	—	TKBO20	—	—	—	—	—
49	P84	—	—	—	—	—	—	—	(INTP0)	—	—	—	—	—	—	—	—	(SCLA1)	(TxDA0)
50	P83	—	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—	(SDAA1)	(RxDA0)
51	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
52	P56	—	—	—	—	—	—	SEG10	—	—	TS17	TI06/ TO06	—	—	—	—	—	—	—
53	P55	—	—	—	—	—	—	SEG9	INTP5	—	TS18	(TI02)/ (TO02)	—	—	—	—	—	—	—
54	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/ TO02	—	—	—	—	SCL01/ SCK01	—	—
55	P53	—	—	(PCLBUZ0)	—	—	—	SEG7	INTP2	—	TS20	(TI07)/ (TO07)	—	—	—	—	SI01/ SDA01	—	—
56	P52	—	—	(PCLBUZ1)	—	—	—	SEG6	INTP1	—	TS21	TI00/ TO00	—	—	—	—	SO01	—	—
57	P51	—	—	—	—	—	—	SEG5	(INTP6)	—	TS22	—	—	—	—	—	—	—	RxDA3
58	P50	—	—	—	—	—	—	SEG4	—	—	TS23	—	—	—	—	—	—	—	TxDA3
59	P97	—	—	—	—	—	—	SEG3/ COM7	—	—	TS24	(TI05)/ (TO05)	—	—	—	—	(SCK01)/ (SCL01)	(SCLA1)	—
60	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/ COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/ (SDA01)	(SDAA1)	—

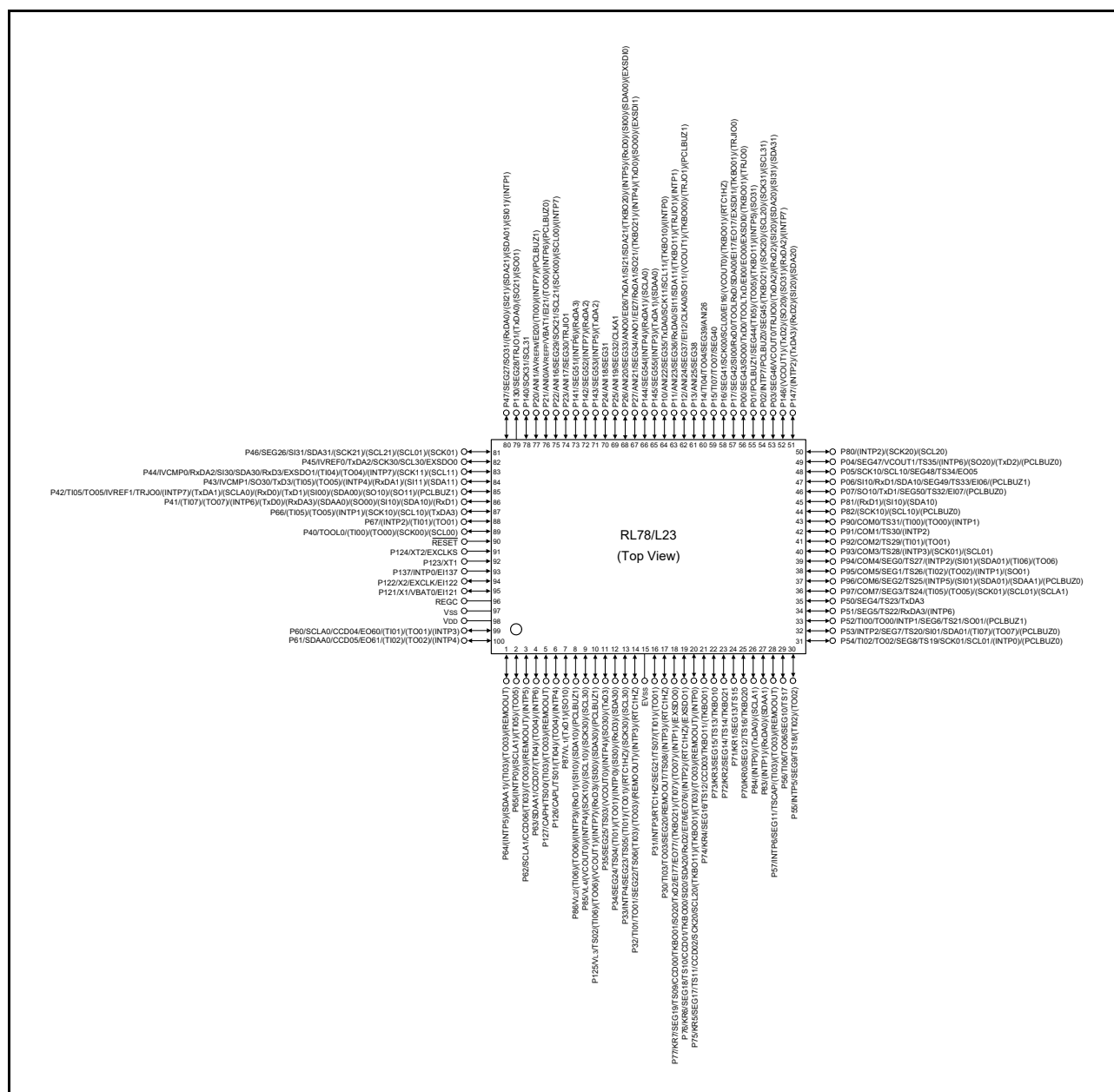
Table 1 - 7 Multiplexed Pin Functions of the 100-pin Products (4/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
61	P95	—	—	—	—	—	—	SEG1/COM5	(INTP1)	—	TS26	(TI02)/(TO02)	—	—	—	—	(SO01)	—	—
62	P94	—	—	—	—	—	—	SEG0/COM4	(INTP2)	—	TS27	(TI06)/(TO06)	—	—	—	—	(SI01)/(SDA01)	—	—
63	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/(SCL01)	—	—
64	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/(TO01)	—	—	—	—	—	—	—
65	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
66	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/(TO00)	—	—	—	—	—	—	—
67	P82	—	—	(PCLBUZ0)	—	—	—	—	—	—	—	—	—	—	—	—	(SCK10)/(SCL10)	—	—
68	P81	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(RxD1)/(SI10)/(SDA10)	—	—
69	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/TxD1	—	—
70	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/RxD1/SDA10	—	—
71	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/SCL10	—	—
72	P04	—	—	(PCLBUZ0)	—	—	VCOUT1	SEG47	(INTP6)	—	TS35	—	—	—	—	—	(TxD2)/(SO20)	—	—
73	P80	—	—	—	—	—	—	—	(INTP2)	—	—	—	—	—	—	—	(SCK20)/(SCL20)	—	—
74	P147	—	—	—	—	—	—	—	(INTP2)	—	—	—	—	—	—	—	(RxD2)/(SI20)/(SDA20)	—	(TxDA3)
75	P146	—	—	—	—	—	VCOUT1	—	(INTP7)	—	—	—	—	—	—	—	(TxD2)/(SO31)/(SO20)	—	(RxDA2)
76	P03	—	—	—	—	—	VCOUT0	SEG46	—	—	—	—	TRJIO0	—	—	—	(SI31)/(SDA31)/(RxD2)/(SI20)/(SDA20)	—	(TxDA2)
77	P02	—	—	PCLBUZ0	—	—	—	SEG45	INTP7	—	—	—	—	(TKBO21)	—	—	(SCK31)/(SCL31)/(SCK20)/(SCL20)	—	—
78	P01	—	—	PCLBUZ1	—	—	—	SEG44	(INTP5)	—	—	(TI05)/(TO05)	—	(TKBO11)	—	—	(SO31)	—	—
79	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJ00)	(TKBO01)	EXSDI0	—	SO00/TxD0	—	—
80	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJIO0)	(TKBO01)	EXSDI1	—	SI00/RxD0/SDA00	—	—

Table 1 - 7 Multiplexed Pin Functions of the 100-pin Products (5/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	100LFQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
81	P16	—	EI16	—	—	—	VCOUT0	SEG41	—	—	—	—	—	(TKBO01)	—	(RTC1HZ)	SCK00/ SCL00	—	—
82	P15	—	—	—	—	—	—	SEG40	—	—	—	TI07/ TO07	—	—	—	—	—	—	—
83	P14	—	—	—	ANI26	—	—	SEG39	—	—	—	TI04/ TO04	—	—	—	—	—	—	—
84	P13	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	—
85	P12	—	EI12	(PCLBUZ1)	ANI24	—	VCOUT1	SEG37	—	—	—	—	(TRJO1)	(TKBO00)	—	—	SO11	—	CLKA0
86	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJIO1)	(TKBO11)	—	—	SI11/ SDA11	—	RxDA0
87	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKBO10)	—	—	SCK11/ SCL11	—	TxDA0
88	P145	—	—	—	—	—	—	SEG55	(INTP3)	—	—	—	—	—	—	—	—	(SDAA0)	(TxDA1)
89	P144	—	—	—	—	—	—	SEG54	(INTP4)	—	—	—	—	—	—	—	—	(SCLA0)	(RxDA1)
90	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKBO21)	(EXSDI1)	—	(TxD0)/ (SO00)/ SO21	—	RxDA1
91	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKBO20)	(EXSDI0)	—	SI21/ SDA21/ (RxDO)/ (SI00)/ (SDA00)	—	TxDA1
92	P25	—	—	—	ANI19	—	—	SEG32	—	—	—	—	—	—	—	—	—	—	CLKA1
93	P24	—	—	—	ANI18	—	—	SEG31	—	—	—	—	—	—	—	—	—	—	—
94	P143	—	—	—	—	—	—	SEG53	(INTP5)	—	—	—	—	—	—	—	—	—	(TxDA2)
95	P142	—	—	—	—	—	—	SEG52	(INTP7)	—	—	—	—	—	—	—	—	—	(RxDA2)
96	P141	—	—	—	—	—	—	SEG51	(INTP6)	—	—	—	—	—	—	—	—	—	(RxDA3)
97	P23	—	—	—	ANI17	—	—	SEG30	—	—	—	—	TRJIO1	—	—	—	—	—	—
98	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/ (SCL00)/ SCK21/ SCL21	—	—
99	P21	—	EI21	VBAT1/ (PCLBUZ0)	ANI0/ AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
100	P20	—	EI20	(PCLBUZ1)	ANI1/ AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—

- 100-pin plastic LQFP (14 × 20 mm, 0.65-mm pitch)



Caution 1. Connect the EVss pin to the same ground as the Vss pin.

Caution 2. 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. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to connect the Vss and EVss pins to separate ground lines.

Remark 3. Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

Table 1 - 8 Multiplexed Pin Functions of the 100-pin Products (1/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS2U2La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
1	P64	—	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	(SDAA1)	—
2	P65	—	—	—	—	—	—	—	(INTP0)	—	—	(TI05)/ (TO05)	—	—	—	—	—	(SCLA1)	—
3	P62	CCD06	—	—	—	—	—	—	(INTP5)	—	—	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	SCLA1	—
4	P63	CCD07	—	—	—	—	—	—	(INTP6)	—	—	(TI04)/ (TO04)	—	—	—	—	—	SDAA1	—
5	P127	—	—	—	—	—	—	CAPH	—	—	TS00	(TI03)/ (TO03)/ (REMO OUT)	—	—	—	—	—	—	—
6	P126	—	—	—	—	—	—	CAPL	(INTP4)	—	TS01	(TI04)/ (TO04)	—	—	—	—	—	—	—
7	P87	—	—	—	—	—	—	VL1	—	—	—	—	—	—	—	—	(TxD1)/ (SO10)	—	—
8	P86	—	—	(PCLBUZ1)	—	—	—	VL2	(INTP3)	—	—	(TI06)/ (TO06)	—	—	—	—	—	(RxD1)/ (SI10)/ (SDA10)	—
9	P85	—	—	—	—	—	(VCOUT0)	VL4	(INTP4)	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)/ (SCK30)/ (SCL30)	—
10	P125	—	—	(PCLBUZ1)	—	—	(VCOUT1)	VL3	(INTP7)	—	TS02	(TI06)/ (TO06)	—	—	—	—	—	(SI30)/ (RxD3)/ (SDA30)	—
11	P35	—	—	—	—	—	(VCOUT0)	SEG25	(INTP4)	—	TS03	—	—	—	—	—	—	(SO30)/ (TxD3)	—
12	P34	—	—	—	—	—	—	SEG24	(INTP0)	—	TS04	(TI01)/ (TO01)	—	—	—	—	—	(SI30)/ (RxD3)/ (SDA30)	—
13	P33	—	—	—	—	—	—	SEG23	INTP4	—	TS05	(TI01)/ (TO01)	—	—	—	(RTC1HZ)	(SCK30)/ (SCL30)	—	—
14	P32	—	—	—	—	—	—	SEG22	(INTP3)	—	TS06	TI01/ TO01/ (TI03)/ (TO03)/ (REMO OUT)	—	—	—	(RTC1HZ)	—	—	—
15	—	—	—	EVss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	P31	—	—	—	—	—	—	SEG21	INTP3	—	TS07	(TI01)/ (TO01)	—	—	—	RTC1HZ	—	—	—

Table 1 - 8 Multiplexed Pin Functions of the 100-pin Products (2/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
17	P30	—	—	—	—	—	—	SEG20	(INTP3)	—	TS08	TI03/TO03/REMOOUT	—	—	—	(RTC1HZ)	—	—	—
18	P77	CCD00	EI77/EO77	—	—	—	—	SEG19	(INTP1)	KR7	TS09	(TI07)/(TO07)	—	TKBO01/(TKBO21)	EXSD00	—	SO20/TxD2	—	—
19	P76	CCD01	EI76/EO76	—	—	—	—	SEG18	(INTP2)	KR6	TS10	—	—	TKBO00	EXSD01	(RTC1HZ)	SI20/SDA20/RxD2	—	—
20	P75	CCD02	—	—	—	—	—	SEG17	(INTP0)	KR5	TS11	(TI03)/(TO03)/(REMOOUT)	—	TKBO11/(TKBO01)	—	—	SCK20/SCL20	—	—
21	P74	CCD03	—	—	—	—	—	SEG16	—	KR4	TS12	—	—	TKBO11/(TKBO01)	—	—	—	—	—
22	P73	—	—	—	—	—	—	SEG15	—	KR3	TS13	—	—	TKBO10	—	—	—	—	—
23	P72	—	—	—	—	—	—	SEG14	—	KR2	TS14	—	—	TKBO21	—	—	—	—	—
24	P71	—	—	—	—	—	—	SEG13	—	KR1	TS15	—	—	—	—	—	—	—	—
25	P70	—	—	—	—	—	—	SEG12	—	KR0	TS16	—	—	TKBO20	—	—	—	—	—
26	P84	—	—	—	—	—	—	—	(INTP0)	—	—	—	—	—	—	—	—	(SCLA1)	(TxDA0)
27	P83	—	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—	(SDAA1)	(RxD A0)
28	P57	—	—	—	—	—	—	SEG11	INTP6	—	TSCAP	(TI03)/(TO03)/(REMOOUT)	—	—	—	—	—	—	—
29	P56	—	—	—	—	—	—	SEG10	—	—	TS17	TI06/TO06	—	—	—	—	—	—	—
30	P55	—	—	—	—	—	—	SEG9	INTP5	—	TS18	(TI02)/(TO02)	—	—	—	—	—	—	—
31	P54	—	—	(PCLBUZ0)	—	—	—	SEG8	(INTP0)	—	TS19	TI02/TO02	—	—	—	—	SCL01/SCK01	—	—
32	P53	—	—	(PCLBUZ0)	—	—	—	SEG7	INTP2	—	TS20	(TI07)/(TO07)	—	—	—	—	SI01/SDA01	—	—
33	P52	—	—	(PCLBUZ1)	—	—	—	SEG6	INTP1	—	TS21	TI00/TO00	—	—	—	—	SO01	—	—
34	P51	—	—	—	—	—	—	SEG5	(INTP6)	—	TS22	—	—	—	—	—	—	—	RxD A3
35	P50	—	—	—	—	—	—	SEG4	—	—	TS23	—	—	—	—	—	—	—	TxD A3
36	P97	—	—	—	—	—	—	SEG3/COM7	—	—	TS24	(TI05)/(TO05)	—	—	—	—	(SCK01)/(SCL01)	(SCLA1)	—
37	P96	—	—	(PCLBUZ0)	—	—	—	SEG2/COM6	(INTP5)	—	TS25	—	—	—	—	—	(SI01)/(SDA01)	(SDAA1)	—
38	P95	—	—	—	—	—	—	SEG1/COM5	(INTP1)	—	TS26	(TI02)/(TO02)	—	—	—	—	(SO01)	—	—

Table 1 - 8 Multiplexed Pin Functions of the 100-pin Products (3/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI's			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02LA)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
39	P94	—	—	—	—	—	—	SEG0/COM4	(INTP2)	—	TS27	(TI06)/(TO06)	—	—	—	—	(SI01)/(SDA01)	—	—
40	P93	—	—	—	—	—	—	COM3	(INTP3)	—	TS28	—	—	—	—	—	(SCK01)/(SCL01)	—	—
41	P92	—	—	—	—	—	—	COM2	—	—	TS29	(TI01)/(TO01)	—	—	—	—	—	—	—
42	P91	—	—	—	—	—	—	COM1	(INTP2)	—	TS30	—	—	—	—	—	—	—	—
43	P90	—	—	—	—	—	—	COM0	(INTP1)	—	TS31	(TI00)/(TO00)	—	—	—	—	—	—	—
44	P82	—	—	(PCLBUZ0)	—	—	—	—	—	—	—	—	—	—	—	—	(SCK10)/(SCL10)	—	—
45	P81	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	(RxD1)/(SI10)/(SDA10)	—	—
46	P07	—	EI07	(PCLBUZ0)	—	—	—	SEG50	—	—	TS32	—	—	—	—	—	SO10/TxD1	—	—
47	P06	—	EI06	(PCLBUZ1)	—	—	—	SEG49	—	—	TS33	—	—	—	—	—	SI10/RxD1/SDA10	—	—
48	P05	—	EO05	—	—	—	—	SEG48	—	—	TS34	—	—	—	—	—	SCK10/SCL10	—	—
49	P04	—	—	(PCLBUZ0)	—	—	VCOUT1	SEG47	(INTP6)	—	TS35	—	—	—	—	—	(TxD2)/(SO20)	—	—
50	P80	—	—	—	—	—	—	—	(INTP2)	—	—	—	—	—	—	—	(SCK20)/(SCL20)	—	—
51	P147	—	—	—	—	—	—	—	(INTP2)	—	—	—	—	—	—	—	(RxD2)/(SI20)/(SDA20)	—	(TxDA3)
52	P146	—	—	—	—	—	(VCOUT1)	—	(INTP7)	—	—	—	—	—	—	—	(TxD2)/(SO31)/(SO20)	—	(RxDA2)
53	P03	—	—	—	—	—	VCOUT0	SEG46	—	—	—	—	TRJIO0	—	—	—	(SI31)/(SDA31)/(RxD2)/(SI20)/(SDA20)	—	(TxDA2)
54	P02	—	—	PCLBUZ0	—	—	—	SEG45	INTP7	—	—	—	—	(TKBO21)	—	—	(SCK31)/(SCL31)/(SCK20)/(SCL20)	—	—
55	P01	—	—	PCLBUZ1	—	—	—	SEG44	(INTP5)	—	—	(TI05)/(TO05)	—	(TKBO11)	—	—	(SO31)	—	—
56	P00	—	EI00/EO00	TOOLTxD	—	—	—	SEG43	—	—	—	—	(TRJIO0)	(TKBO01)	EXSDI0	—	SO00/TxD0	—	—
57	P17	—	EI17/EO17	TOOLRxD	—	—	—	SEG42	—	—	—	—	(TRJIO0)	(TKBO01)	EXSDI1	—	SI00/RxD0/SDA00	—	—
58	P16	—	EI16	—	—	—	(VCOUT0)	SEG41	—	—	—	—	—	(TKBO01)	—	(RTC1HZ)	SCK00/SCL00	—	—

Table 1 - 8 Multiplexed Pin Functions of the 100-pin Products (4/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMI			Timers					Communications Interfaces		
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)
59	P15	—	—	—	—	—	—	SEG40	—	—	—	TI07/TO07	—	—	—	—	—	—	—
60	P14	—	—	—	ANI26	—	—	SEG39	—	—	—	TI04/TO04	—	—	—	—	—	—	—
61	P13	—	—	—	ANI25	—	—	SEG38	—	—	—	—	—	—	—	—	—	—	—
62	P12	—	EI12	(PCLBUZ1)	ANI24	—	VCOUT1	SEG37	—	—	—	—	(TRJO1)	(TKBO00)	—	—	SO11	—	CLKA0
63	P11	—	—	—	ANI23	—	—	SEG36	(INTP1)	—	—	—	(TRJIO1)	(TKBO11)	—	—	SI11/SDA11	—	RxDA0
64	P10	—	—	—	ANI22	—	—	SEG35	(INTP0)	—	—	—	—	(TKBO10)	—	—	SCK11/SCL11	—	TxDA0
65	P145	—	—	—	—	—	—	SEG55	(INTP3)	—	—	—	—	—	—	—	—	(SDAA0)	(TxDA1)
66	P144	—	—	—	—	—	—	SEG54	(INTP4)	—	—	—	—	—	—	—	—	(SCLA0)	(RxDA1)
67	P27	—	EI27	—	ANI21	ANO1	—	SEG34	(INTP4)	—	—	—	—	(TKBO21)	(EXSDI1)	—	(TxD0)/(SO00)/SO21	—	RxDA1
68	P26	—	EI26	—	ANI20	ANO0	—	SEG33	(INTP5)	—	—	—	—	(TKBO20)	(EXSDI0)	—	SI21/SDA21/(RxD0)/(SI00)/(SDA00)	—	TxDA1
69	P25	—	—	—	ANI19	—	—	SEG32	—	—	—	—	—	—	—	—	—	—	CLKA1
70	P24	—	—	—	ANI18	—	—	SEG31	—	—	—	—	—	—	—	—	—	—	—
71	P143	—	—	—	—	—	—	SEG53	(INTP5)	—	—	—	—	—	—	—	—	—	(TxDA2)
72	P142	—	—	—	—	—	—	SEG52	(INTP7)	—	—	—	—	—	—	—	—	—	(RxDA2)
73	P141	—	—	—	—	—	—	SEG51	(INTP6)	—	—	—	—	—	—	—	—	—	(RxDA3)
74	P23	—	—	—	ANI17	—	—	SEG30	—	—	—	—	TRJIO1	—	—	—	—	—	—
75	P22	—	—	—	ANI16	—	—	SEG29	(INTP7)	—	—	—	—	—	—	—	(SCK00)/(SCL00)/SCK21/SCL21	—	—
76	P21	—	EI21	VBAT1/(PCLBUZ0)	ANI0/AVREFP	—	—	—	(INTP6)	—	—	(TO00)	—	—	—	—	—	—	—
77	P20	—	EI20	(PCLBUZ1)	ANI1/AVREFM	—	—	—	(INTP7)	—	—	(TI00)	—	—	—	—	—	—	—
78	P140	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCK31/SCL31	—	—
79	P130	—	—	—	—	—	—	SEG28	—	—	—	—	TRJO1	—	—	—	(SO21)/(SO01)	—	(TxDA0)
80	P47	—	—	—	—	—	—	SEG27	(INTP1)	—	—	—	—	—	—	—	SO31/(SI01)/(SI21)/(SDA01)/(SDA21)	—	(RxDA0)

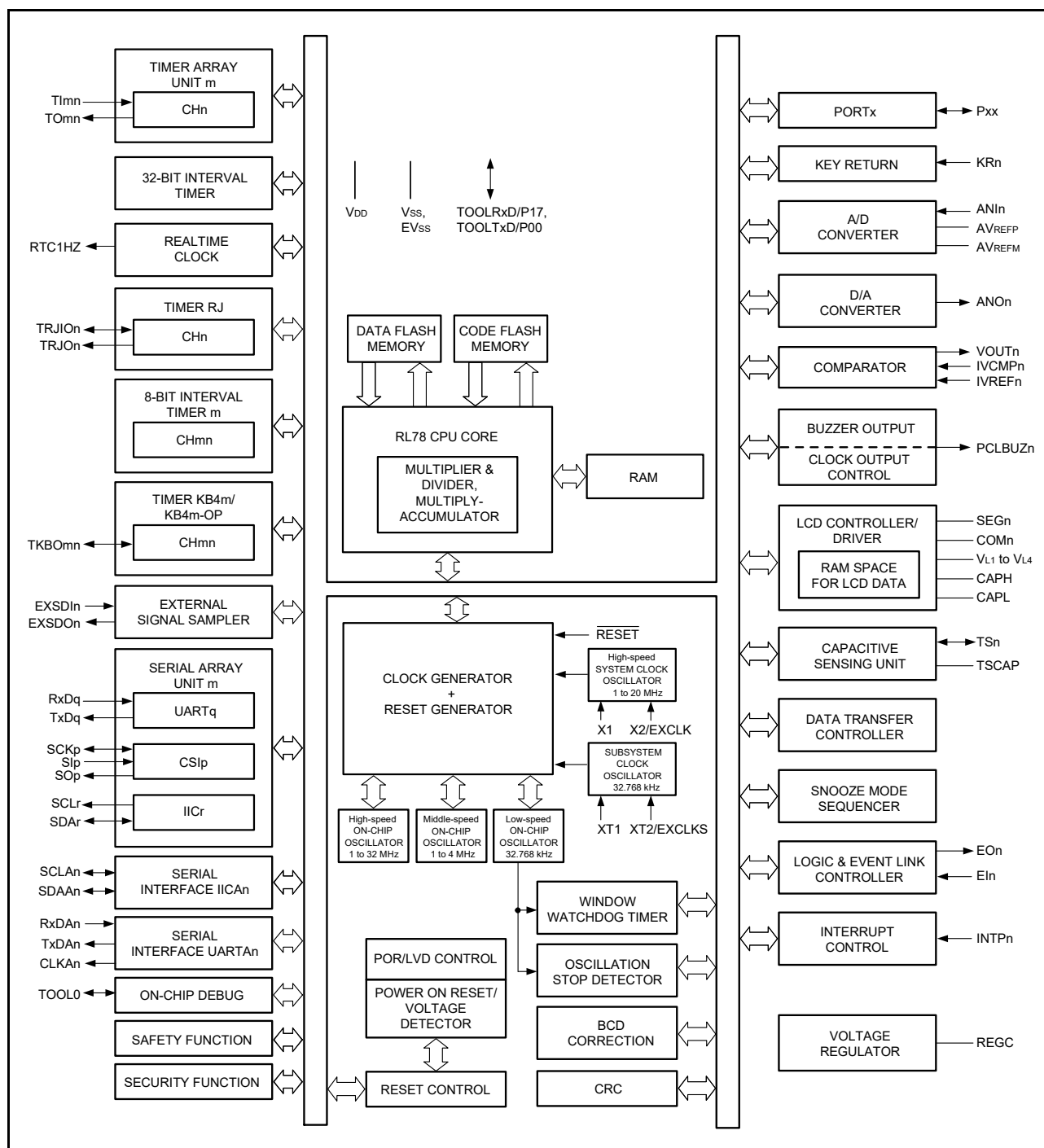
Table 1 - 8 Multiplexed Pin Functions of the 100-pin Products (5/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits				HMIs			Timers					Communications Interfaces			
	100LQFP	Digital port	Controlled current drive port		ELCL input/output port	A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	LCD controller/driver (LCD)	Interrupt	Key interrupt	Capacitive sensing unit (CTS02La)	Timer array unit (TAU)	Timer RJ (TRJ)	16-bit timers KB40, KB41, and KB42	External signal sampler	Realtime clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)
81	P46	—	—	—	—	—	—	SEG26	—	—	—	—	—	—	—	—	—	SI31/ SDA31/ (SCL01)/ (SCK01)/ (SCK21)/ (SCL21)	—	—
82	P45	—	—	—	—	—	IVREF0	—	—	—	—	—	—	—	EXSD00	—	—	SCK30/ SCL30	—	TxDA2
83	P44	—	—	—	—	—	IVCMP0	—	(INTP7)	—	—	(TI04)/ (TO04)	—	—	EXSD01	—	—	SI30/ SDA30/ (SCL11)/ (SCK11)/ RxD3	—	RxDA2
84	P43	—	—	—	—	—	IVCMP1	—	(INTP4)	—	—	(TI05)/ (TO05)	—	—	—	—	—	SO30/ (SI11)/ (SDA11)/ TxD3	—	(RxDA1)
85	P42	—	—	(PCLBUZ1)	—	—	IVREF1	—	(INTP7)	—	—	TI05/ TO05	TRJ00	—	—	—	—	(SI00)/ (RxD0)/ (SDA00)/ (SO11)/ (TxD1)/ (SO10)	(SCLA0)	(TxDA1)
86	P41	—	—	—	—	—	—	—	(INTP6)	—	—	(TI07)/ (TO07)	—	—	—	—	—	(SI10)/ (SO00)/ (TxD0)/ (SDA10)/ (RxD1)	(SDAA0)	(RxDA3)
87	P66	—	—	—	—	—	—	—	(INTP1)	—	—	(TI05)/ (TO05)	—	—	—	—	—	(SCK10)/ (SCL10)	—	(TxDA3)
88	P67	—	—	—	—	—	—	—	(INTP2)	—	—	(TI01)/ (TO01)	—	—	—	—	—	—	—	—
89	P40	—	—	TOOL0	—	—	—	—	—	—	—	(TI00)/ (TO00)	—	—	—	—	—	(SCL00)/ (SCK00)	—	—
90	—	—	—	RESET	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
91	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
92	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
93	P137	—	EI137	—	—	—	—	—	INTP0	—	—	—	—	—	—	—	—	—	—	—
94	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
95	P121	—	EI121	X1/VBAT0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
96	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
97	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
98	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
99	P60	CCD04	EO60	—	—	—	—	—	(INTP3)	—	—	(TI01)/ (TO01)	—	—	—	—	—	—	SCLA0	—
100	P61	CCD05	EO61	—	—	—	—	—	(INTP4)	—	—	(TI02)/ (TO02)	—	—	—	—	—	—	SDAA0	—

1.4 Pin Identification

ANI0, ANI1,		PCLBUZ0, PCLBUZ1:	Programmable clock output/buzzer output
ANI16 to ANI26:	Analog input	REGC:	Regulator capacitance
ANO0, ANO1:	Analog output	RESET:	Reset
AVREFM:	Analog reference voltage minus	RTC1HZ:	Realtime clock correction clock (1 Hz) output
AVREFP:	Analog reference voltage plus	RxD0 to RxD3, RxDA0,	
CCD00 to CCD07:	Controlled current drive output	RxDA1, RxDA2, RxDA3:	Receive data
CLKA0, CLKA1:	Asynchronous serial clock output	SCLA0, SCLA1,	
CAPH, CAPL:	Capacitor for LCD	SCK00, SCK01, SCK10,	
COM0 to COM7:	LCD common output	SCK11, SCK20, SCK21,	
EI00, EI06, EI07, EI12,		SCK30, SCK31:	Serial clock input/output
EI16, EI17, EI20, EI21,		SCL00, SCL01, SCL10,	
EI26, EI27, EI76, EI77,		SCL11, SCL20, SCL21,	
EI121, EI122, EI137:	Logic & event link controller input	SCL30, SCL31:	Serial clock output
EO00, EO05, EO17,		SDAA0, SDAA1, SDA00,	
EO60, EO61, EO76,		SDA01, SDA10, SDA11,	
EO77:	Logic & event link controller output	SDA20, SDA21, SDA30,	
EVss:	Ground for port	SDA31:	Serial data input/output
EXCLK:	External clock input (main system clock)	SEG0 to SEG55:	LCD segment output
EXCLKS:	External clock input (subsystem clock)	SI00, SI01, SI10, SI11,	
EXSDI0, EXSDI1:	External signal sampler input	SI20, SI21, SI30, SI31:	Serial data input
EXSDO0, EXSDO1:	External signal sampler clock output	SO00, SO01, SO10,	
INTP0 to INTP7:	Interrupt request from peripheral modules	SO11, SO20, SO21,	
IVCMP0, IVCMP1:	Comparator input	SO30, SO31:	Serial data output
IVREF0, IVREF1:	Comparator reference input	TSCAP:	Touch sensor capacitance
KR0 to KR7:	Key return	TI00 to TI07:	Timer input
P00 to P07:	Port 0	TO00 to TO07:	Timer output
P10 to P17:	Port 1	TKBO00, TKBO01, TKBO10,	
P20 to P27:	Port 2	TKBO11, TKBO20, TKBO21:	Timer KB4 output
P30 to P35:	Port 3	TOOL0:	Data input/output for tool
P40 to P47:	Port 4	TOOLRxD, TOOLTxD:	Data input/output for external device
P50 to P57:	Port 5	TRJIO0, TRJIO1:	Timer input/output
P60 to P67:	Port 6	TS00 to TS35:	Capacitive sensor
P70 to P77:	Port 7	TxD0 to TxD3, TxDA0,	
P80 to P87:	Port 8	TxDA1, TxDA2, TxDA3:	Transmit data
P90 to P97:	Port 9	VBAT0, VBAT1:	Battery backup power supply
P121 to P127:	Port 12	VCOUT0, VCOUT1:	Comparator output
P130, P137:	Port 13	VDD:	Power supply
P140 to P147:	Port 14	VL1 to VL4:	LCD power supply
		Vss:	Ground
		X1, X2:	Crystal oscillator (main system clock)
		XT1, XT2:	Crystal oscillator (subsystem clock)
		REMOOUT:	Remote control output

1.5 Block Diagram



Remark m: Unit number, n: Channel number, p: 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 the peripheral I/O redirection registers (PIORx) are set to 00H.

(1/4)

Item		44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
		R7F100LFx	R7F100LGx	R7F100LJx	R7F100LLx	R7F100LMx	R7F100LPx
Code flash memory		512 KB (256 KB × 2 banks or a single 512-KB bank) 256 KB (128 KB × 2 banks or a single 256-KB bank) 128 KB (single bank) 64 KB (single bank)				512 KB (256 KB × 2 banks or a single 512-KB bank) 256 KB (128 KB × 2 banks or a single 256-KB bank) 128 KB (single bank)	
Data flash memory		8 KB	8 KB	8 KB	8 KB	8 KB	8 KB
RAM		16 to 32 KB	16 to 32 KB	16 to 32 KB	16 to 32 KB	16 to 32 KB	16 to 32 KB
Address space		1 MB					
CPU/peripheral hardware clock frequency (fCLK)	Main system clock	HS (high-speed main) mode: 1 to 32 MHz (VDD = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz ^{Note 1} (VDD = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (VDD = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz ^{Note 1} (VDD = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz ^{Note 2} (VDD = 1.6 to 5.5 V)					
	Subsystem clock	SUB mode: 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 = 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.					
I/O port	Total number of pins	40	44	48	60	76	95
	CMOS I/O	35 (N-ch open-drain I/O [VDD withstand voltage]: 24)	39 (N-ch open-drain I/O [VDD withstand voltage]: 27)	43 (N-ch open-drain I/O [VDD withstand voltage]: 29)	53 (N-ch open-drain I/O [VDD withstand voltage]: 35)	68 (N-ch open-drain I/O [VDD withstand voltage]: 42)	87 (N-ch open-drain I/O [VDD withstand voltage]: 57)
	CMOS input	3	3	3	3	3	3
	CMOS output	—	—	—	—	1	1
	N-ch open-drain I/O (withstand voltage: 6 V)	2	2	2	4	4	4
	Controlled current drive port	4	5	6	8	8	8

(2/4)

Item		44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
		R7F100LFx	R7F100LGx	R7F100LJx	R7F100LLx	R7F100LMx	R7F100LPx
Timers	16-bit timer	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					
	8-bit interval timer	Products with 256 or 512 Kbytes of flash memory: 8 units Products with 64 or 128 Kbytes of flash memory: 4 units				8 units	
	16-bit timer RJ	Products with 256 or 512 Kbytes of flash memory: 2 channels Products with 64 or 128 Kbytes of flash memory: 1 channel				2 channels	
	16-bit timer KB4	Products with 256 or 512 Kbytes of flash memory: 3 channels Products with 64 or 128 Kbytes of flash memory: 1 channel				3 channels	
	External signal sampler	1 channel					
	Oscillation stop detector	1 channel					
	Timer output	6 channels (PWM outputs: 5 ^{Note 3}), 8 channels (PWM outputs: 7 ^{Note 3}) ^{Note 4}			8 channels (PWM outputs: 7 ^{Note 3})		
RTC output	1 channel						
Clock output/buzzer output		2 channels					
		• 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 (f _{MAIN})) • 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 (f _{SP}))					
8-/10-/12-bit resolution A/D converter		8 channels	9 channels	10 channels	10 channels	13 channels	13 channels
D/A converter		3 channels (one of the three interfaces is only for the internal reference voltage for the comparator)					
Comparator		1 channel			2 channels		

(3/4)

Item		44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
		R7F100LFx	R7F100LGx	R7F100LJx	R7F100LLx	R7F100LMx	R7F100LPx
Serial interfaces		[44-pin products] • Simplified SPI (CSI): 2 channels ^{Note 4} /simplified I ² C: 2 channels ^{Note 4} /UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 1 channel/simplified I ² C: 1 channel/UART (UART supporting LIN-bus): 1 channel • Simplified I ² C: 1 channel ^{Note 4}					
		[48-pin products] • Simplified SPI (CSI): 2 channels ^{Note 4} /simplified I ² C: 2 channels ^{Note 4} /UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART (UART supporting LIN-bus): 1 channel • Simplified I ² C: 1 channel					
		[52-pin products] • Simplified SPI (CSI): 2 channels ^{Note 4} /simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 1 channel/simplified I ² C: 1 channel/UART: 1 channel					
		[64- and 80-pin products] • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 2 channels ^{Note 4} /simplified I ² C: 2 channels ^{Note 4} /UART: 1 channel					
		[100-pin products] • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART (UART supporting LIN-bus): 1 channel • Simplified SPI (CSI): 2 channels/simplified I ² C: 2 channels/UART: 1 channel					
	UARTA	2 channels	Products with 256 or 512 Kbytes of flash memory: 3 channels Products with 64 or 128 Kbytes of flash memory: 2 channels			4 channels	
	I ² C bus	2 channels					
Data transfer controller (DTC)		48 sources (44- and 48-pin products with 64 to 128 Kbytes of flash memory) 49 sources (52- and 64-pin products with 64 to 128 Kbytes of flash memory) 51 sources (44-pin products with 256 to 512 Kbytes of flash memory) 53 sources (48-pin products with 256 to 512 Kbytes of flash memory) 54 sources (52- and 64-pin products with 256 to 512 Kbytes of flash memory) 56 sources (80- and 100-pin products with 128 to 512 Kbytes of flash memory)					
Logic and event link controller (ELCL)		1					
SNOOZE mode sequencer (SMS)		1					
Capacitive sensing unit		16	18	20	25	34	36
LCD controller/driver		Internal boosting, capacitive division, and external resistor division are switchable as methods of generating the driving voltage.					
	Segment signal outputs × common signal outputs	19 seg × 4 com 17 seg × 6 com 15 seg × 8 com	22 seg × 4 com 20 seg × 6 com 18 seg × 8 com	25 seg × 4 com 23 seg × 6 com 21 seg × 8 com	34 seg × 4 com 32 seg × 6 com 30 seg × 8 com	49 seg × 4 com 47 seg × 6 com 45 seg × 8 com	56 seg × 4 com 54 seg × 6 com 52 seg × 8 com
Vectored interrupt sources	Internal	47	49	50	50	52	52
	External	9	9	9	9	9	9
Key interrupt		2	3	4	5	8	8

(4/4)

Item		44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
		R7F100LFx	R7F100LGx	R7F100LJx	R7F100LLx	R7F100LMx	R7F100LPx
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 5} • Internal reset by RAM parity error • Internal reset by illegal-memory access					
Power-on-reset circuit		Detection voltage • 1.50 V (typ.)					
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 V to 3.96 V (6 stages) • Falling edge: 1.65 V to 3.88 V (6 stages)					
	LVD1	Detection voltage • Rising edge: 1.67 V to 4.16 V (18 stages) • Falling edge: 1.63 V to 4.08 V (18 stages)					
On-chip debugging		Available (tracing supported)					
Power supply voltage		VDD = 1.6 to 5.5 V					
Operating ambient temperature		TA = -40 to +105°C (3C: Industrial applications)					

Note 1. Overwrite the flash memory during operation at 2 MHz or a lower frequency.

Note 2. When the flash memory is to be overwritten, switch to high-speed main (HS) mode or low-speed main (LS) mode.

Note 3. 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 4. This applies when the corresponding sets of bits in the peripheral I/O redirection registers (PIORx) have suitable non-0 values.

Note 5. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

Section 2 Pin Functions

2.1 Functions of Port Pins

The power supply for the I/O buffers of pins is a single power supply in all products.

Power Supply	Corresponding Pins
VDD	All pins

2.1.1 44-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/ EI00/EO00/EXSDI0/(TKBO01)/ (TRJO0)	Port 0. 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 P05 and P06 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00 and P05 to P07 as outputs. P00 and P05 to P07 can be set to LCD output ^{Note 2} .
P05	8-44-3			SCK10/SCL10/SEG48/TS34/ EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/ TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/ (PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/ SCL11/(TKBO10) ^{Note 3} /(INTP0)	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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10, P11, P16, and P17 as outputs. P10 to P12 can set to analog input ^{Note 4} . P10 to P12 can be set to LCD output ^{Note 4} . P16 and P17 can be set to LCD output ^{Note 2} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/ SDA11/(TKBO11) ^{Note 3} / (TRJO1) ^{Note 3} /(INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/ SO11/(TKBO00)/(TRJO1) ^{Note 3} / (PCLBUZ1)	
P16	8-5-18		Digital input invalid ^{Note 1}	SEG41/SCK00/SCL00/EI16/ (VCOUT0)/(TKBO01)/ (RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRxD/ SDA00/EI17/EO17/EXSDI1/ (TKBO01)/(TRJO0)	
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/ (INTP7)/(PCLBUZ1)	Port 2. 5-bit I/O port. Input or output can be specified in 1-bit units. For P22, P26, and P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 5} . P22, P26, and P27 can be set to analog input ^{Note 4} . P22, P26, and P27 can be set to LCD output ^{Note 4} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/ (TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/ (SCK00)/(SCL00)/(INTP7)	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/ TxD1/SI21/SDA21/ (TKBO20) ^{Note 3} /(INTP5)/(RxD0)/ (SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/ RxD1/SO21/(TKBO21) ^{Note 3} / (INTP4)/(TxD0)/(SO00)/ (EXSDI1)	
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)	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. TTL input buffers can be set for P40 and P44 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P40 and P44 as outputs. P44 can be set to analog input ^{Note 5} .
P44	8-6-10		Analog input	IVCMP0/EXSDO1/(TI04)/ (TO04)/(INTP7)	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P57	7-44-1	I/O	Digital input invalid ^{Note 1}	INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	Port 5. 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. P57 can be set to LCD output ^{Note 2} .
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	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 (withstand voltage of 6 V). P60 and P61 can be set as controlled current drive port pins.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P76	8-45-2	I/O	Digital input invalid ^{Note 1}	KR6/SEG18/TS10/CCD01/TKBO00/RxD2/EI76/EO76/(INTP2)/(RTC1HZ)/(EXSDO1)	Port 7. 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 P76 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P76 and P77 as outputs. P76 and P77 can be set to LCD output ^{Note 2} . P76 and P77 can be set as controlled current drive port pins.
P77	7-45-2			KR7/SEG19/TS09/CCD00/TKBO01/TxD2/EI77/EO77/(TKBO21) ^{Note 3} /(TI07)/(TO07)/(INTP1)/(EXSDO0)	
P85	8-5-20	I/O	Digital input invalid ^{Note 1}	VL4/(VCOUT0)/(INTP4)/(SCK10)/(SCL10)/(SCL30)	Port 8. 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. TTL input buffers can be set for P85 and P86 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P85 to P87 as outputs.
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/(RxD1)/(SI10)/(SDA10)/(PCLBUZ1)	
P87	7-5-32			VL1/(TxD1)/(SO10)	
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/(INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 2} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/(SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/(SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/(TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/(SI01)/(SDA01)/(SDAA1)/(PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/(TO05)/(SCK01)/(SCL01)/(SCLA1)	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, 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 P125 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P125 as an output.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/(VCOUT1)/(INTP7)/(SDA30)/(PCLBUZ1)	
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/(INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/(REMOOUT)	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
RESET	2-1-1	—	—	—	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 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 3. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 4. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 5. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.1.2 48-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/EI00/EO00/EXSDI0/(TKBO01)/(TRJO0)	Port 0. 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 P05 and P06 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P00 and P05 to P07 as outputs. P00 and P05 to P07 can be set to LCD output ^{Note 2} .
P05	8-44-3			SCK10/SCL10/SEG48/TS34/EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/(PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/SCL11/(TKBO10) ^{Note 3} /(INTP0)	Port 1. 6-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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P10, P11, P16, and P17 as outputs. P10 to P13 can be set to analog input ^{Note 4} . P10 to P13 can be set to LCD output ^{Note 4} . P16 and P17 can be set to LCD output ^{Note 2} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/SDA11/(TKBO11) ^{Note 3} /(TRJO1) ^{Note 3} /(INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/SO11/(TKBO00)/(TRJO1) ^{Note 3} /(PCLBUZ1)	
P13	7-10-3			ANI25/SEG38	
P16	8-5-18		Digital input invalid ^{Note 1}	SEG41/SCK00/SCL00/EI16/(VCOUT0)/(TKBO01)/(RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRxD/SDA00/EI17/EO17/EXSDI1/(TKBO01)/(TRJO0)	
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/(INTP7)/(PCLBUZ1)	Port 2. 5-bit I/O port. Input or output can be specified in 1-bit units. For P22, P26, and P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 5} . P22, P26, and P27 can be set to analog input ^{Note 4} . P22, P26, and P27 can be set to LCD output ^{Note 4} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/(TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/(SCK00)/(SCL00)/(INTP7)	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/TxDA1/SI21/SDA21/(TKBO20) ^{Note 3} /(INTP5)/(RxD0)/(SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/RxDA1/SO21/(TKBO21) ^{Note 3} /(INTP4)/(TxD0)/(SO00)/(EXSDI1)	
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 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. TTL input buffers can be set for P40, P44, and P45 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P40, P44, and P45 as outputs. P44 and P45 can be set to analog input ^{Note 5} .
P44	8-6-10		Analog input	IVCMP0/RxDA2 ^{Note 3} /SDA30/EXSDO1/(TI04)/(TO04)/(INTP7)	
P45	8-6-10			IVREF0/TxDA2 ^{Note 3} /SCL30/EXSDO0	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P54	8-44-2	I/O	Digital input invalid ^{Note 1}	TI02/TO02/SEG8/TS19/(INTP0)/(PCLBUZ0)	Port 5. 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 P54 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P54 as an output. P54 and P57 can be set to LCD output ^{Note 2} .
P57	7-44-1			INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	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 (withstand voltage of 6 V). P60 and P61 can be set as controlled current drive port pins.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P75	8-45-1	I/O	Digital input invalid ^{Note 1}	KR5/SEG17/TS11/CCD02/SCK20/SCL20/(TKBO11) ^{Note 3} /(TKBO01)/(TI03)/(TO03)/(REMOOUT)/(INTP0)	Port 7. 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. TTL input buffers can be set for P75 and P76 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P75 to P77 as outputs. P75 to P77 can be set to LCD output ^{Note 2} . P75 to P77 can be set as controlled current drive port pins.
P76	8-45-2			KR6/SEG18/TS10/CCD01/TKBO00/SI20/SDA20/RxD2/EI76/EO76/(INTP2)/(RTC1HZ)/(EXSDO1)	
P77	7-45-2			KR7/SEG19/TS09/CCD00/TKBO01/SO20/TxD2/EI77/EO77/(TKBO21) ^{Note 3} /(TI07)/(TO07)/(INTP1)/(EXSDO0)	
P85	8-5-20	I/O	Digital input invalid ^{Note 1}	VL4/(VCOUT0)/(INTP4)/(SCK10)/(SCL10)/(SCL30)	Port 8. 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. TTL input buffers can be set for P85 and P86 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P85 to P87 as outputs.
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/(RxD1)/(SI10)/(SDA10)/(PCLBUZ1)	
P87	7-5-32			VL1/(TxD1)/(SO10)	
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/(INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 2} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/(SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/(SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/(TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/(SI01)/(SDA01)/(SDAA1)/(PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/(TO05)/(SCK01)/(SCL01)/(SCLA1)	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, 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 P125 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P125 as an output.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/(INTP7)/(SDA30)/(PCLBUZ1)	
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/(INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/(REMOOUT)	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
RESET	2-1-1	—	—	—	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 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 3. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 4. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 5. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.1.3 52-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/ EI00/EO00/EXSDI0/(TKBO01)/(TRJO0)	Port 0. 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 P05 and P06 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00 and P05 to P07 as outputs. P00 and P05 to P07 can be set to LCD output ^{Note 2} .
P05	8-44-3			SCK10/SCL10/SEG48/TS34/EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/(PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/SCL11/(TKBO10) ^{Note 3} /(INTP0)	Port 1. 7-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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10, P11, P16, and P17 as outputs. P10 to P14 can be set to analog input ^{Note 4} . P10 to P14 can be set to LCD output ^{Note 4} . P16 and P17 can be set to LCD output ^{Note 2} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/SDA11/(TKBO11) ^{Note 3} /(TRJO1) ^{Note 3} /(INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/SO11/(VCOUT1)/(TKBO00)/(TRJO1) ^{Note 3} /(PCLBUZ1)	
P13	7-10-3			ANI25/SEG38	
P14	7-10-3			TI04/TO04/SEG39/ANI26	
P16	8-5-18		Digital input invalid ^{Note 1}	SEG41/SCK00/SCL00/EI16/(VCOUT0)/(TKBO01)/(RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRxD/SDA00/EI17/EO17/EXSDI1/(TKBO01)/(TRJO0)	
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/(INTP7)/(PCLBUZ1)	Port 2. 5-bit I/O port. Input or output can be specified in 1-bit units. For P22, P26, and P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 5} . P22, P26, and P27 can be set to analog input ^{Note 4} . P22, P26, and P27 can be set to LCD output ^{Note 4} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/(TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/(SCK00)/(SCL00)/(INTP7)	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/TxDA1/SI21/SDA21/(TKBO20) ^{Note 3} /(INTP5)/(RxD0)/(SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/RxDA1/SO21/(TKBO21) ^{Note 3} /(INTP4)/(TxD0)/(SO00)/(EXSDI1)	
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)	Port 4. 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 P40 and P43 to P45 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P40 and P43 to P45 as outputs. P43 to P45 can be set to analog input ^{Note 5} .
P43	8-6-10		Analog input	IVCMP1/SO30/TxD3/(TI05)/(TO05)/(INTP4)/(SDA11)	
P44	8-6-10			IVCMP0/RxDA2 ^{Note 3} /SI30/SDA30/RxD3/EXSDO1/(TI04)/(TO04)/(INTP7)/(SCL11)	
P45	8-6-10			IVREF0/TxDA2 ^{Note 3} /SCK30/SCL30/EXSDO0	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P53	8-44-2	I/O	Digital input invalid ^{Note 1}	INTP2/SEG7/TS20/SDA01/(TI07)/(TO07)/(PCLBUZ0)	Port 5. 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. TTL input buffers can be set for P53 and P54 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P53 and P54 as outputs. P53, P54, and P57 can be set to LCD output ^{Note 2} .
P54	8-44-2			TI02/TO02/SEG8/TS19/SCL01/(INTP0)/(PCLBUZ0)	
P57	7-44-1			INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	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 (withstand voltage of 6 V). P60 and P61 can be set as controlled current drive port pins.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P74	7-45-1	I/O	Digital input invalid ^{Note 1}	KR4/SEG16/TS12/CCD03/TKBO11 ^{Note 3} /(TKBO01)	Port 7. 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 P75 and P76 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P75 to P77 as outputs. P74 to P77 can be set to LCD output ^{Note 2} . P74 to P77 can be set as controlled current drive port pins.
P75	8-45-1			KR5/SEG17/TS11/CCD02/SCK20/SCL20/(TKBO11) ^{Note 3} /(TKBO01)/(TI03)/(TO03)/(REMOOUT)/(INTP0)	
P76	8-45-2			KR6/SEG18/TS10/CCD01/TKBO00/SI20/SDA20/RxD2/EI76/EO76/(INTP2)/(RTC1HZ)/(EXSDO1)	
P77	7-45-2			KR7/SEG19/TS09/CCD00/TKBO01/SO20/TxD2/EI77/EO77/(TKBO21) ^{Note 3} /(TI07)/(TO07)/(INTP1)/(EXSDO0)	
P85	8-5-20	I/O	Digital input invalid ^{Note 1}	VL4/(VCOU0)/(INTP4)/(SCK10)/(SCL10)/(SCL30)	Port 8. 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. TTL input buffers can be set for P85 and P86 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P85 to P87 as outputs.
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/(RxD1)/(SI10)/(SDA10)/(PCLBUZ1)	
P87	7-5-32			VL1/(TxD1)/(SO10)	
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/(INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 2} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/(SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/(SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/(TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/(SI01)/(SDA01)/(SDAA1)/(PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/(TO05)/(SCK01)/(SCL01)/(SCLA1)	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, 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 P125 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P125 as an output.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/(VCOUT1)/(INTP7)/(SDA30)/(PCLBUZ1)	
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/(INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/(REMOOUT)	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
RESET	2-1-1	—	—	—	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 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 3. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 4. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 5. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.1.4 64-pin products

(1/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/ EI00/EO00/EXSDI0/(TKBO01)/ (TRJ00)	Port 0. 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 P02, P03, P05, and P06 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P00 to P07 as outputs. P00 to P07 can be set to LCD output ^{Note 3} .
P01	7-5-30			PCLBUZ1/SEG44/(TI05)/ (TO05)/(TKBO11) ^{Note 2} / (INTP5)/(SO31)	
P02	8-5-18			INTP7/PCLBUZ0/SEG45/ (SCK20)/(SCL20)/(SCK31)/ (SCL31)	
P03	8-5-17			SEG46/VCOU0/TRJIO0/ (RxD2)/(SI20)/(SDA20)/(SI31)/ (SDA31)	
P04	7-44-3			SEG47/VCOU1/TS35/(INTP6)/ (SO20)/(TxD2)/(PCLBUZ0)	
P05	8-44-3			SCK10/SCL10/SEG48/TS34/ EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/ TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/ (PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/ SCL11/(TKBO10) ^{Note 2} /(INTP0)	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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P10, P11, P16, and P17 as outputs. P10 to P14 can be set to analog input ^{Note 4} . P10 to P14 can be set to LCD output ^{Note 4} . P15 to P17 can be set to LCD output ^{Note 3} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/ SDA11/(TKBO11) ^{Note 2} / (TRJIO1) ^{Note 2} /(INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/ SO11/(VCOU1)/(TKBO00)/ (TRJIO1) ^{Note 2} /(PCLBUZ1)	
P13	7-10-3			ANI25/SEG38	
P14	7-10-3			TI04/TO04/SEG39/ANI26	
P15	7-5-4			TI07/TO07/SEG40	
P16	8-5-18		Digital input invalid ^{Note 1}	SEG41/SCK00/SCL00/EI16/ (VCOU0)/(TKBO01)/ (RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRxD/ SDA00/EI17/EO17/EXSDI1/ (TKBO01)/(TRJIO0)	

(2/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/(INTP7)/(PCLBUZ1)	Port 2. 5-bit I/O port. Input or output can be specified in 1-bit units. For P22, P26, and P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 5} . P22, P26, and P27 can be set to analog input ^{Note 4} . P22, P26, and P27 can be set to LCD output ^{Note 4} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/(TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/(SCK00)/(SCL00)/(INTP7)	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/TxDA1/SI21/SDA21/(TKBO20) ^{Note 2} /(INTP5)/(RxD0)/(SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/RxDA1/SO21/(TKBO21) ^{Note 2} /(INTP4)/(TxD0)/(SO00)/(EXSDI1)	
P30	7-44-1	I/O	Digital input invalid ^{Note 1}	TI03/TO03/SEG20/REMOOUT/TS08/(INTP3)/(RTC1HZ)	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. P30 and P31 can be set to LCD output ^{Note 3} .
P31	7-44-1			INTP3/RTC1HZ/SEG21/TS07/(TI01)/(TO01)	
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)/(SCL00)	Port 4. 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 P40 and P42 to P45 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P40 and P42 to P45 as outputs. P42 to P45 can be set to analog input ^{Note 5} .
P42	8-6-10		Analog input	TI05/TO05/IVREF1/TRJ00/(INTP7)/(TxDA1)/(SDA00)/(SO11)/(PCLBUZ1)	
P43	8-6-10			IVCMP1/SO30/TxD3/(TI05)/(TO05)/(INTP4)/(RxDA1)/(SI11)/(SDA11)	
P44	8-6-10			IVCMP0/RxDA2 ^{Note 2} /SI30/SDA30/RxD3/EXSD01/(TI04)/(TO04)/(INTP7)/(SCK11)/(SCL11)	
P45	8-6-10			IVREF0/TxDA2 ^{Note 2} /SCK30/SCL30/EXSD00	
P52	7-44-3	I/O	Digital input invalid ^{Note 1}	TI00/TO00/INTP1/SEG6/TS21/SO01/(PCLBUZ1)	Port 5. 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 P53 and P54 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P52 to P54 as outputs. P52 to P54 and P57 can be set to LCD output ^{Note 3} .
P53	8-44-2			INTP2/SEG7/TS20/SI01/SDA01/(TI07)/(TO07)/(PCLBUZ0)	
P54	8-44-2			TI02/TO02/SEG8/TS19/SCK01/SCL01/(INTP0)/(PCLBUZ0)	
P57	7-44-1			INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	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 (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P62	12-38-2			SCLA1/CCD06/(TI03)/(TO03)/(REMOOUT)/(INTP5)	
P63	12-38-2			SDAA1/CCD07/(TI04)/(TO04)/(INTP6)	

(3/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P70	7-44-2	I/O	Digital input invalid ^{Note 1}	KR0/SEG12/TS16/ TKBO20 ^{Note 2}	Port 7. 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 P75 and P76 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P75 to P77 as outputs. P70 and P74 to P77 can be set to LCD output ^{Note 3} . P74 to P77 can be set as controlled current drive port pins.
P74	7-45-1			KR4/SEG16/TS12/CCD03/ TKBO11 ^{Note 2} /(TKBO01)	
P75	8-45-1			KR5/SEG17/TS11/CCD02/ SCK20/SCL20/(TKBO11) ^{Note 2} / (TKBO01)/(TI03)/(TO03)/ (REMOOUT)/(INTP0)	
P76	8-45-2			KR6/SEG18/TS10/CCD01/ TKBO00/SI20/SDA20/RxD2/ EI76/EO76/(INTP2)/(RTC1HZ)/ (EXSDO1)	
P77	7-45-2			KR7/SEG19/TS09/CCD00/ TKBO01/SO20/TxD2/EI77/ EO77/(TKBO21) ^{Note 2} /(TI07)/ (TO07)/(INTP1)/(EXSDO0)	
P85	8-5-20	I/O	Digital input invalid ^{Note 1}	VL4/(VCOUT0)/(INTP4)/ (SCK10)/(SCL10)/(SCL30)	Port 8. 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. TTL input buffers can be set for P85 and P86 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P85 to P87 as outputs.
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/ (RxD1)/(SI10)/(SDA10)/ (PCLBUZ1)	
P87	7-5-32			VL1/(TxD1)/(SO10)	
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/ (INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 3} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/ (SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/ (SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/ (TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/ (SI01)/(SDA01)/(SDAA1)/ (PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/ (TO05)/(SCK01)/(SCL01)/ (SCLA1)	
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/ (VCOUT1)/(INTP7)/(SDA30)/ (PCLBUZ1)	A TTL input buffer can be set for P125 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P125 as an output.
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/ (INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/ (REMOOUT)	

(4/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
$\overline{\text{RESET}}$	2-1-1	—	—	—	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 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 3. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 4. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 5. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.1.5 80-pin products

(1/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/ EI00/EO00/EXSDI0/(TKBO01)/ (TRJO0)	Port 0. 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 P02, P03, P05, and P06 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P00 to P07 as outputs. P00 to P07 can be set to LCD output ^{Note 2} .
P01	7-5-30			PCLBUZ1/SEG44/(TI05)/ (TO05)/(TKBO11)/(INTP5)/ (SO31)	
P02	8-5-18			INTP7/PCLBUZ0/SEG45/ (TKBO21)/(SCK20)/(SCL20)/ (SCK31)/(SCL31)	
P03	8-5-17			SEG46/VCOUT0/TRJIO0/ (RxD2)/(SI20)/(SDA20)/(SI31)/ (SDA31)	
P04	7-44-3			SEG47/VCOUT1/TS35/(INTP6)/ (SO20)/(TxD2)/(PCLBUZ0)	
P05	8-44-3			SCK10/SCL10/SEG48/TS34/ EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/ TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/ (PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/ SCL11/(TKBO10)/(INTP0)	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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P10, P11, P16, and P17 as outputs. P10 to P14 can be set to analog input ^{Note 3} . P10 to P14 can be set to LCD output ^{Note 3} . P15 to P17 can be set to LCD output ^{Note 2} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/ SDA11/(TKBO11)/(TRJIO1)/ (INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/ SO11/(VCOUT1)/(TKBO00)/ (TRJO1)/(PCLBUZ1)	
P13	7-10-3			ANI25/SEG38	
P14	7-10-3			TI04/TO04/SEG39/ANI26	
P15	7-5-4		Digital input invalid ^{Note 1}	TI07/TO07/SEG40	
P16	8-5-18			SEG41/SCK00/SCL00/EI16/ (VCOUT0)/(TKBO01)/ (RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRxD/ SDA00/EI17/EO17/EXSDI1/ (TKBO01)/(TRJIO0)	

(2/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/(INTP7)/(PCLBUZ1)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. For P22 to P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 4} . P22 to P27 can be set to analog input ^{Note 3} . P22 to P27 can be set to LCD output ^{Note 3} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/(TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/(SCK00)/(SCL00)/(INTP7)	
P23	7-10-3			ANI17/SEG30/TRJIO1	
P24	7-10-3			ANI18/SEG31	
P25	7-10-3			ANI19/SEG32/CLKA1	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/TxDA1/SI21/SDA21/(TKBO20)/(INTP5)/(RxD0)/(SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/RxDA1/SO21/(TKBO21)/(INTP4)/(TxD0)/(SO00)/(EXSDI1)	
P30	7-44-1	I/O	Digital input invalid ^{Note 1}	TI03/TO03/SEG20/REMOOUT/TS08/(INTP3)/(RTC1HZ)	Port 3. 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. A TTL input buffer can be set for P34 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P34 and P35 as outputs. P30, P31, P34, and P35 can be set to LCD output ^{Note 2} .
P31	7-44-1			INTP3/RTC1HZ/SEG21/TS07/(TI01)/(TO01)	
P34	8-44-2			SEG24/TS04/(TI01)/(TO01)/(INTP0)/(RxD3)	
P35	7-44-3			SEG25/TS03/(VCOUT0)/(INTP4)/(SO30)/(TxD3)	
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)/(SCK00)/(SCL00)	Port 4. 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 P40 to P47 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P40 to P47 as outputs. P42 to P45 can be set to analog input ^{Note 4} . P46 and P47 can be set to LCD output ^{Note 2} .
P41	8-1-10			(TI07)/(TO07)/(INTP6)/(TxD0)/(SDAA0)/(SO00)/(RxD1)	
P42	8-6-10		Analog input	TI05/TO05/IVREF1/TRJO0/(INTP7)/(TxDA1)/(SCLA0)/(RxD0)/(TxD1)/(SI00)/(SDA00)/(SO11)/(PCLBUZ1)	
P43	8-6-10			IVCMP1/SO30/TxD3/(TI05)/(TO05)/(INTP4)/(RxDA1)/(SI11)/(SDA11)	
P44	8-6-10			IVCMP0/RxDA2/SI30/SDA30/RxD3/EXSDO1/(TI04)/(TO04)/(INTP7)/(SCK11)/(SCL11)	
P45	8-6-10			IVREF0/TxDA2/SCK30/SCL30/EXSDO0	
P46	8-5-17		Digital input invalid ^{Note 1}	SEG26/(SCK21)/(SCL21)/(SCL01)/(SCK01)	
P47	8-5-17			SEG27/(RxDA0)/(SI21)/(SDA21)/(SDA01)/(SI01)/(INTP1)	

(3/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-44-3	I/O	Digital input invalid ^{Note 1}	SEG4/TS23/TxDA3	Port 5. 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 P51, P53, and P54 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P50 and P52 to P54 as outputs. P50 to P57 can be set to LCD output ^{Note 2} .
P51	8-44-1			SEG5/TS22/RxDA3/(INTP6)	
P52	7-44-3			TI00/TO00/INTP1/SEG6/TS21/SO01/(PCLBUZ1)	
P53	8-44-2			INTP2/SEG7/TS20/SI01/SDA01/(TI07)/(TO07)/(PCLBUZ0)	
P54	8-44-2			TI02/TO02/SEG8/TS19/SCK01/SCL01/(INTP0)/(PCLBUZ0)	
P55	7-44-1			INTP5/SEG9/TS18/(TI02)/(TO02)	
P56	7-44-1			TI06/TO06/SEG10/TS17	
P57	7-44-1			INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	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 (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P62	12-38-2			SCLA1/CCD06/(TI03)/(TO03)/(REMOOUT)/(INTP5)	
P63	12-38-2			SDAA1/CCD07/(TI04)/(TO04)/(INTP6)	
P70	7-44-2	I/O	Digital input invalid ^{Note 1}	KR0/SEG12/TS16/TKBO20	Port 7. 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 P75 and P76 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P75 to P77 as outputs. P70 to P77 can be set to LCD output ^{Note 2} . P74 to P77 can be set as controlled current drive port pins.
P71	7-44-1			KR1/SEG13/TS15	
P72	7-44-2			KR2/SEG14/TS14/TKBO21	
P73	7-44-2			KR3/SEG15/TS13/TKBO10	
P74	7-45-1			KR4/SEG16/TS12/CCD03/TKBO11/(TKBO01)	
P75	8-45-1			KR5/SEG17/TS11/CCD02/SCK20/SCL20/(TKBO11)/(TKBO01)/(TI03)/(TO03)/(REMOOUT)/(INTP0)	
P76	8-45-2			KR6/SEG18/TS10/CCD01/TKBO00/SI20/SDA20/RxD2/EI76/EO76/(INTP2)/(RTC1HZ)/(EXSD01)	
P77	7-45-2			KR7/SEG19/TS09/CCD00/TKBO01/SO20/TxD2/EI77/EO77/(TKBO21)/(TI07)/(TO07)/(INTP1)/(EXSD00)	
P85	8-5-20	I/O	Digital input invalid ^{Note 1}	VL4/(VCOUT0)/(INTP4)/(SCK10)/(SCL10)/(SCK30)/(SCL30)	Port 8. 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. TTL input buffers can be set for P85 and P86 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P85 to P87 as outputs.
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/(RxD1)/(SI10)/(SDA10)/(PCLBUZ1)	
P87	7-5-32			VL1/(TxD1)/(SO10)	

(4/4)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/(INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 2} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/(SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/(SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/(TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/(SI01)/(SDA01)/(SDAA1)/(PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/(TO05)/(SCK01)/(SCL01)/(SCLA1)	
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/(VCOUT1)/(INTP7)/(RxD3)/(SI30)/(SDA30)/(PCLBUZ1)	A TTL input buffer can be set for P125 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P125 as an output.
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/(INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/(REMOOUT)	
P130	1-5-1	Output	Output mode	SEG28/TRJO1/(TxDA0)/(SO21)/(SO01)	Port 13. 1-bit output-only port and 1-bit input-only port. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P130 as an output. P130 can be set to LCD output ^{Note 2} .
P137	2-1-3	Input	Input port	EI137/INTP0	
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 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 3. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 4. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.1.6 100-pin products

(1/5)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-5-31	I/O	Digital input invalid ^{Note 1}	SEG43/SO00/TxD0/TOOLTxD/ EI00/EO00/EXSDI0/(TKBO01)/ (TRJO0)	Port 0. 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 P02, P03, P05, and P06 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P00 to P07 as outputs. P00 to P07 can be set to LCD output ^{Note 2} .
P01	7-5-30			PCLBUZ1/SEG44/(TI05)/ (TO05)/(TKBO11)/(INTP5)/ (SO31)	
P02	8-5-18			INTP7/PCLBUZ0/SEG45/ (TKBO21)/(SCK20)/(SCL20)/ (SCK31)/(SCL31)	
P03	8-5-17			SEG46/VCOUT0/TRJIO0/ (TxDA2)/(RxD2)/(SI20)/ (SDA20)/(SI31)/(SDA31)	
P04	7-44-3			SEG47/VCOUT1/TS35/(INTP6)/ (SO20)/(TxD2)/(PCLBUZ0)	
P05	8-44-3			SCK10/SCL10/SEG48/TS34/ EO05	
P06	8-44-2			SI10/RxD1/SDA10/SEG49/ TS33/EI06/(PCLBUZ1)	
P07	7-44-3			SO10/TxD1/SEG50/TS32/EI07/ (PCLBUZ0)	
P10	7-10-6	I/O	Analog input	ANI22/SEG35/TxDA0/SCK11/ SCL11/(TKBO10)/(INTP0)	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 P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P10, P11, P16, and P17 as outputs. P10 to P14 can be set to analog input ^{Note 3} . P10 to P14 can be set to LCD output ^{Note 3} . P15 to P17 can be set to LCD output ^{Note 2} .
P11	8-10-2			ANI23/SEG36/RxDA0/SI11/ SDA11/(TKBO11)/(TRJIO1)/ (INTP1)	
P12	7-10-5			ANI24/SEG37/EI12/CLKA0/ SO11/(VCOUT1)/(TKBO00)/ (TRJO1)/(PCLBUZ1)	
P13	7-10-3			ANI25/SEG38	
P14	7-10-3			TI04/TO04/SEG39/ANI26	
P15	7-5-4		Digital input invalid ^{Note 1}	TI07/TO07/SEG40	
P16	8-5-18			SEG41/SCK00/SCL00/EI16/ (VCOUT0)/(TKBO01)/ (RTC1HZ)	
P17	8-5-19			SEG42/SI00/RxD0/TOOLRx/D/ SDA00/EI17/EO17/EXSDI1/ (TKBO01)/(TRJIO0)	

(2/5)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-3-5	I/O	Analog input	ANI1/AVREFM/EI20/(TI00)/(INTP7)/(PCLBUZ1)	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. For P22 to P27, 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 P22, P26, and P27 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P22, P26, and P27 as outputs. P20 and P21 can be set to analog input ^{Note 4} . P22 to P27 can be set to analog input ^{Note 3} . P22 to P27 can be set to LCD output ^{Note 3} .
P21	4-3-5			ANI0/AVREFP/VBAT1/EI21/(TO00)/(INTP6)/(PCLBUZ0)	
P22	8-10-1			ANI16/SEG29/SCK21/SCL21/(SCK00)/(SCL00)/(INTP7)	
P23	7-10-3			ANI17/SEG30/TRJIO1	
P24	7-10-3			ANI18/SEG31	
P25	7-10-3			ANI19/SEG32/CLKA1	
P26	8-43-1			ANI20/SEG33/ANO0/EI26/TxDA1/SI21/SDA21/(TKBO20)/(INTP5)/(RxD0)/(SI00)/(SDA00)/(EXSDI0)	
P27	8-43-1			ANI21/SEG34/ANO1/EI27/RxDA1/SO21/(TKBO21)/(INTP4)/(TxD0)/(SO00)/(EXSDI1)	
P30	7-44-1	I/O	Digital input invalid ^{Note 1}	TI03/TO03/SEG20/REMOOUT/TS08/(INTP3)/(RTC1HZ)	Port 3. 6-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 P33 and P34 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P33 to P35 as outputs. P30 to P35 can be set to LCD output ^{Note 2} .
P31	7-44-1			INTP3/RTC1HZ/SEG21/TS07/(TI01)/(TO01)	
P32	7-44-1			TI01/TO01/SEG22/TS06/(TI03)/(TO03)/(REMOOUT)/(INTP3)/(RTC1HZ)	
P33	8-44-2			INTP4/SEG23/TS05/(TI01)/(TO01)/(RTC1HZ)/(SCK30)/(SCL30)	
P34	8-44-2			SEG24/TS04/(TI01)/(TO01)/(INTP0)/(SI30)/(RxD3)/(SDA30)	
P35	7-44-3			SEG25/TS03/(VCOUT0)/(INTP4)/(SO30)/(TxD3)	

(3/5)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P40	8-1-10	I/O	Input port	TOOL0/(TI00)/(TO00)/(SCK00)/(SCL00)	Port 4. 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 P40 to P47 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P40 to P47 as outputs. P42 to P45 can be set to analog input ^{Note 4} . P46 and P47 can be set to LCD output ^{Note 2} .
P41	8-1-10			(TI07)/(TO07)/(INTP6)/(TxD0)/(RxD A3)/(SDA A0)/(SO00)/(SI10)/(SDA10)/(RxD1)	
P42	8-6-10		Analog input	TI05/TO05/IVREF1/TRJ00/(INTP7)/(TxDA1)/(SCLA0)/(RxD0)/(TxD1)/(SI00)/(SDA00)/(SO10)/(SO11)/(PCLBUZ1)	
P43	8-6-10			IVCMP1/SO30/TxD3/(TI05)/(TO05)/(INTP4)/(RxD A1)/(SI11)/(SDA11)	
P44	8-6-10			IVCMP0/RxD A2/SI30/SDA30/RxD3/EXSDO1/(TI04)/(TO04)/(INTP7)/(SCK11)/(SCL11)	
P45	8-6-10			IVREF0/TxDA2/SCK30/SCL30/EXSDO0	
P46	8-5-17		Digital input invalid ^{Note 1}	SEG26/SI31/SDA31/(SCK21)/(SCL21)/(SCL01)/(SCK01)	
P47	8-5-17			SEG27/SO31/(RxD A0)/(SI21)/(SDA21)/(SDA01)/(SI01)/(INTP1)	
P50	7-44-3	I/O	Digital input invalid ^{Note 1}	SEG4/TS23/TxDA3	Port 5. 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 P51, P53, and P54 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P50 and P52 to P54 as outputs. P50 to P57 can be set to LCD output ^{Note 2} .
P51	8-44-1			SEG5/TS22/RxDA3/(INTP6)	
P52	7-44-3			TI00/TO00/INTP1/SEG6/TS21/SO01/(PCLBUZ1)	
P53	8-44-2			INTP2/SEG7/TS20/SI01/SDA01/(TI07)/(TO07)/(PCLBUZ0)	
P54	8-44-2			TI02/TO02/SEG8/TS19/SCK01/SCL01/(INTP0)/(PCLBUZ0)	
P55	7-44-1			INTP5/SEG9/TS18/(TI02)/(TO02)	
P56	7-44-1			TI06/TO06/SEG10/TS17	
P57	7-44-1			INTP6/SEG11/TSCAP/(TI03)/(TO03)/(REMOOUT)	

(4/5)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P60	12-38-3	I/O	Input port	SCLA0/CCD04/EO60/(TI01)/(TO01)/(INTP3)	Port 6. 8-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 (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins. For P64 to P67, 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 P64 to P66 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P64 to P66 as outputs.
P61	12-38-3			SDAA0/CCD05/EO61/(TI02)/(TO02)/(INTP4)	
P62	12-38-2			SCLA1/CCD06/(TI03)/(TO03)/(REMOOUT)/(INTP5)	
P63	12-38-2			SDAA1/CCD07/(TI04)/(TO04)/(INTP6)	
P64	8-1-10			(INTP5)/(SDAA1)/(TI03)/(TO03)/(REMOOUT)	
P65	8-1-10			(INTP0)/(SCLA1)/(TI05)/(TO05)	
P66	8-1-10			(TI05)/(TO05)/(INTP1)/(SCK10)/(SCL10)/(TxDA3)	
P67	7-1-3			(INTP2)/(TI01)/(TO01)	
P70	7-44-2	I/O	Digital input invalid ^{Note 1}	KR0/SEG12/TS16/TKBO20	Port 7. 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 P75 and P76 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P75 to P77 as outputs. P70 to P77 can be set to LCD output ^{Note 2} . P74 to P77 can be set as controlled current drive port pins.
P71	7-44-1			KR1/SEG13/TS15	
P72	7-44-2			KR2/SEG14/TS14/TKBO21	
P73	7-44-2			KR3/SEG15/TS13/TKBO10	
P74	7-45-1			KR4/SEG16/TS12/CCD03/TKBO11/(TKBO01)	
P75	8-45-1			KR5/SEG17/TS11/CCD02/SCK20/SCL20/(TKBO11)/(TKBO01)/(TI03)/(TO03)/(REMOOUT)/(INTP0)	
P76	8-45-2			KR6/SEG18/TS10/CCD01/TKBO00/SI20/SDA20/RxD2/EI76/EO76/(INTP2)/(RTC1HZ)/(EXSDO1)	
P77	7-45-2			KR7/SEG19/TS09/CCD00/TKBO01/SO20/TxD2/EI77/EO77/(TKBO21)/(TI07)/(TO07)/(INTP1)/(EXSDO0)	
P80	8-1-10	I/O	Input port	(INTP2)/(SCK20)/(SCL20)	Port 8. 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 P80 to P86 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P80 to P87 as outputs.
P81	8-1-10			(RxD1)/(SI10)/(SDA10)	
P82	8-1-10			(SCK10)/(SCL10)/(PCLBUZ0)	
P83	8-1-10			(INTP1)/(RxD0)/(SDAA1)	
P84	8-1-10			(INTP0)/(TxDA0)/(SCLA1)	
P85	8-5-20		Digital input invalid ^{Note 1}	VL4/(VCOUT0)/(INTP4)/(SCK10)/(SCL10)/(SCK30)/(SCL30)	
P86	8-5-20			VL2/(TI06)/(TO06)/(INTP3)/(RxD1)/(SI10)/(SDA10)/(PCLBUZ1)	
P87	7-5-32			VL1/(TxDA1)/(SO10)	

(5/5)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P90	7-44-3	I/O	Digital input invalid ^{Note 1}	COM0/TS31/(TI00)/(TO00)/(INTP1)	Port 9. 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 P93, P94, P96, and P97 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P93 to P97 as outputs. P90 to P97 can be set to LCD output ^{Note 2} .
P91	7-44-3			COM1/TS30/(INTP2)	
P92	7-44-3			COM2/TS29/(TI01)/(TO01)	
P93	8-44-2			COM3/TS28/(INTP3)/(SCK01)/(SCL01)	
P94	8-44-2			COM4/SEG0/TS27/(INTP2)/(SI01)/(SDA01)/(TI06)/(TO06)	
P95	7-44-3			COM5/SEG1/TS26/(TI02)/(TO02)/(INTP1)/(SO01)	
P96	8-44-2			COM6/SEG2/TS25/(INTP5)/(SI01)/(SDA01)/(SDAA1)/(PCLBUZ0)	
P97	8-44-2			COM7/SEG3/TS24/(TI05)/(TO05)/(SCK01)/(SCL01)/(SCLA1)	
P121	7-2-1	I/O	Input port	X1/VBAT0/EI121	Port 12. 5-bit I/O port and 2-bit input-only port. For P121, P122, and P125 to P127, input or output can be specified in 1-bit units. For P121, P122, and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122	7-2-1			X2/EXCLK/EI122	
P123	2-2-1			XT1	
P124	2-2-1			XT2/EXCLKS	
P125	8-44-4	I/O	Digital input invalid ^{Note 1}	VL3/TS02/(TI06)/(TO06)/(VCOUT1)/(INTP7)/(RxD3)/(SI30)/(SDA30)/(PCLBUZ1)	A TTL input buffer can be set for P125 as an input. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P125 as an output.
P126	7-44-4			CAPL/TS01/(TI04)/(TO04)/(INTP4)	
P127	7-44-4			CAPH/TS00/(TI03)/(TO03)/(REMOOUT)	
P130	1-5-1	Output	Output mode	SEG28/TRJ01/(TxDA0)/(SO21)/(SO01)	Port 13. 1-bit output-only port and 1-bit input-only port. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P130 as an output. P130 can be set to LCD output ^{Note 2} .
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	8-1-10	I/O	Input port	SCK31/SCL31	Port 14. 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 P140 to P142 and P144 to P147 as inputs. N-ch open-drain output (withstand voltage of V _{DD}) can be set for P140 and P143 to P147 as outputs. P141 to P145 can be set to LCD output ^{Note 2} .
P141	8-5-4		Digital input invalid ^{Note 1}	SEG51/(INTP6)/(RxDA3)	
P142	8-5-4			SEG52/(INTP7)/(RxDA2)	
P143	7-5-33			SEG53/(INTP5)/(TxDA2)	
P144	8-5-17			SEG54/(INTP4)/(RxDA1)/(SCLA0)	
P145	8-5-17			SEG55/(INTP3)/(TxDA1)/(SDAA0)	
P146	8-1-10		Input port	(VCOUT1)/(TxD2)/(SO20)/(SO31)/(RxDA2)/(INTP7)	
P147	8-1-10			(INTP2)/(TxDA3)/(RxD2)/(SI20)/(SDA20)	
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 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

Note 2. Digital or LCD can be selected per pin (in 1-bit units) with the port mode register (PMxx) and LCD port function register (PFSEGx).

Note 3. Digital, analog, or LCD can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx), port mode register (PMxx), and LCD port function register (PFSEGx).

Note 4. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

2.2 Pin Functions Other than Port Pin Functions

2.2.1 Functions for each product

(1/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
ANI0	✓	✓	✓	✓	✓	✓
ANI1	✓	✓	✓	✓	✓	✓
ANI16	✓	✓	✓	✓	✓	✓
ANI17	✓	✓	—	—	—	—
ANI18	✓	✓	—	—	—	—
ANI19	✓	✓	—	—	—	—
ANI20	✓	✓	✓	✓	✓	✓
ANI21	✓	✓	✓	✓	✓	✓
ANI22	✓	✓	✓	✓	✓	✓
ANI23	✓	✓	✓	✓	✓	✓
ANI24	✓	✓	✓	✓	✓	✓
ANI25	✓	✓	✓	✓	✓	—
ANI26	✓	✓	✓	✓	—	—
ANO0	✓	✓	✓	✓	✓	✓
ANO1	✓	✓	✓	✓	✓	✓
IVCMP0	✓	✓	✓	✓	✓	✓
IVCMP1	✓	✓	✓	✓	—	—
IVREF0	✓	✓	✓	✓	✓	—
IVREF1	✓	✓	✓	—	—	—
VCOU0	✓	✓	✓	(✓)	(✓)	(✓)
VCOU1	✓	✓	✓	(✓)	—	—
TS00	✓	✓	✓	✓	✓	✓
TS01	✓	✓	✓	✓	✓	✓
TS02	✓	✓	✓	✓	✓	✓
TS03	✓	✓	—	—	—	—
TS04	✓	✓	—	—	—	—
TS05	✓	—	—	—	—	—
TS06	✓	—	—	—	—	—
TS07	✓	✓	✓	—	—	—
TS08	✓	✓	✓	—	—	—
TS09	✓	✓	✓	✓	✓	✓
TS10	✓	✓	✓	✓	✓	✓
TS11	✓	✓	✓	✓	✓	—
TS12	✓	✓	✓	✓	—	—
TS13	✓	✓	—	—	—	—
TS14	✓	✓	—	—	—	—

(2/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
TS15	✓	✓	—	—	—	—
TS16	✓	✓	✓	—	—	—
TS17	✓	✓	—	—	—	—
TS18	✓	✓	—	—	—	—
TS19	✓	✓	✓	✓	✓	—
TS20	✓	✓	✓	✓	—	—
TS21	✓	✓	✓	—	—	—
TS22	✓	✓	—	—	—	—
TS23	✓	✓	—	—	—	—
TS24	✓	✓	✓	✓	✓	✓
TS25	✓	✓	✓	✓	✓	✓
TS26	✓	✓	✓	✓	✓	✓
TS27	✓	✓	✓	✓	✓	✓
TS28	✓	✓	✓	✓	✓	✓
TS29	✓	✓	✓	✓	✓	✓
TS30	✓	✓	✓	✓	✓	✓
TS31	✓	✓	✓	✓	✓	✓
TS32	✓	✓	✓	✓	✓	✓
TS33	✓	✓	✓	✓	✓	✓
TS34	✓	✓	✓	✓	✓	✓
TS35	✓	✓	✓	—	—	—
TSCAP	✓	✓	✓	✓	✓	✓
CCD00	✓	✓	✓	✓	✓	✓
CCD01	✓	✓	✓	✓	✓	✓
CCD02	✓	✓	✓	✓	✓	—
CCD03	✓	✓	✓	✓	—	—
CCD04	✓	✓	✓	✓	✓	✓
CCD05	✓	✓	✓	✓	✓	✓
CCD06	✓	✓	✓	—	—	—
CCD07	✓	✓	✓	—	—	—
INTP0	✓	✓	✓	✓	✓	✓
INTP1	✓	✓	✓	(✓)	(✓)	(✓)
INTP2	✓	✓	✓	✓	(✓)	(✓)
INTP3	✓	✓	✓	(✓)	(✓)	(✓)
INTP4	✓	(✓)	(✓)	(✓)	(✓)	(✓)
INTP5	✓	✓	(✓)	(✓)	(✓)	(✓)
INTP6	✓	✓	✓	✓	✓	✓
INTP7	✓	✓	✓	(✓)	(✓)	(✓)
KR0	✓	✓	✓	—	—	—
KR1	✓	✓	—	—	—	—

(3/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
KR2	✓	✓	—	—	—	—
KR3	✓	✓	—	—	—	—
KR4	✓	✓	✓	✓	—	—
KR5	✓	✓	✓	✓	✓	—
KR6	✓	✓	✓	✓	✓	✓
KR7	✓	✓	✓	✓	✓	✓
PCLBUZ0	✓	✓	✓	(✓)	(✓)	(✓)
PCLBUZ1	✓	✓	✓	(✓)	(✓)	(✓)
REGC	✓	✓	✓	✓	✓	✓
REMOOUT	✓	✓	✓	(✓)	(✓)	(✓)
RTC1HZ	✓	✓	✓	(✓)	(✓)	(✓)
RESET	✓	✓	✓	✓	✓	✓
RxD0	✓	✓	✓	✓	✓	✓
RxD1	✓	✓	✓	✓	✓	✓
RxD2	✓	✓	✓	✓	✓	✓
RxD3	✓	✓	✓	✓	—	—
TxD0	✓	✓	✓	✓	✓	✓
TxD1	✓	✓	✓	✓	✓	✓
TxD2	✓	✓	✓	✓	✓	✓
TxD3	✓	✓	✓	✓	—	—
RxDA0	✓	✓	✓	✓	✓	✓
RxDA1	✓	✓	✓	✓	✓	✓
RxDA2	✓	✓	✓Note	✓Note	✓Note	—
RxDA3	✓	✓	—	—	—	—
TxDA0	✓	✓	✓	✓	✓	✓
TxDA1	✓	✓	✓	✓	✓	✓
TxDA2	✓	✓	✓Note	✓Note	✓Note	—
TxDA3	✓	✓	—	—	—	—
CLKA0	✓	✓	✓	✓	✓	✓
CLKA1	✓	✓	—	—	—	—
SCK00	✓	✓	✓	✓	✓	✓
SCK01	✓	✓	✓	(✓)	(✓)	(✓)
SCK10	✓	✓	✓	✓	✓	✓
SCK11	✓	✓	✓	✓	✓	✓
SCK20	✓	✓	✓	✓	✓	—
SCK21	✓	✓	✓	✓	✓	✓
SCK30	✓	✓	✓	✓	—	—
SCK31	✓	(✓)	(✓)	—	—	—
SCL00	✓	✓	✓	✓	✓	✓
SCL01	✓	✓	✓	✓	(✓)	(✓)

(4/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
SCL10	✓	✓	✓	✓	✓	✓
SCL11	✓	✓	✓	✓	✓	✓
SCL20	✓	✓	✓	✓	✓	—
SCL21	✓	✓	✓	✓	✓	✓
SCL30	✓	✓	✓	✓	✓	(✓)
SCL31	✓	(✓)	(✓)	—	—	—
SDA00	✓	✓	✓	✓	✓	✓
SDA01	✓	✓	✓	✓	(✓)	(✓)
SDA10	✓	✓	✓	✓	✓	✓
SDA11	✓	✓	✓	✓	✓	✓
SDA20	✓	✓	✓	✓	✓	—
SDA21	✓	✓	✓	✓	✓	✓
SDA30	✓	✓	✓	✓	✓	(✓)
SDA31	✓	(✓)	(✓)	—	—	—
SI00	✓	✓	✓	✓	✓	✓
SI01	✓	✓	✓	(✓)	(✓)	(✓)
SI10	✓	✓	✓	✓	✓	✓
SI11	✓	✓	✓	✓	✓	✓
SI20	✓	✓	✓	✓	✓	—
SI21	✓	✓	✓	✓	✓	✓
SI30	✓	✓	✓	✓	—	—
SI31	✓	(✓)	(✓)	—	—	—
SO00	✓	✓	✓	✓	✓	✓
SO01	✓	✓	✓	(✓)	(✓)	(✓)
SO10	✓	✓	✓	✓	✓	✓
SO11	✓	✓	✓	✓	✓	✓
SO20	✓	✓	✓	✓	✓	—
SO21	✓	✓	✓	✓	✓	✓
SO30	✓	✓	✓	✓	—	—
SO31	✓	(✓)	(✓)	—	—	—
SCLA0	✓	✓	✓	✓	✓	✓
SCLA1	✓	✓	✓	(✓)	(✓)	(✓)
SDAA0	✓	✓	✓	✓	✓	✓
SDAA1	✓	✓	✓	(✓)	(✓)	(✓)
TI00	✓	✓	✓	(✓)	(✓)	(✓)
TI01	✓	(✓)	(✓)	(✓)	(✓)	(✓)
TI02	✓	✓	✓	✓	✓	(✓)
TI03	✓	✓	✓	(✓)	(✓)	(✓)
TI04	✓	✓	✓	✓	(✓)	(✓)
TI05	✓	✓	✓	(✓)	(✓)	(✓)

(5/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
TI06	✓	✓	(✓)	(✓)	(✓)	(✓)
TI07	✓	✓	✓	(✓)	(✓)	(✓)
TO00	✓	✓	✓	(✓)	(✓)	(✓)
TO01	✓	(✓)	(✓)	(✓)	(✓)	(✓)
TO02	✓	✓	✓	✓	✓	(✓)
TO03	✓	✓	✓	(✓)	(✓)	(✓)
TO04	✓	✓	✓	✓	(✓)	(✓)
TO05	✓	✓	✓	(✓)	(✓)	(✓)
TO06	✓	✓	(✓)	(✓)	(✓)	(✓)
TO07	✓	✓	✓	(✓)	(✓)	(✓)
EI00	✓	✓	✓	✓	✓	✓
EI06	✓	✓	✓	✓	✓	✓
EI07	✓	✓	✓	✓	✓	✓
EI12	✓	✓	✓	✓	✓	✓
EI16	✓	✓	✓	✓	✓	✓
EI17	✓	✓	✓	✓	✓	✓
EI20	✓	✓	✓	✓	✓	✓
EI21	✓	✓	✓	✓	✓	✓
EI26	✓	✓	✓	✓	✓	✓
EI27	✓	✓	✓	✓	✓	✓
EI76	✓	✓	✓	✓	✓	✓
EI77	✓	✓	✓	✓	✓	✓
EI121	✓	✓	✓	✓	✓	✓
EI122	✓	✓	✓	✓	✓	✓
EI137	✓	✓	✓	✓	✓	✓
EO00	✓	✓	✓	✓	✓	✓
EO05	✓	✓	✓	✓	✓	✓
EO17	✓	✓	✓	✓	✓	✓
EO60	✓	✓	✓	✓	✓	✓
EO61	✓	✓	✓	✓	✓	✓
EO76	✓	✓	✓	✓	✓	✓
EO77	✓	✓	✓	✓	✓	✓
TRJIO0	✓	✓	✓	(✓)	(✓)	(✓)
TRJIO1	✓	✓	(✓)Note	(✓)Note	(✓)Note	(✓)Note
TRJO0	✓	✓	✓	(✓)	(✓)	(✓)
TRJO1	✓	✓	(✓)Note	(✓)Note	(✓)Note	(✓)Note
TKBO00	✓	✓	✓	✓	✓	✓
TKBO01	✓	✓	✓	✓	✓	✓
TKBO10	✓	✓	(✓)Note	(✓)Note	(✓)Note	(✓)Note
TKBO11	✓	✓	✓ Note	✓ Note	(✓)Note	(✓)Note

(6/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
TKBO20	✓	✓	✓Note	(✓)Note	(✓)Note	(✓)Note
TKBO21	✓	✓	(✓)Note	(✓)Note	(✓)Note	(✓)Note
VL1	✓	✓	✓	✓	✓	✓
VL2	✓	✓	✓	✓	✓	✓
VL3	✓	✓	✓	✓	✓	✓
VL4	✓	✓	✓	✓	✓	✓
CAPH	✓	✓	✓	✓	✓	✓
CAPL	✓	✓	✓	✓	✓	✓
X1	✓	✓	✓	✓	✓	✓
X2	✓	✓	✓	✓	✓	✓
EXCLK	✓	✓	✓	✓	✓	✓
XT1	✓	✓	✓	✓	✓	✓
XT2	✓	✓	✓	✓	✓	✓
EXCLKS	✓	✓	✓	✓	✓	✓
VDD	✓	✓	✓	✓	✓	✓
VBAT0	✓	✓	✓	✓	✓	✓
VBAT1	✓	✓	✓	✓	✓	✓
AVREFP	✓	✓	✓	✓	✓	✓
AVREFM	✓	✓	✓	✓	✓	✓
Vss	✓	✓	✓	✓	✓	✓
EVss	✓	—	—	—	—	—
TOOLRxD	✓	✓	✓	✓	✓	✓
TOOLTxD	✓	✓	✓	✓	✓	✓
TOOL0	✓	✓	✓	✓	✓	✓
COM0	✓	✓	✓	✓	✓	✓
COM1	✓	✓	✓	✓	✓	✓
COM2	✓	✓	✓	✓	✓	✓
COM3	✓	✓	✓	✓	✓	✓
COM4	✓	✓	✓	✓	✓	✓
COM5	✓	✓	✓	✓	✓	✓
COM6	✓	✓	✓	✓	✓	✓
COM7	✓	✓	✓	✓	✓	✓
SEG0	✓	✓	✓	✓	✓	✓
SEG1	✓	✓	✓	✓	✓	✓
SEG2	✓	✓	✓	✓	✓	✓
SEG3	✓	✓	✓	✓	✓	✓
SEG4	✓	✓	—	—	—	—
SEG5	✓	✓	—	—	—	—
SEG6	✓	✓	✓	—	—	—
SEG7	✓	✓	✓	✓	—	—

(7/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
SEG8	✓	✓	✓	✓	✓	—
SEG9	✓	✓	—	—	—	—
SEG10	✓	✓	—	—	—	—
SEG11	✓	✓	✓	✓	✓	✓
SEG12	✓	✓	✓	—	—	—
SEG13	✓	✓	—	—	—	—
SEG14	✓	✓	—	—	—	—
SEG15	✓	✓	—	—	—	—
SEG16	✓	✓	✓	✓	—	—
SEG17	✓	✓	✓	✓	✓	—
SEG18	✓	✓	✓	✓	✓	✓
SEG19	✓	✓	✓	✓	✓	✓
SEG20	✓	✓	✓	—	—	—
SEG21	✓	✓	✓	—	—	—
SEG22	✓	—	—	—	—	—
SEG23	✓	—	—	—	—	—
SEG24	✓	✓	—	—	—	—
SEG25	✓	✓	—	—	—	—
SEG26	✓	✓	—	—	—	—
SEG27	✓	✓	—	—	—	—
SEG28	✓	✓	—	—	—	—
SEG29	✓	✓	✓	✓	✓	✓
SEG30	✓	✓	—	—	—	—
SEG31	✓	✓	—	—	—	—
SEG32	✓	✓	—	—	—	—
SEG33	✓	✓	✓	✓	✓	✓
SEG34	✓	✓	✓	✓	✓	✓
SEG35	✓	✓	✓	✓	✓	✓
SEG36	✓	✓	✓	✓	✓	✓
SEG37	✓	✓	✓	✓	✓	✓
SEG38	✓	✓	✓	✓	✓	—
SEG39	✓	✓	✓	✓	—	—
SEG40	✓	✓	✓	—	—	—
SEG41	✓	✓	✓	✓	✓	✓
SEG42	✓	✓	✓	✓	✓	✓
SEG43	✓	✓	✓	✓	✓	✓
SEG44	✓	✓	✓	—	—	—
SEG45	✓	✓	✓	—	—	—
SEG46	✓	✓	✓	—	—	—
SEG47	✓	✓	✓	—	—	—

(8/8)

Function Name	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
SEG48	✓	✓	✓	✓	✓	✓
SEG49	✓	✓	✓	✓	✓	✓
SEG50	✓	✓	✓	✓	✓	✓
SEG51	✓	—	—	—	—	—
SEG52	✓	—	—	—	—	—
SEG53	✓	—	—	—	—	—
SEG54	✓	—	—	—	—	—
SEG55	✓	—	—	—	—	—
EXSDI0	✓	✓	✓	✓	✓	✓
EXSDI1	✓	✓	✓	✓	✓	✓
EXSDO0	✓	✓	✓	✓	✓	(✓)
EXSDO1	✓	✓	✓	✓	✓	✓

Note Not present in products with 128 or fewer Kbytes of code flash memory.

Remark Parentheses indicate that the pin can be assigned via settings in the peripheral I/O redirection registers (PIORx).

2.2.2 Description of pin functions

(1/2)

Function Name	I/O	Function
ANI0, ANI1, ANI16 to ANI26	Input	Analog inputs for the A/D converter (see Figure 16 - 43 Connections of VDD, AVREFP, and Analog Input Pins)
ANO0, ANO1	Output	D/A converter outputs
IVCMP0, IVCMP1	Input	Analog voltage inputs for the comparator
IVREF0, IVREF1	Input	Reference voltage inputs for the comparator
VCOUT0, VCOUT1	Output	Comparator outputs
TS00 to TS35	I/O	Electrostatic capacitance measurement pins (touch sensor)
TSCAP	—	Pin for connecting a power supply capacitor for use in measuring electrostatic capacitance. 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.
CCD00 to CCD07	Output	Controlled current drive port pins
INTP0 to INTP7	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 KR7	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.
REMOOUT	Output	Pin for remote-control output
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 RxD3	Input	Serial data input pins for serial interfaces UART0, UART1, UART2, and UART3
TxD0 to TxD3	Output	Serial data output pins for serial interfaces UART0, UART1, UART2, and UART3
RxDA0 to RxD3	Input	Serial data input pins for serial interfaces UARTA0, UARTA1, UARTA2, and UARTA3
TxDA0 to TxD3	Output	Serial data output pins for serial interfaces UARTA0, UARTA1, UARTA2, and UARTA3
CLKA0, CLKA1	Output	Clock output pins for serial interfaces UARTA0 and UARTA1
SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, SCK31	I/O	Serial clock I/O pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCL30, SCL31	Output	Serial clock output pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, and IIC31
SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, SDA31	I/O	Serial data I/O pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, and IIC31
SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31	Input	Serial data input pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31	Output	Serial data output pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SCLA0, SCLA1	I/O	Clock I/O pins for serial interfaces IICA0 and IICA1
SDAA0, SDAA1	I/O	Serial data I/O pins for serial interfaces IICA0 and IICA1
EXSDI0, EXSDI1	Input	External signal sampler phase detection inputs
EXSDO0, EXSDO1	Output	External signal sampler clock outputs
TI00 to TI07	Input	Pins for inputting an external counting clock/capture trigger to 16-bit timers 00 to 03
TO00 to TO07	Output	Timer output pins for 16-bit timers 00 to 03

(2/2)

Function Name	I/O	Function
EI00, EI06, EI07, EI12, EI16, EI17, EI20, EI21, EI26, EI27, EI76, EI77, EI121, EI122, EI137	Input	ELCL inputs
EO00, EO05, EO17, EO60, EO61, EO76, EO77	Output	ELCL outputs
TRJIO0, TRJIO1	I/O	Timer RJ I/O
TRJO0, TRJO1	Output	Timer RJ outputs
TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21	Output	Output pins for 16-bit timers KB40, KB41, and KB42
VL1 to VL4	—	LCD drive voltage
CAPH, CAPL	—	Pins for connecting a capacitor for the LCD controller/driver
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 for pins
VBAT0, VBAT1	—	Power supply pin for battery backup
AVREFP	Input	Positive reference voltage input for the A/D converter
AVREFM	Input	Negative reference voltage input for the A/D converter
VSS	—	Ground voltage for pins
EVSS	—	Ground voltage for port pins
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming
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
COM0 to COM7	Output	LCD controller/driver common signal outputs
SEG0 to SEG55	Output	LCD controller/driver segment signal outputs

Caution The relationship between the voltage on P40/TOOL0 and the operating mode after release from the reset state is as follows. For details, see 38.4 Programming Method.

P40/TOOL0	Operating Mode
VDD	Normal operating mode
0 V	Flash memory programming mode

Remark Use bypass capacitors (about 0.1 μ F) as noise and latch-up countermeasures with relatively thick wires at the shortest distance to VDD to VSS and EVSS lines.

2.2.3 VBATn pins (n = 0, 1)

2.2.3.1 Functions of the VBATn pins (n = 0, 1)

The VBATn pins are used to connect the battery for use in backing up. Connecting the VBATn pin to a battery for use in backing up enables the supply of power from the VBATn pin when the power supply to the VDD pin is shut off. The main purpose of the VBATn pins is to continue operation of the realtime clock (RTC).

2.2.3.2 Connecting the VBATn pin to the battery for use in backing up (n = 0, 1)

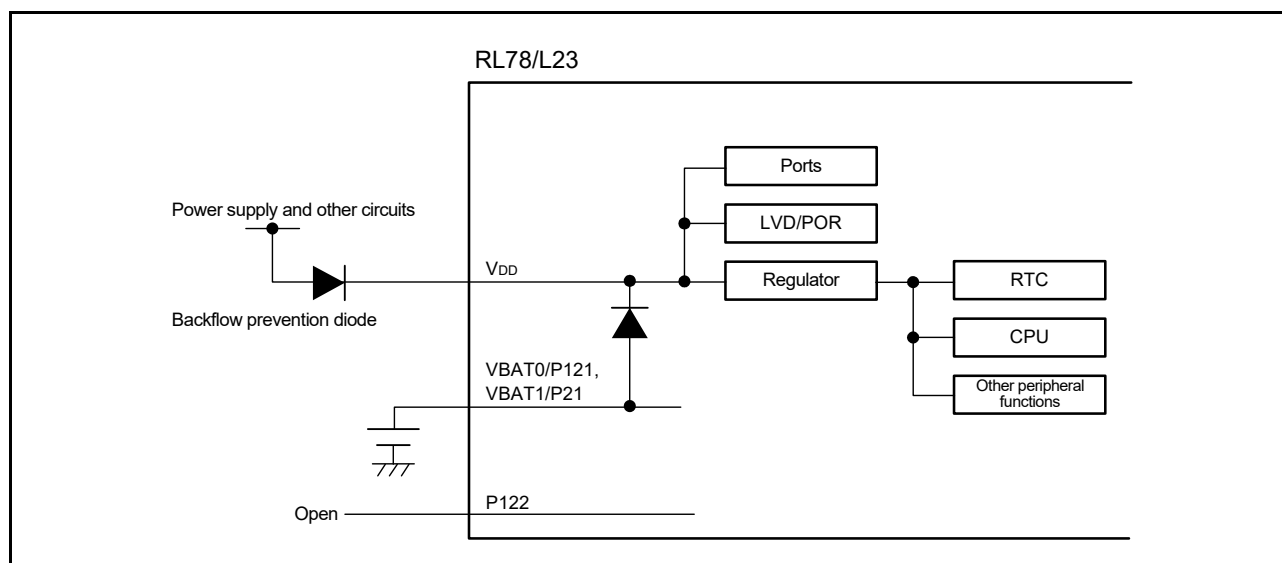
Figure 2 - 1 shows an example of the connection of the VBATn pin.

The VBATn pin supplies power to the VDD pin through an internal diode. The internal diode for use with the VBATn pin is always connected to the VDD pin. If preventing the backward flow of current, that is, current flowing from the VBATn pin to the power supply or other circuits that are connected to the VDD pin through the diode, is required, externally connect a backflow prevention diode to the VDD pin.

The allowed range of input voltage on the VBATn pin is 2.7 to 5.5 V. The input voltage on the VBATn pin falling below 2.7 V while power is being supplied from the VBATn pin may lead to the generation of a POR reset due to the fall in the voltage across the diode. The maximum current that can be supplied through the VBATn pin is 150 μ A.

- <R> When VBAT0/P121 is to be used, connect the backup battery to VBAT0 and leave P122 open-circuit.
When VBAT1/P21 is to be used, the backup battery can be connected to VBAT1 and used with the single pin.

<R> Figure 2 - 1 Example of the Connection of the VBATn Pin (n = 0, 1)



2.2.3.3 Using the VBATn Pin (n = 0, 1)

The following describes how to make the initial settings for the VBATn pin and an example of the procedure for switching the power supply pin to the VBATn pin. This processing is to be completed before the voltage on the VDD pin falls below that supplied from the VBATn pin.

In addition, **Figure 2 - 2** shows the state transitions in switching the power supply pin between the VDD and VBATn pins.

1. Making the initial settings for the VBATn pins

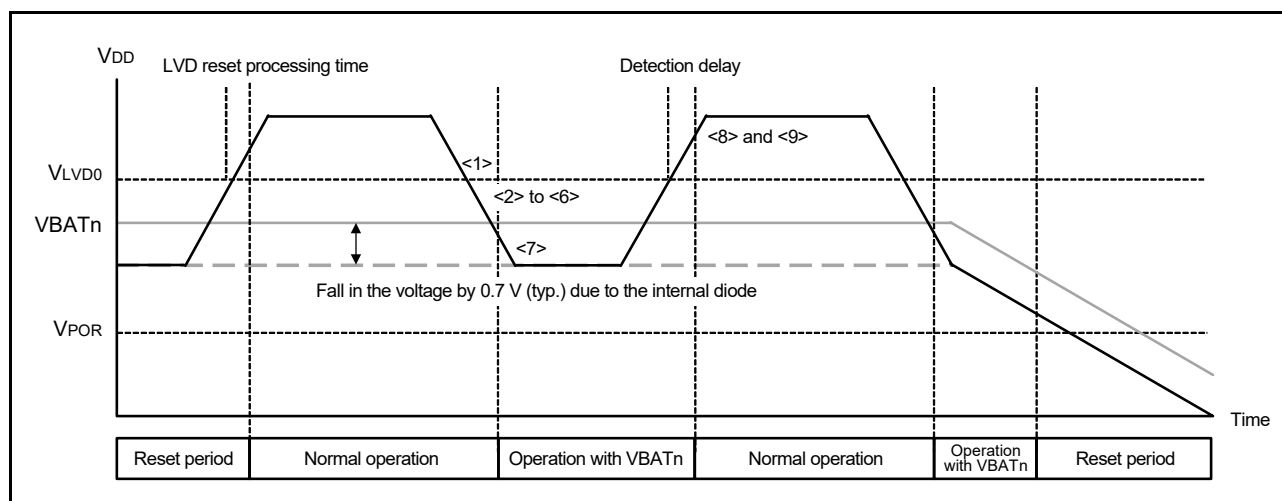
For the VBAT0 pin: Set the P121 pin to X1 oscillation mode (by setting the EXCLK and OSCSEL bits of the CMC register to 0 and 1, respectively, and the MSTOP bit of the CSC register to 1) in the initial settings.

For the VBAT1 pin: Set the PMCA21 bit of the PMCA2 register to 1 as the initial setting for the P21 pin.

2. Example of the procedure for switching the power supply pin to the VBATn pin

- <1> Use LVD0 in the interrupt mode to generate an interrupt request when the power supply voltage (VDD) falls.
- The switching process starts in response to this interrupt request signal.
- <2> Disable interrupts other than LVD0.
- <3> Stop the operation of all peripheral functions other than the realtime clock (RTC).
- <4> Change the output settings for port pins so that no current flows through them.
- <5> Clear the interrupt request flag of LVD0.
- <6> After having confirmed that the value of the LVD0F bit is 1 ($VDD < \text{detection voltage}$), if the CPU is operating with the main system clock, place this LSI chip in the STOP mode. If the CPU is operating with the subsystem clock, place the chip in the HALT mode.
- <7> Keep the chip in the above state until LVD0 generates an interrupt request.
- <8> Supplying the power supply voltage (VDD) back to the chip makes LVD0 generate an interrupt request, thereby releasing the chip from the STOP or HALT mode.
- <9> After having confirmed that the value of the LVD0F bit is 0 ($VDD \geq \text{detection voltage}$), make the settings to return the peripheral functions to normal operation while the power supply voltage (VDD) is being supplied.

Figure 2 - 2 State Transitions in Switching the Power Supply Pin between the VDD and VBATn Pins (n = 0, 1)



When switching between the VDD and VBATn pins is too frequent, waiting for the fluctuations in voltage to settle between steps 4 and 5 in the procedure can prevent excessive switching between the VDD and VBATn pins.

Caution 1. Operation with the main system clock is prohibited while the battery is supplying power.

Caution 2. Make the setting to stop counting by the WDT when the battery is to supply power.

Moreover, input of the low level on the reset pin is prohibited. This is because attempting to do so leads to starting operation with the main system clock after release from the reset state, which requires current exceeding 150 μA , the maximum allowable current through the VBAT pin.

(Caution is listed on the next page.)

- Caution 3.** When the battery is to be used for the supply of power, set the P121 pin to the X1 oscillation mode and the PMCA21 bit of the PMCA2 register to 1 as the setting for the P21 pin. Do not set the P121 and P21 pins to the input or output mode.

2.3 Connection of Unused Pins

Table 2 - 1 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 - 1 Connections of Unused Pins(1/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P07	I/O	<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
P10 to P17		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
P20, P21		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P22 to P27		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
P30 to P35		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment 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 to P45		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P46, P47		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
P50 to P57		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment 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.
P64 to P67		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.

Table 2 - 1 Connections of Unused Pins(2/2)

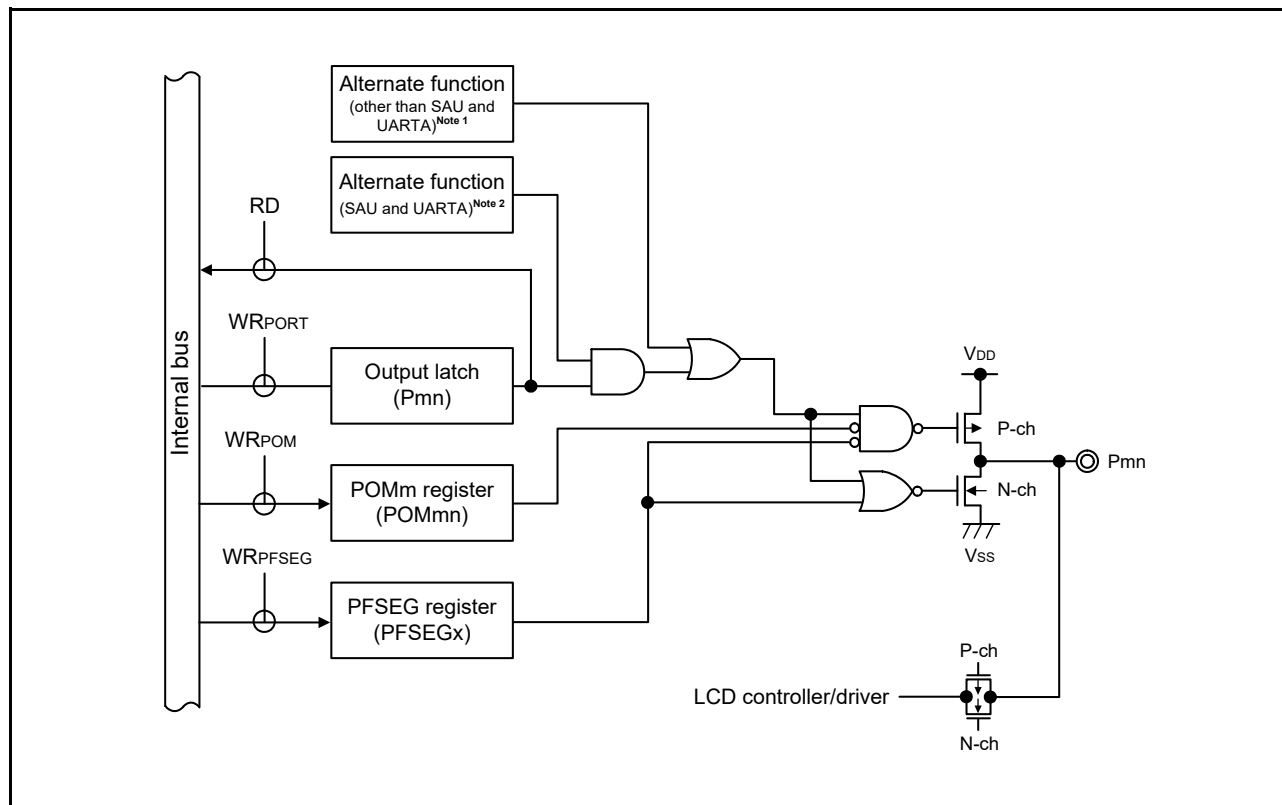
Pin Name	I/O	Recommended Connection of Unused Pins
P70 to P77	I/O	<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
P80 to P87		Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P90 to P97		<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins 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 and OSCSELS bits in the clock operation mode control register (CMC) to 0 and 1, respectively. 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.
P125 to P127	I/O	Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit.
P130	Output	<When setting for port I/O> Leave the pin open-circuit. <When setting for segment output> Leave the pin open-circuit.
P137	Input	Set the PDIDIS137 bit in the 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	I/O	Input: Independently connect the pin to VDD or VSS via a resistor. Output: Leave the pin open-circuit.
P141 to P145	I/O	<When setting for port I/O> Input: Independently connect the pins to VDD or VSS via resistors. Output: Leave the pins open-circuit. <When setting for segment output> Leave the pins open-circuit.
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 μ F).

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 μ A.

2.4 Block Diagrams of Pins

Figures 2 - 3 to 2 - 41 show the pin block diagrams of pins described in 2.1.1 44-pin products to 2.1.6 100-pin products.

Figure 2 - 3 Pin Block Diagram for Pin Type 1-5-1



Note 1. This includes the clock output from the UARTA.

Note 2. This excludes the clock output from the UARTA.

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

Remark 2. SAU: Serial array unit

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

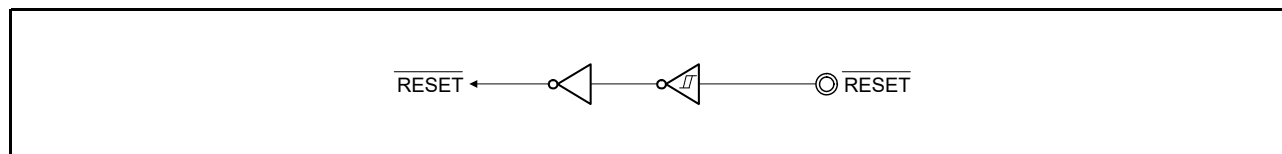
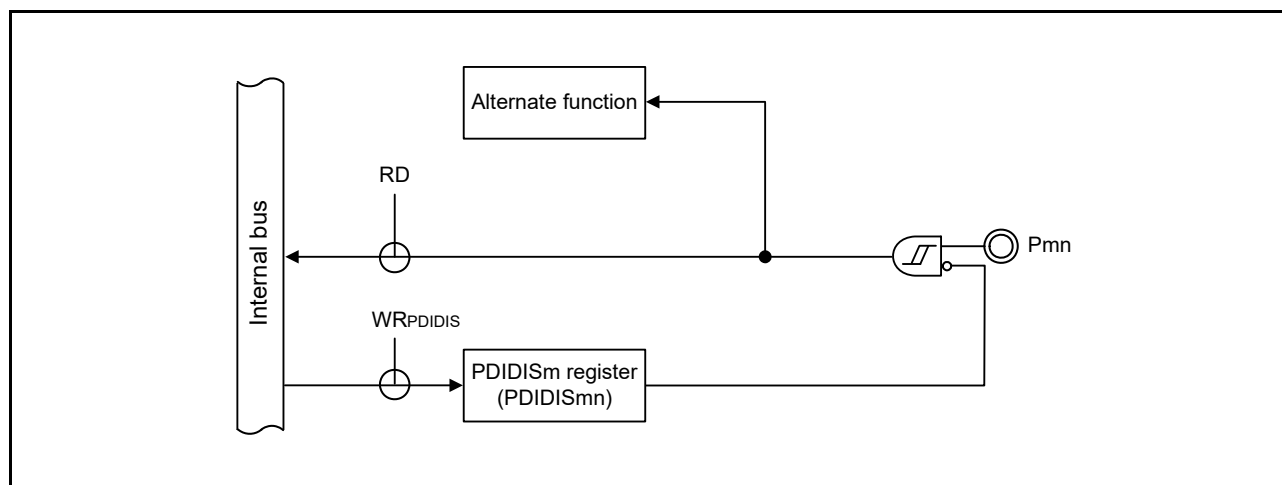
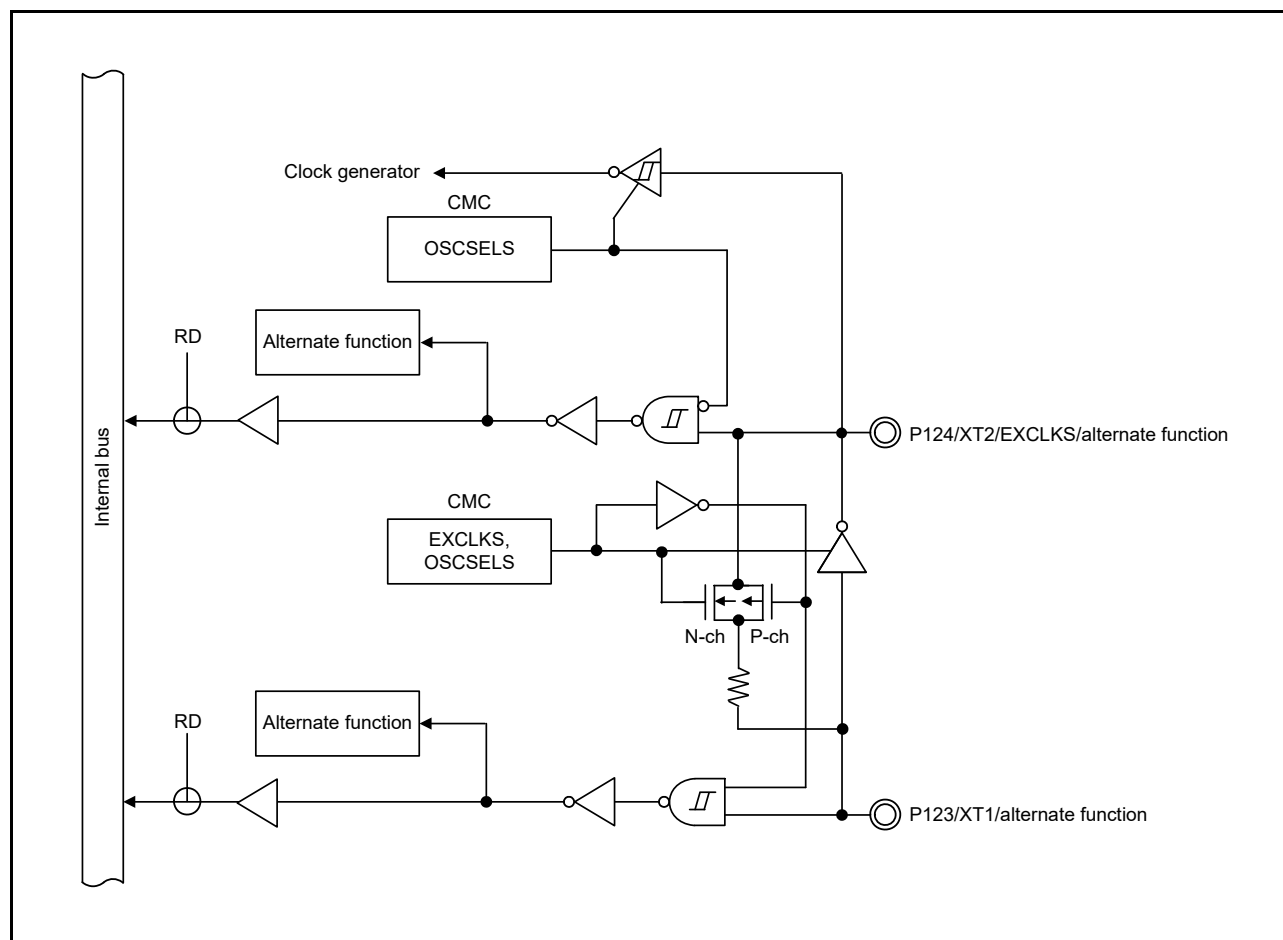


Figure 2 - 5 Pin Block Diagram for Pin Type 2-1-3



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

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



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

Figure 2 - 7 Pin Block Diagram for Pin Type 4-3-5

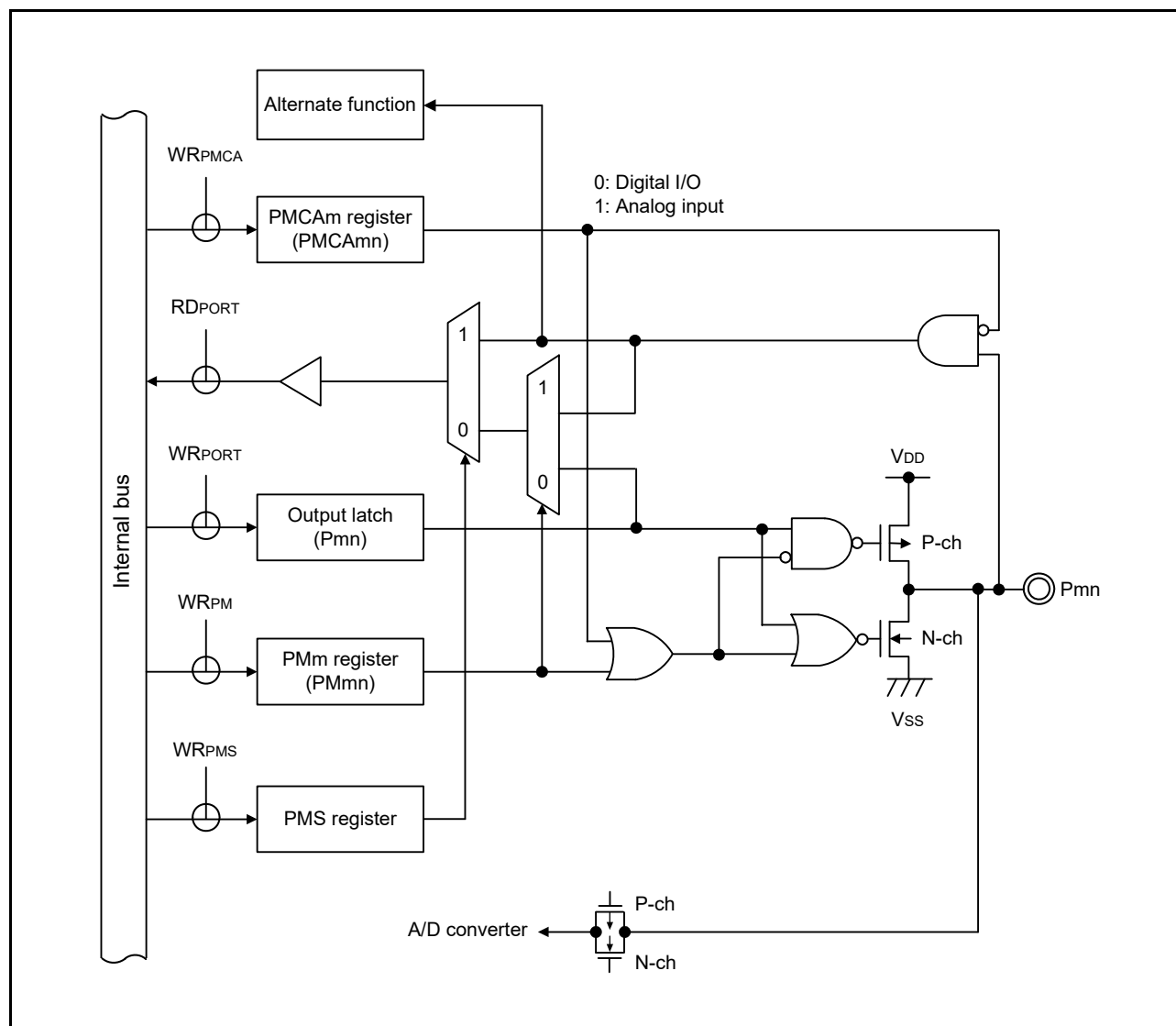
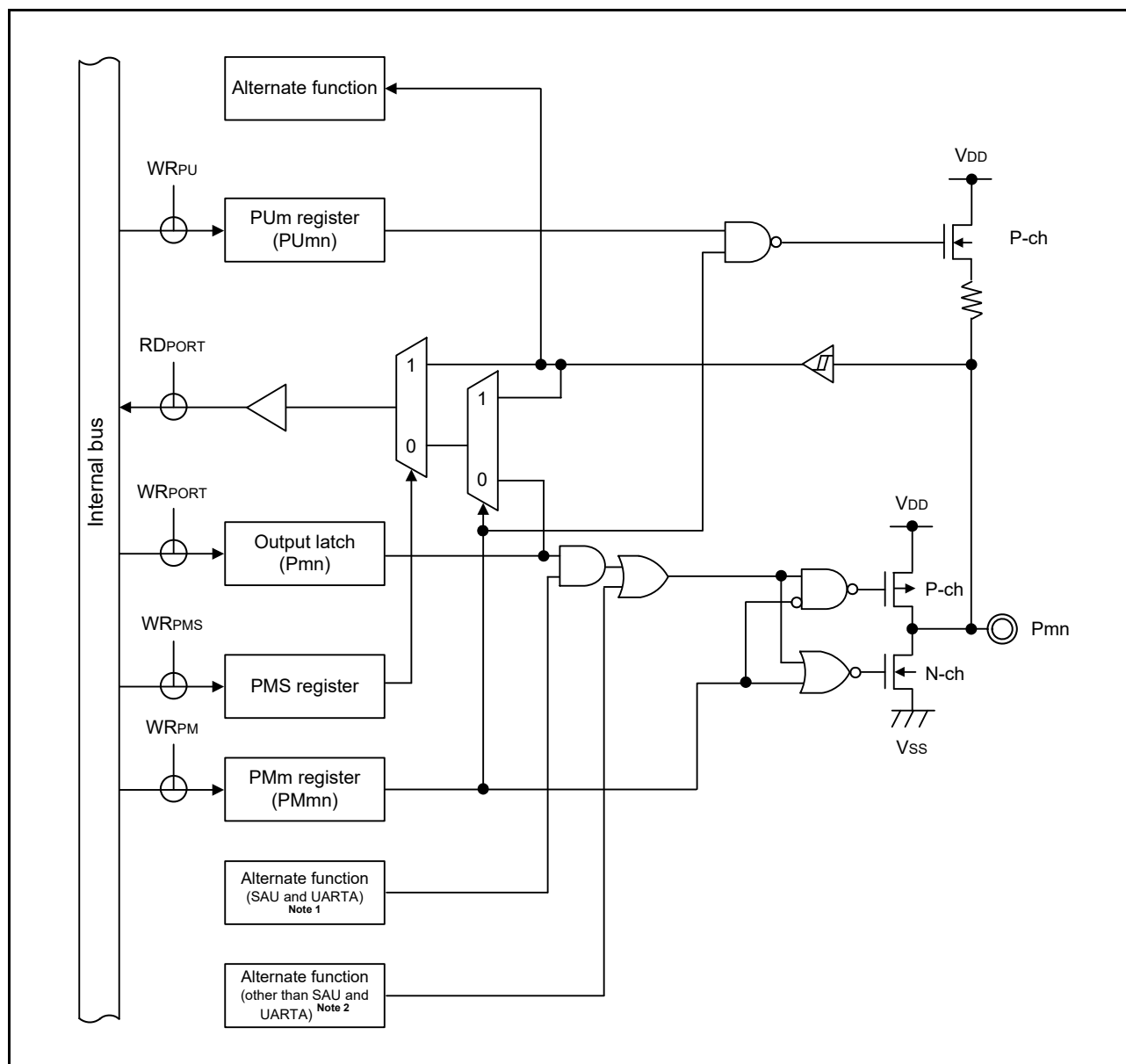


Figure 2 - 8 Pin Block Diagram for Pin Type 7-1-3



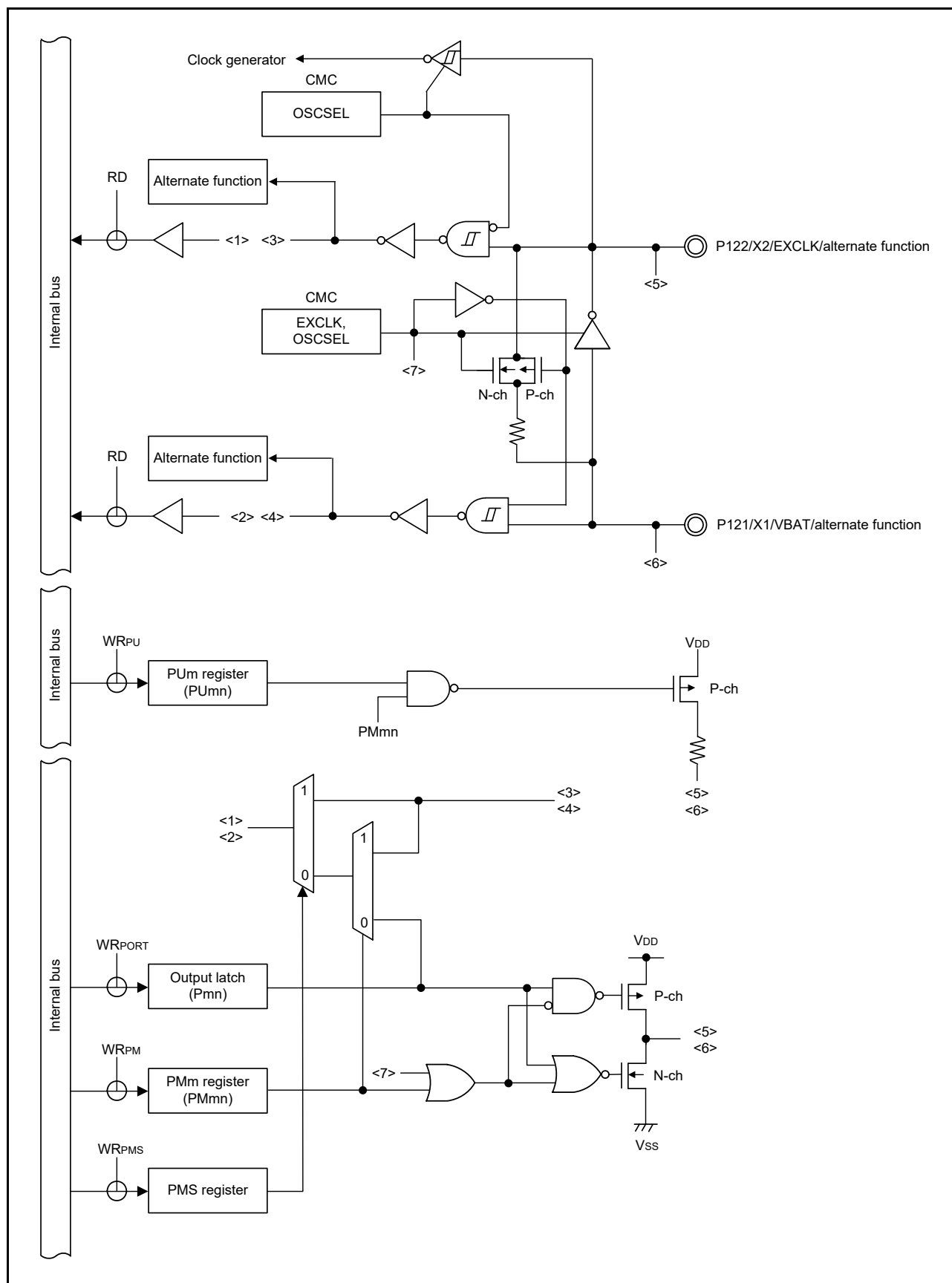
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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

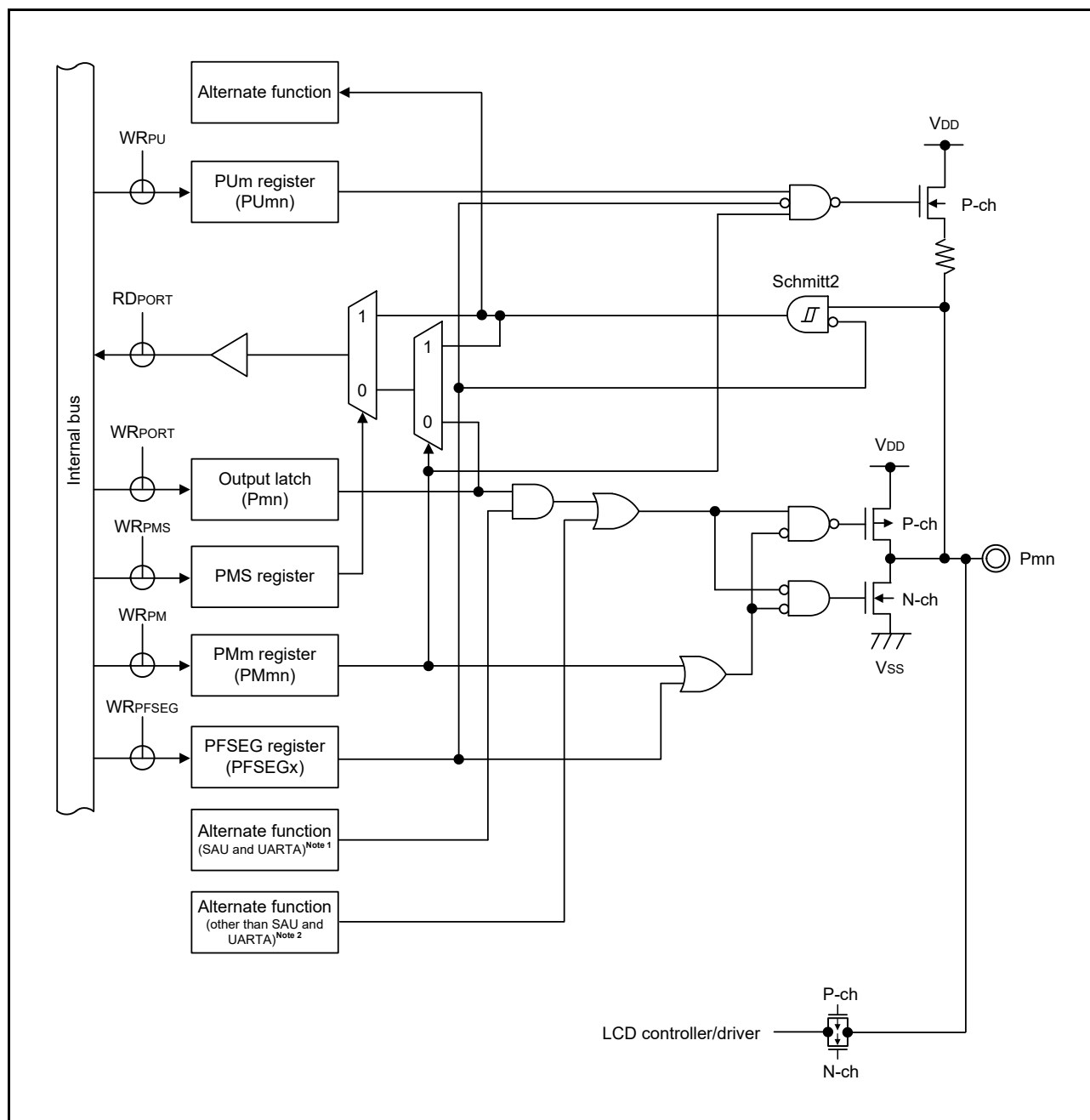
Remark 2. SAU: Serial array unit

Figure 2 - 9 Pin Block Diagram for Pin Type 7-2-1



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

Figure 2 - 10 Pin Block Diagram for Pin Type 7-5-4



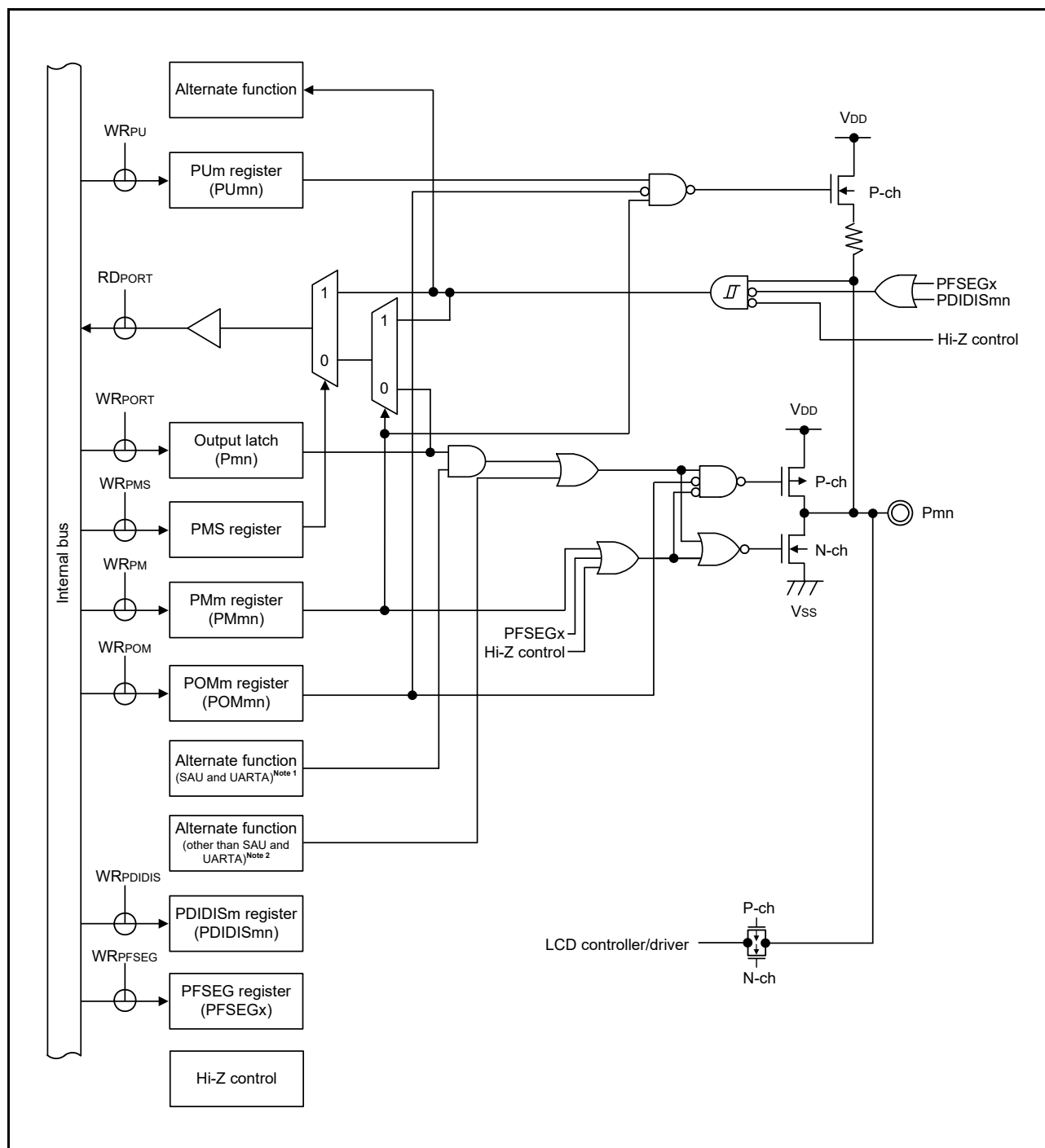
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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 7-5-30



Note 1. This excludes the clock output from the UARTA.

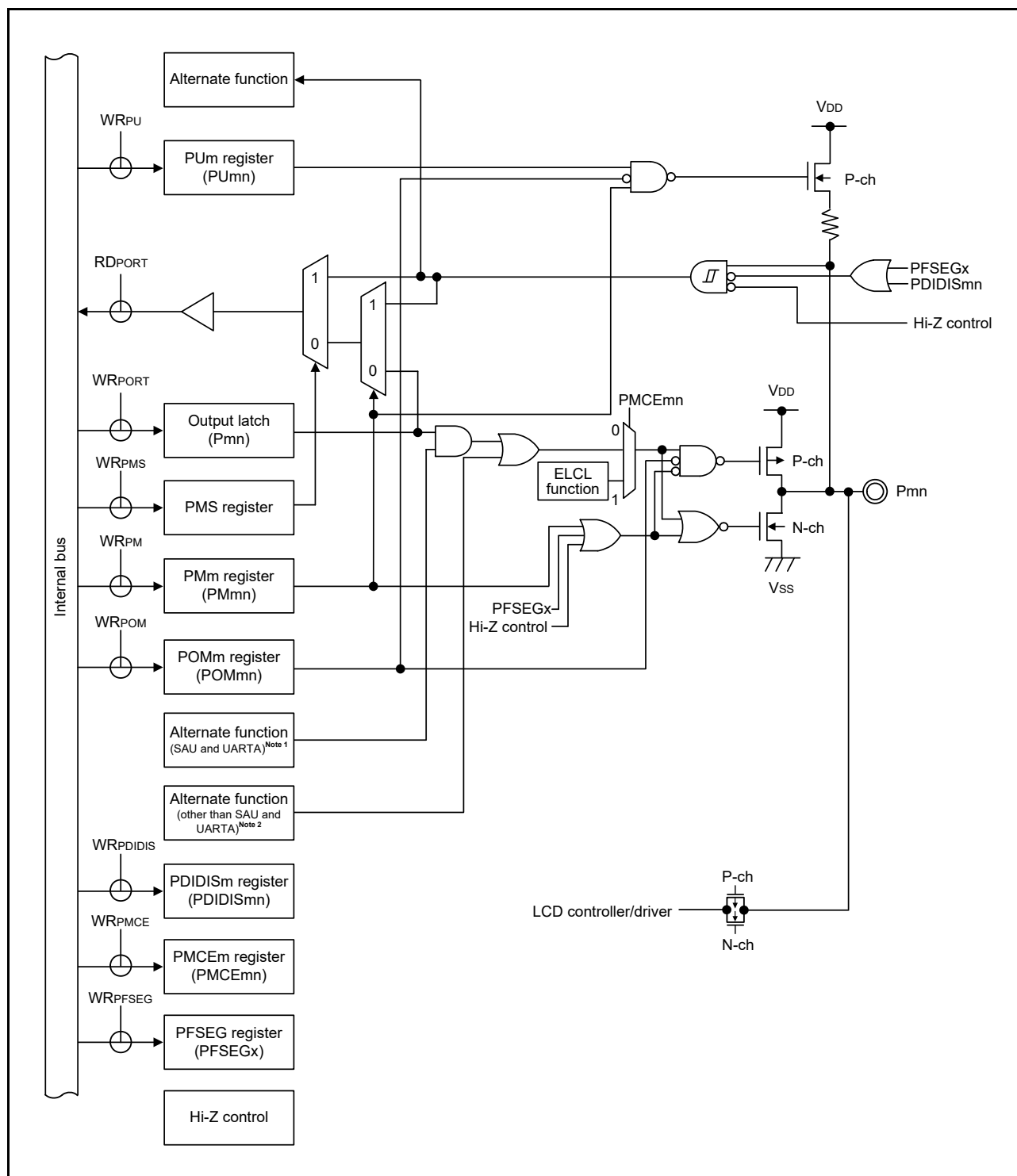
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-5-30 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 (POMm). This may lead to a through current flowing through the type 7-5-30 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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 - 12 Pin Block Diagram for Pin Type 7-5-31



Note 1. This excludes the clock output from the UARTA.

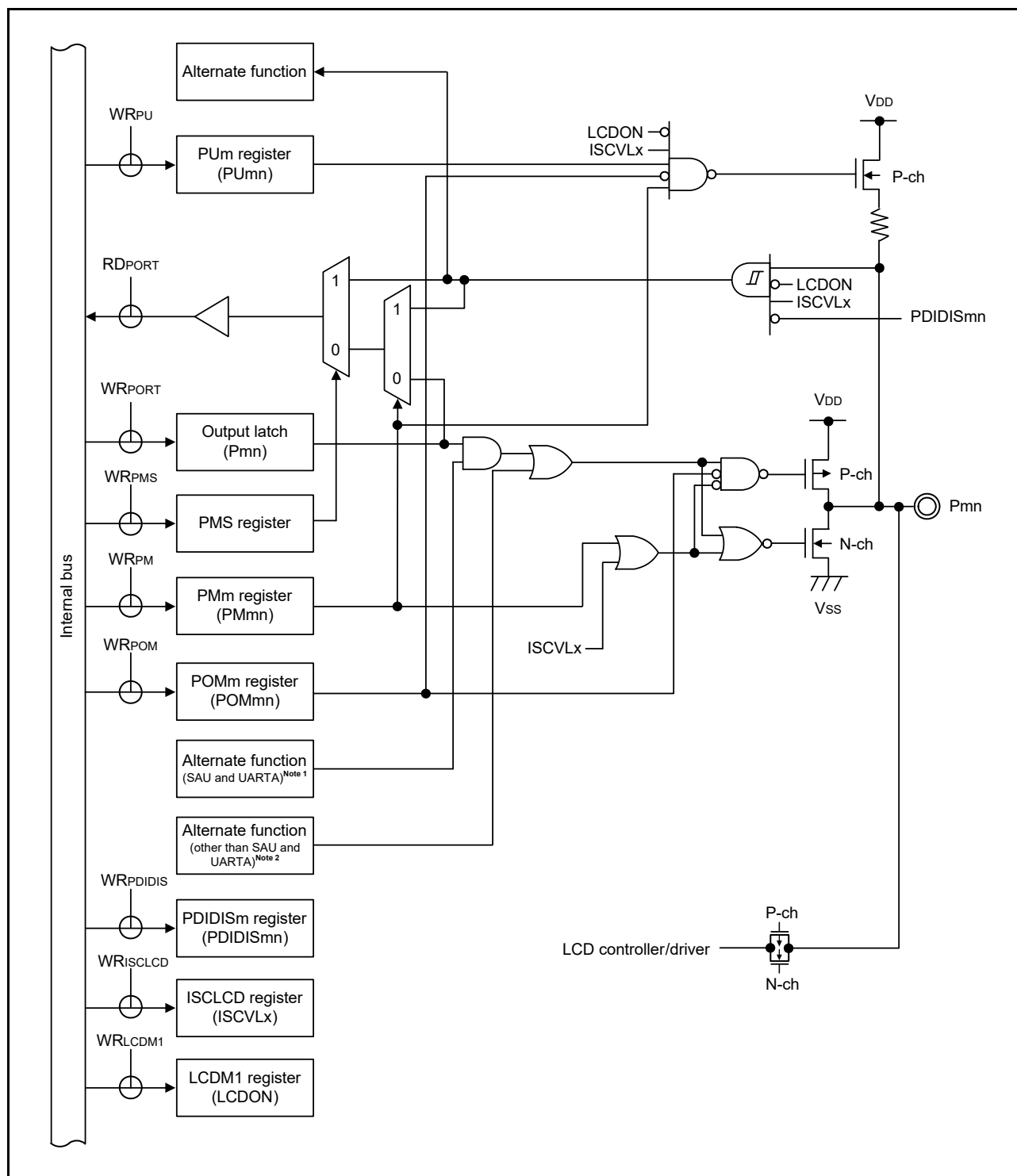
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-5-31 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 (POMm). This may lead to a through current flowing through the type 7-5-31 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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

<R> Figure 2 - 13 Pin Block Diagram for Pin Type 7-5-32



Note 1. This excludes the clock output from the UARTA.

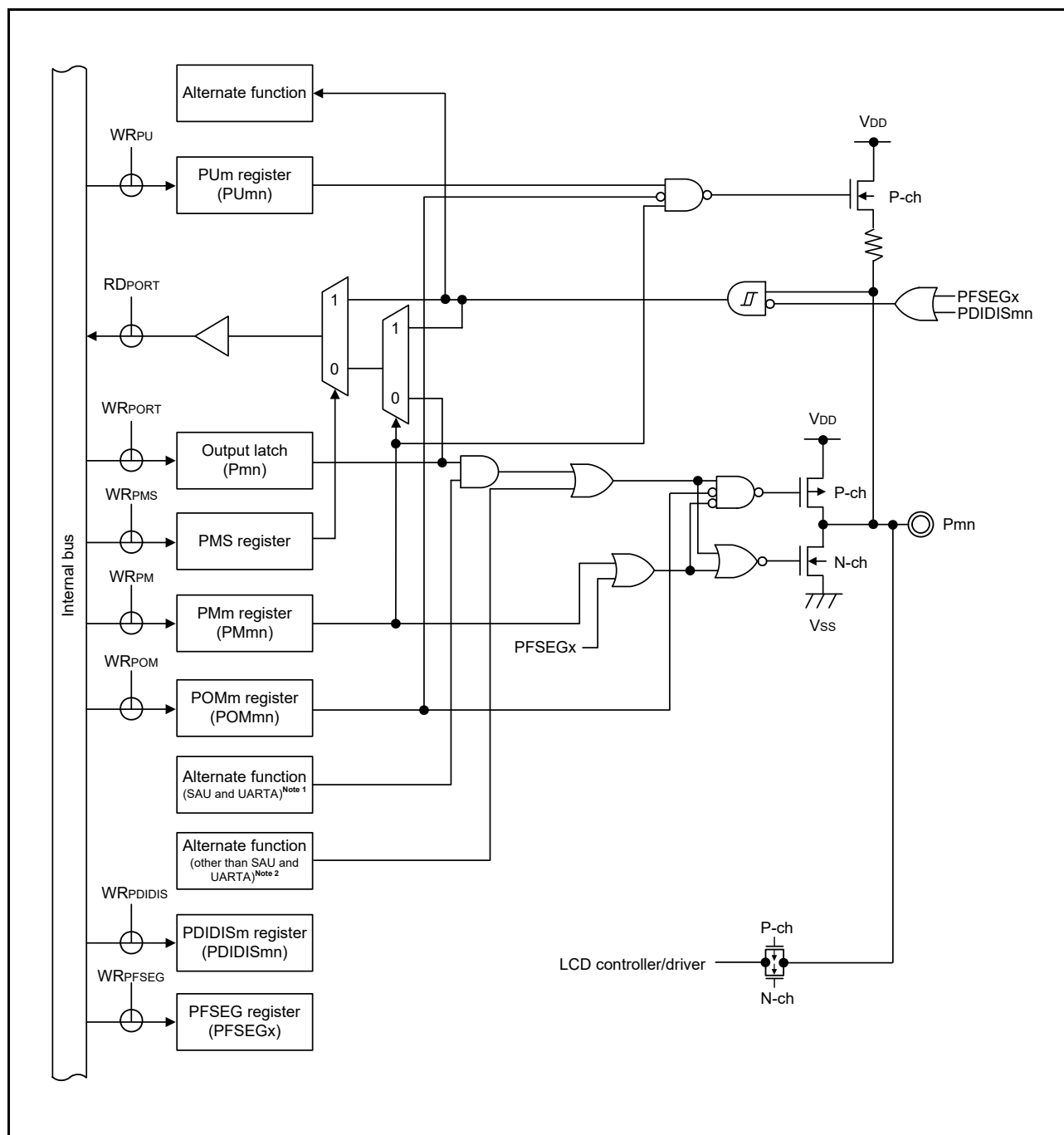
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-5-32 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 (POMm). This may lead to a through current flowing through the type 7-5-32 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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 7-5-33



Note 1. This excludes the clock output from the UARTA.

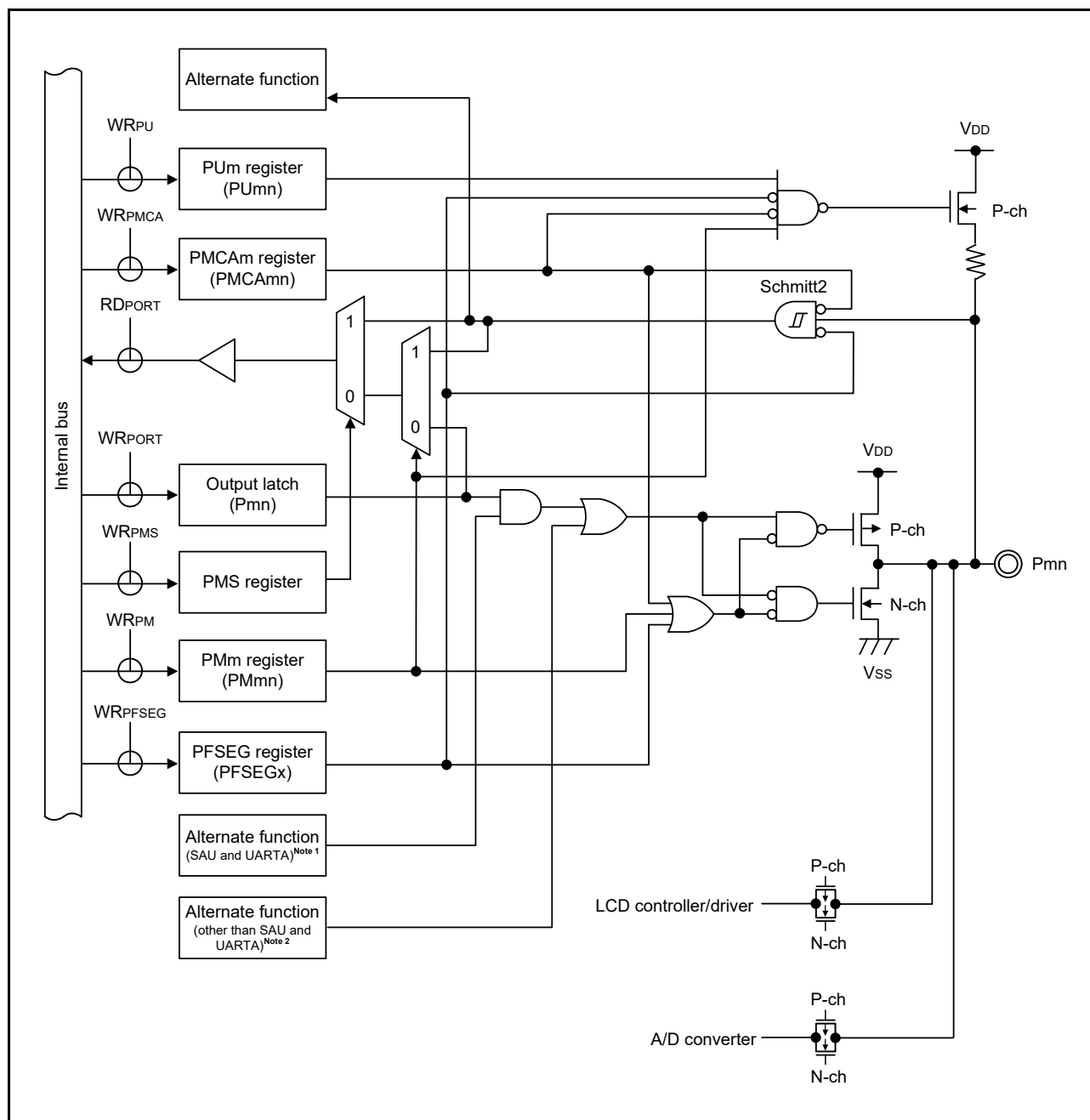
Note 2. This includes the clock output from the UARTA.

<R> Caution The input buffer is enabled even if the type 7-5-33 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 (POMm). This may lead to a through current flowing through the type 7-5-33 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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 - 15 Pin Block Diagram for Pin Type 7-10-3



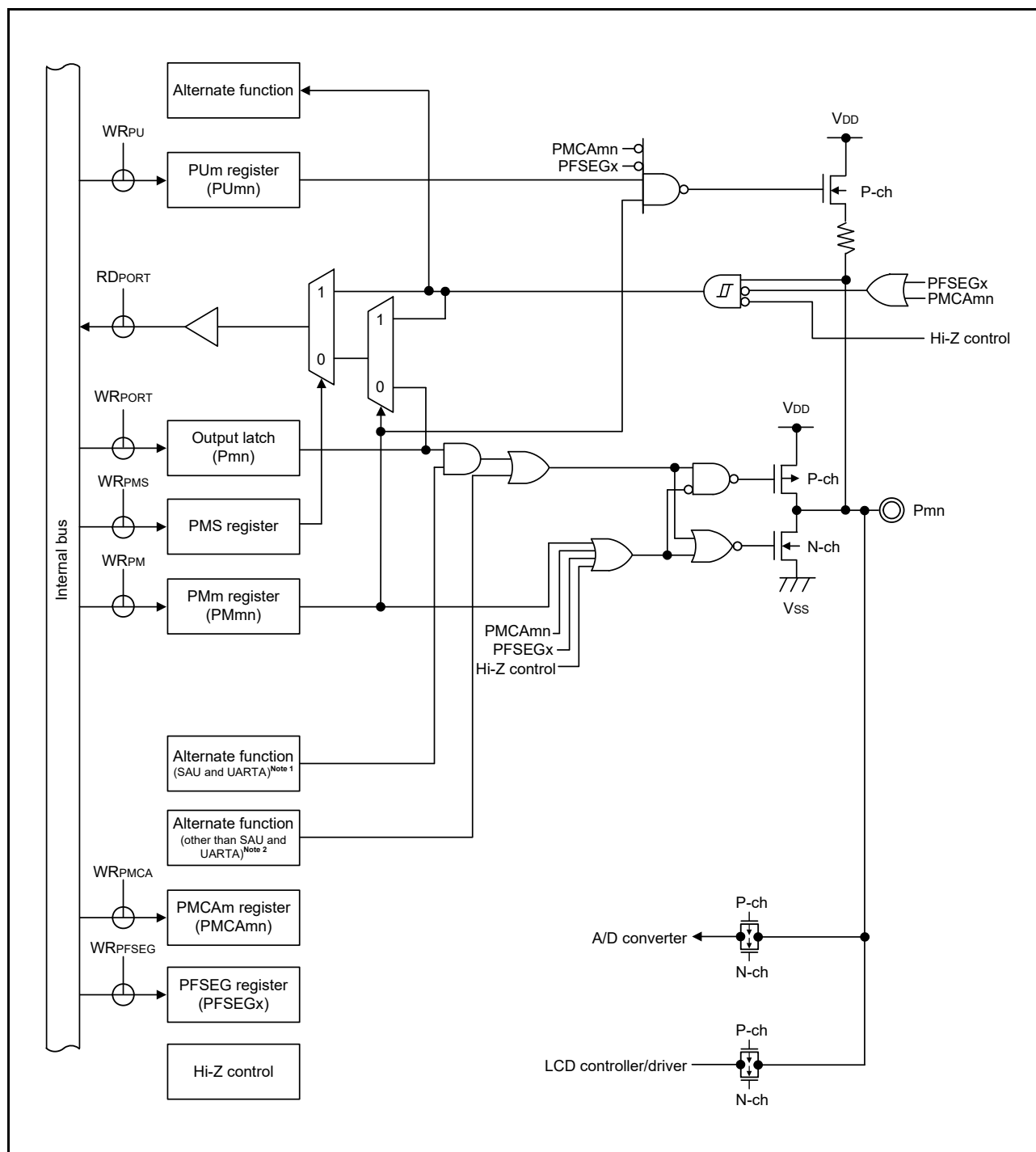
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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

Remark 2. SAU: Serial array unit

Figure 2 - 16 Pin Block Diagram for Pin Type 7-10-5



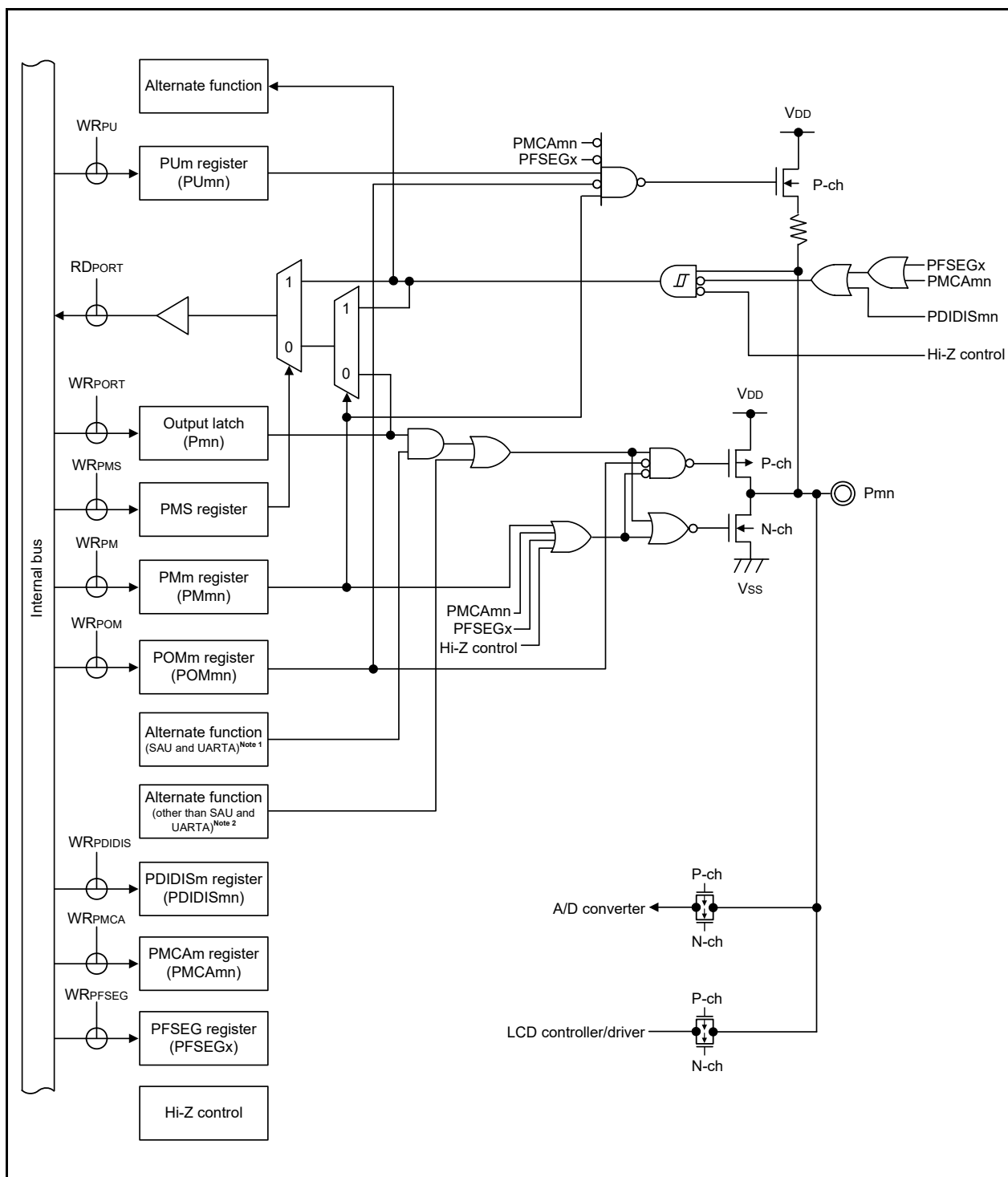
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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

Remark 2. SAU: Serial array unit

Figure 2 - 17 Pin Block Diagram for Pin Type 7-10-6



Note 1. This excludes the clock output from the UARTA.

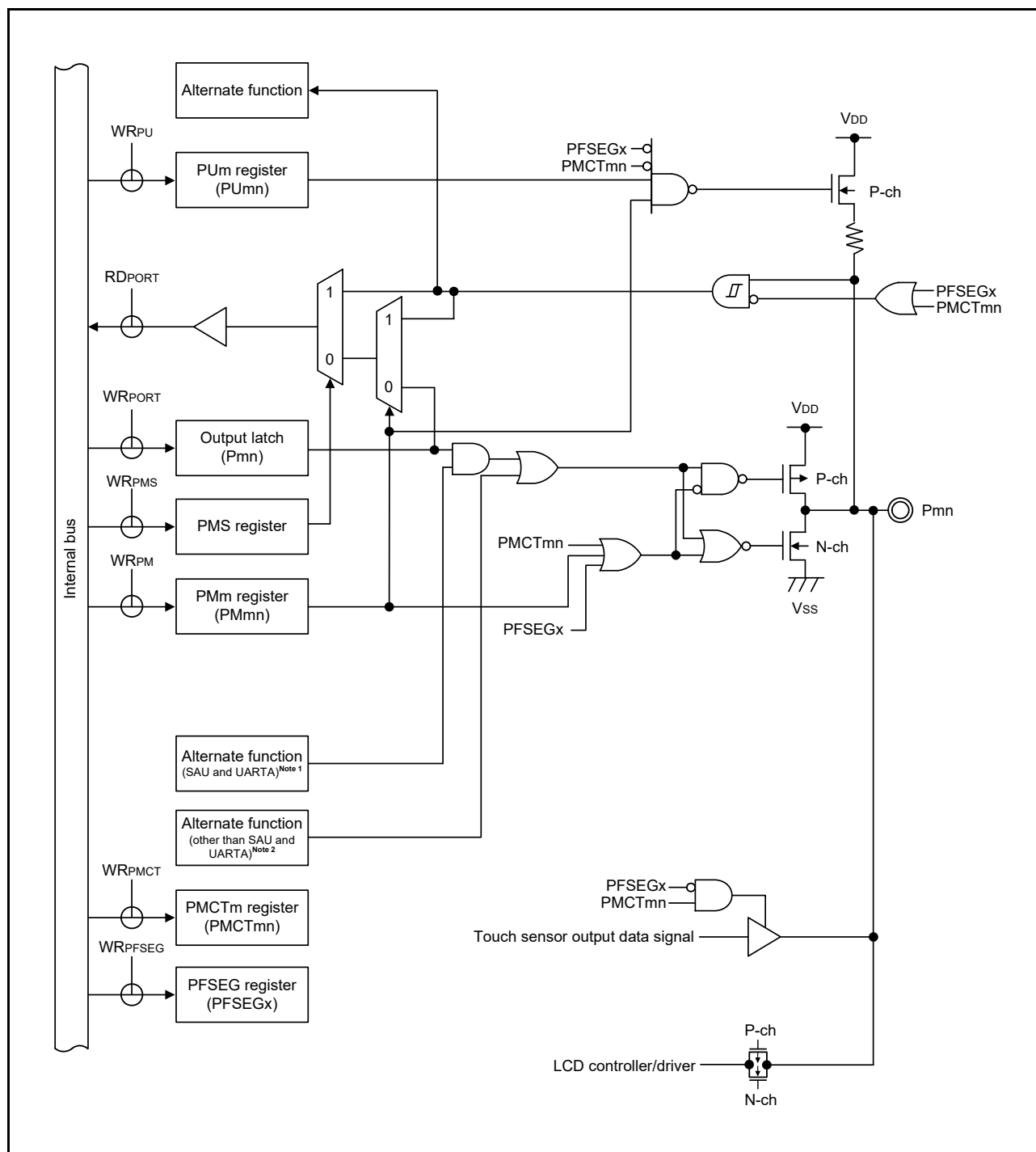
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-10-6 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 (POMm). This may lead to a through current flowing through the type 7-10-6 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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 - 18 Pin Block Diagram for Pin Type 7-44-1



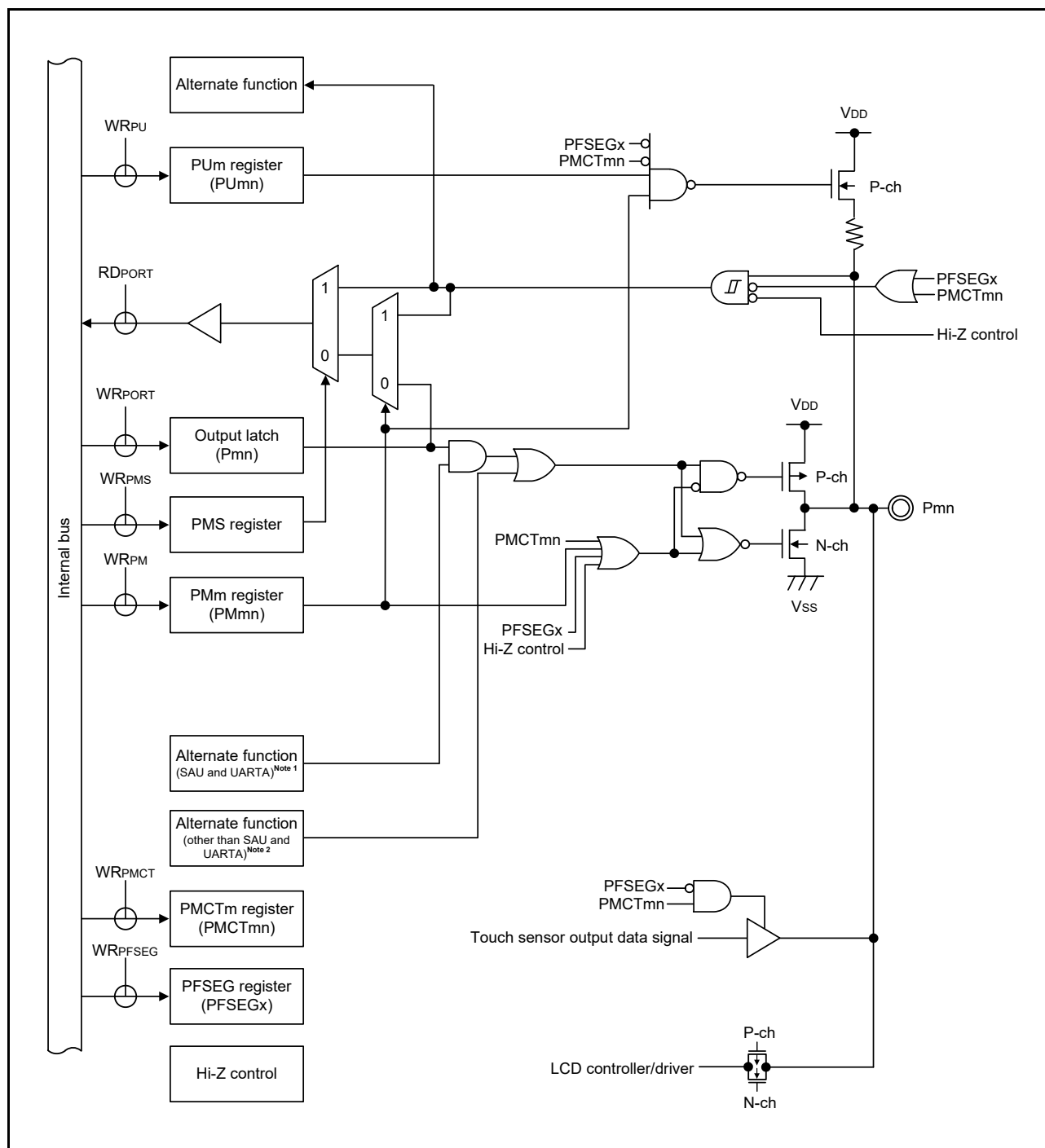
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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 7-44-2



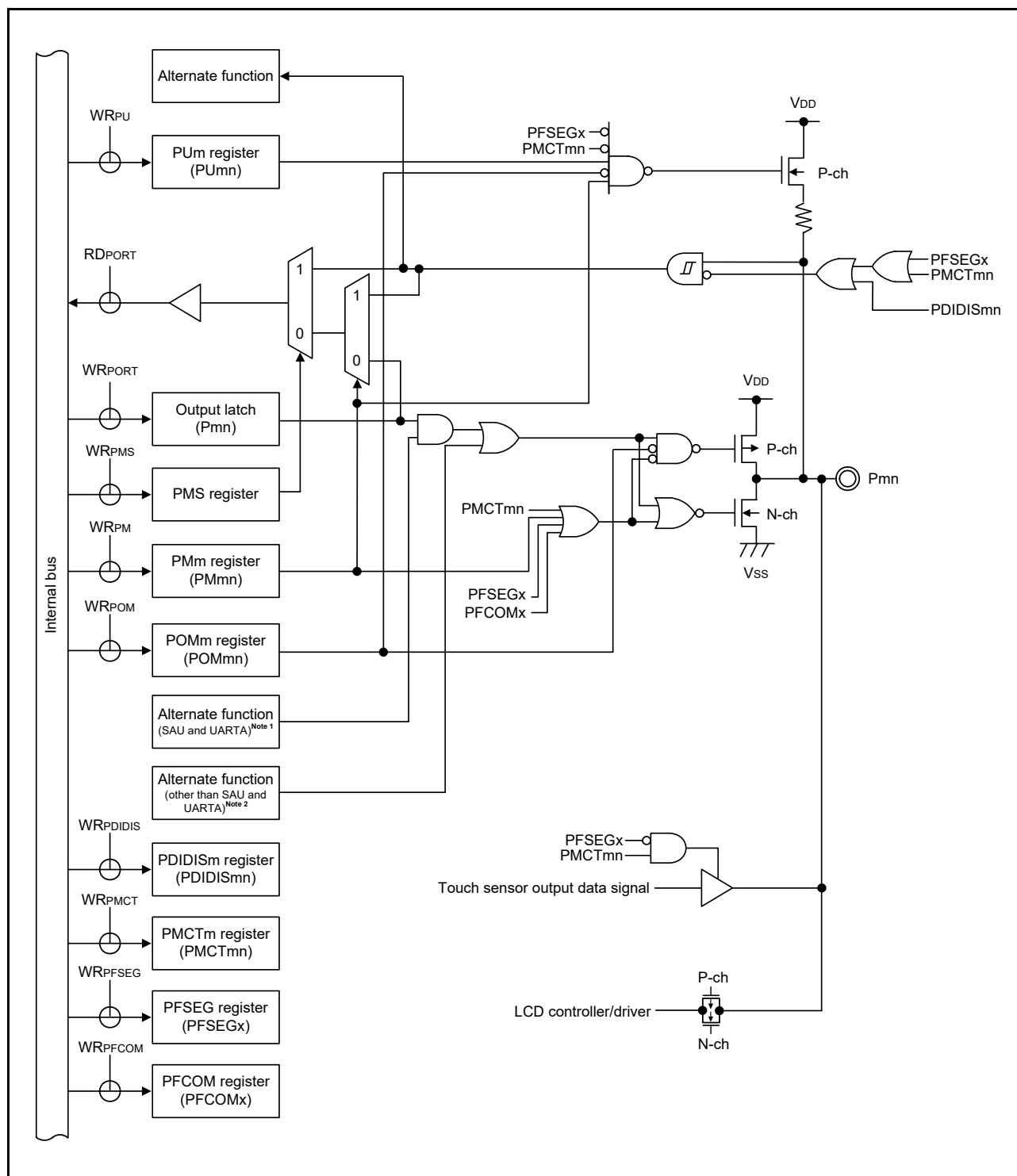
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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 7-44-3



Note 1. This excludes the clock output from the UARTA.

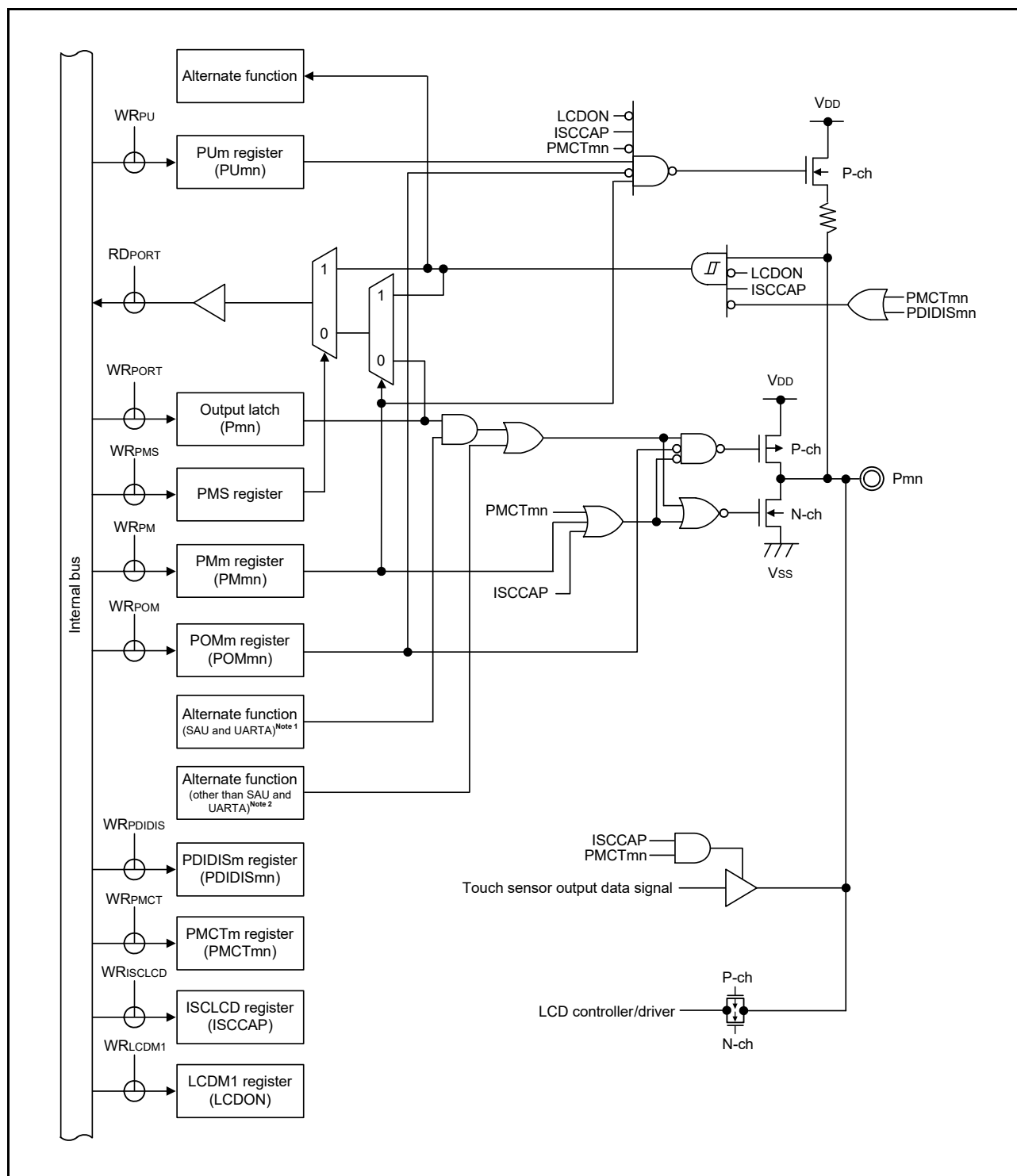
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-44-3 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 (POMm). This may lead to a through current flowing through the type 7-44-3 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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

<R> Figure 2 - 21 Pin Block Diagram for Pin Type 7-44-4



Note 1. This excludes the clock output from the UARTA.

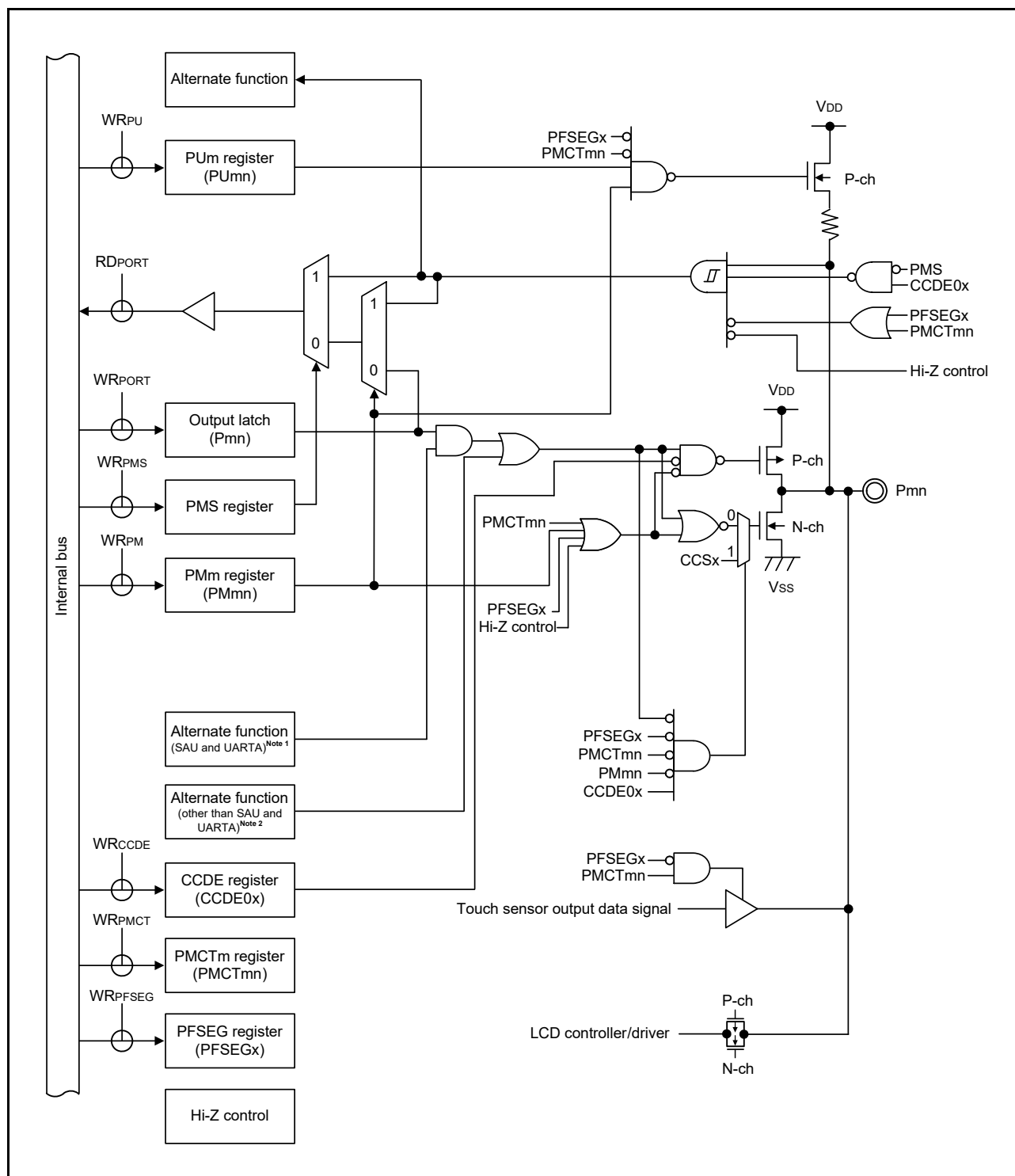
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled even if the type 7-44-4 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 (POMm). This may lead to a through current flowing through the type 7-44-4 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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

<R> Figure 2 - 22 Pin Block Diagram for Pin Type 7-45-1



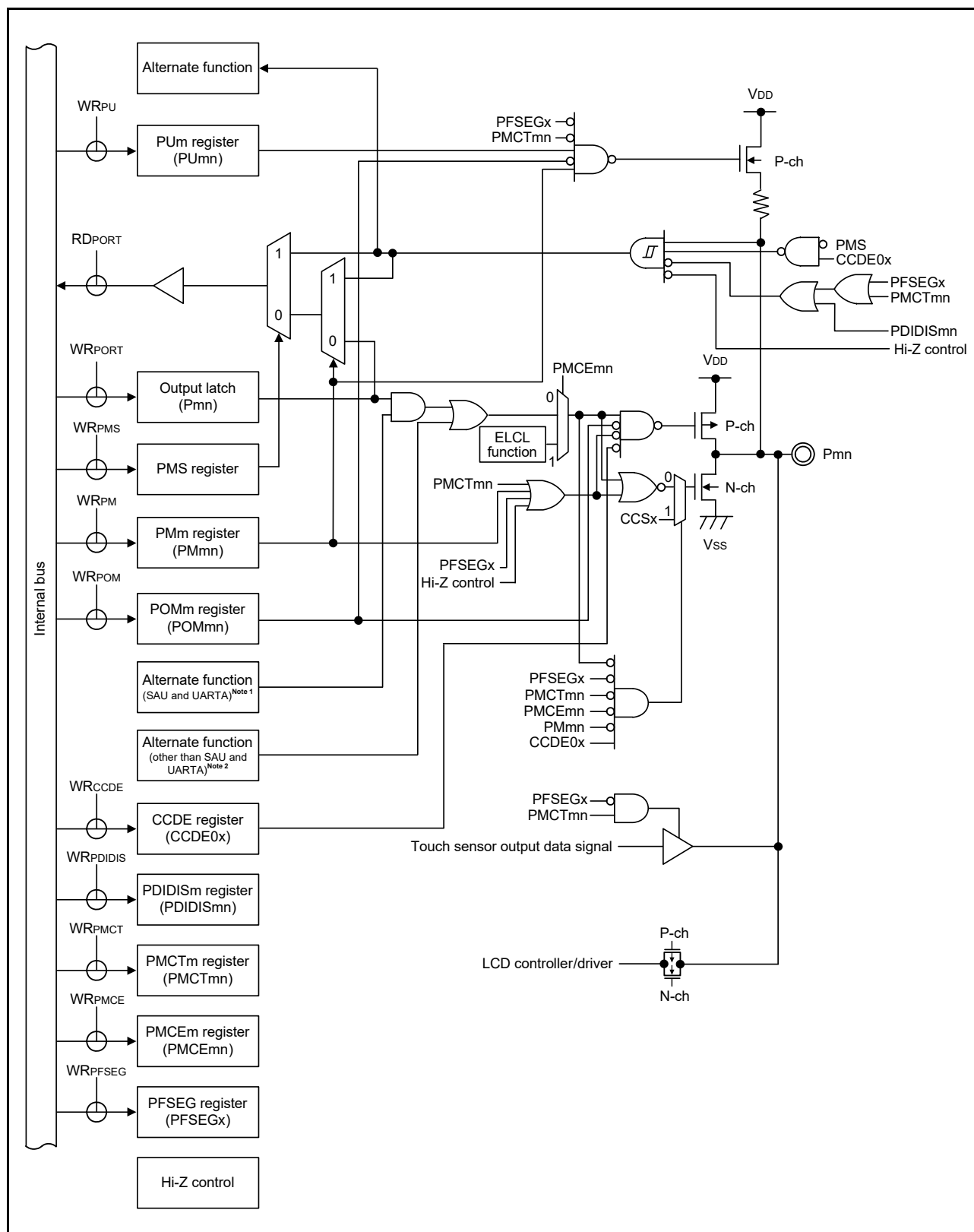
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

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

Remark 2. SAU: Serial array unit

<R> Figure 2 - 23 Pin Block Diagram for Pin Type 7-45-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

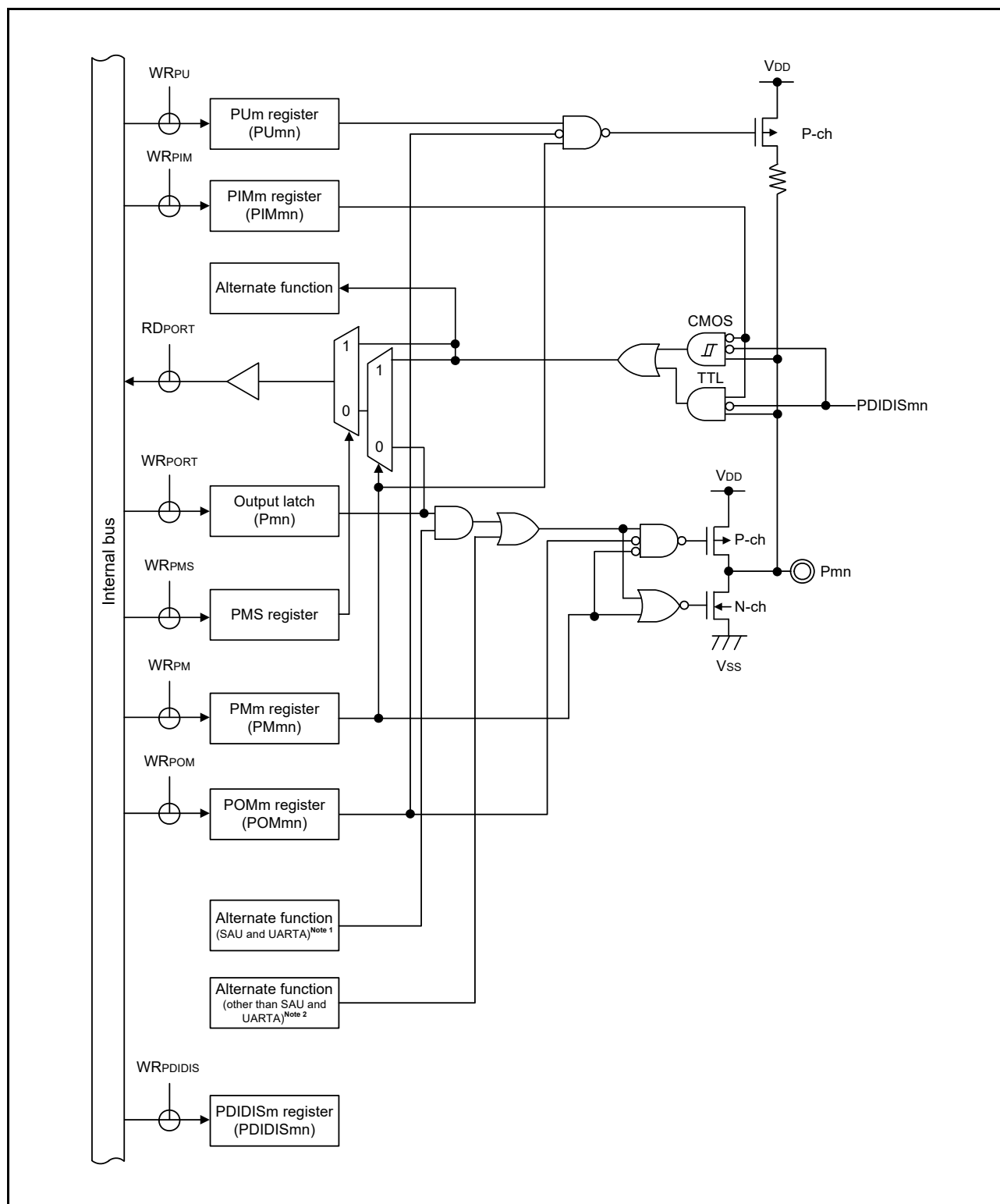
(Caution and Remarks are listed on the next page.)

Caution The input buffer is enabled even if the type 7-45-2 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 (POMm). This may lead to a through current flowing through the type 7-45-2 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm 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 - 24 Pin Block Diagram for Pin Type 8-1-10



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

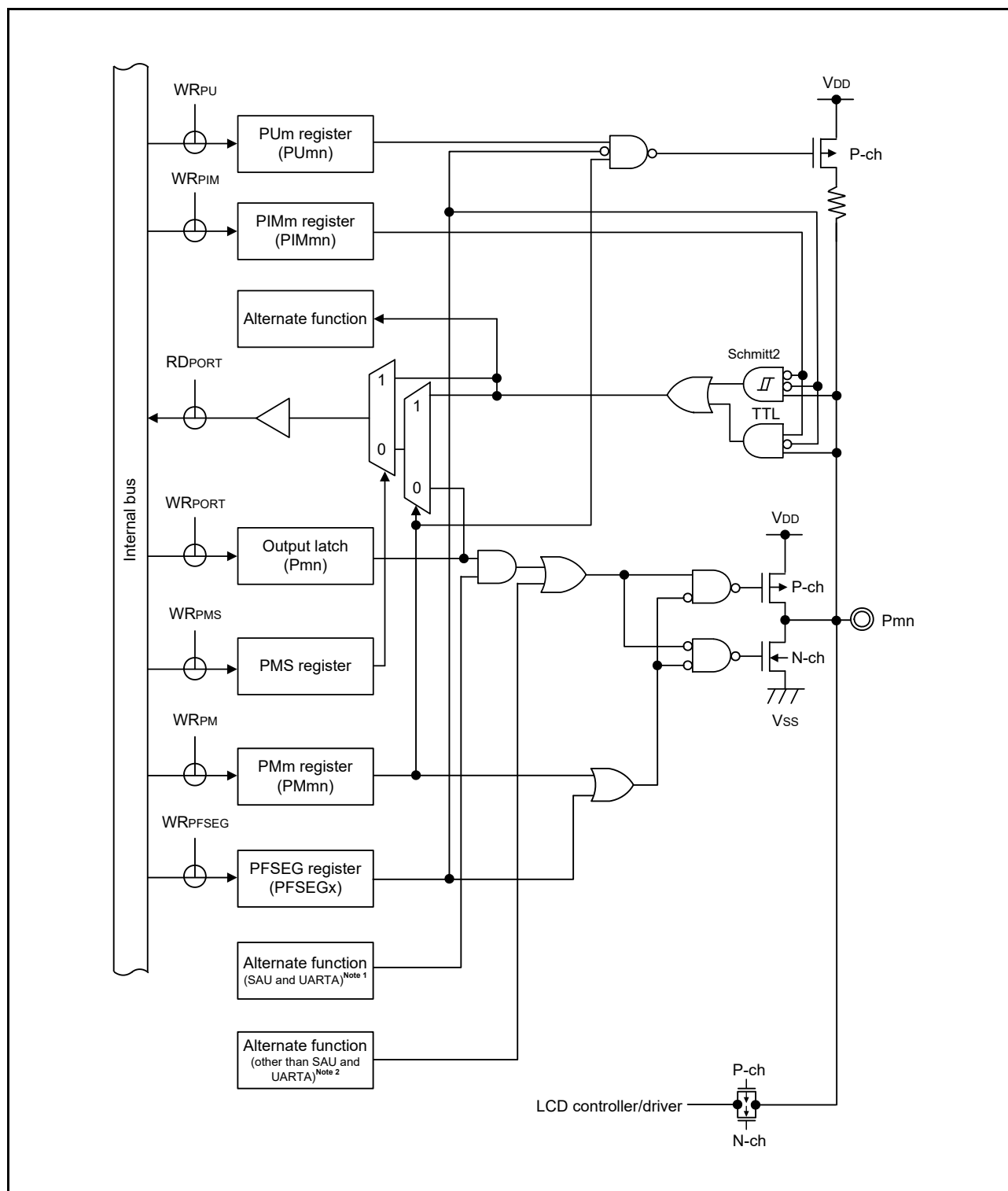
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-1-10 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 (POMm). This may lead to a through current flowing through the type 8-1-10 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-1-10 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-1-10 pin due to the configuration of the TTL input buffer. Drive the type 8-1-10 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 8-5-4



Note 1. This excludes the clock output from the UARTA.

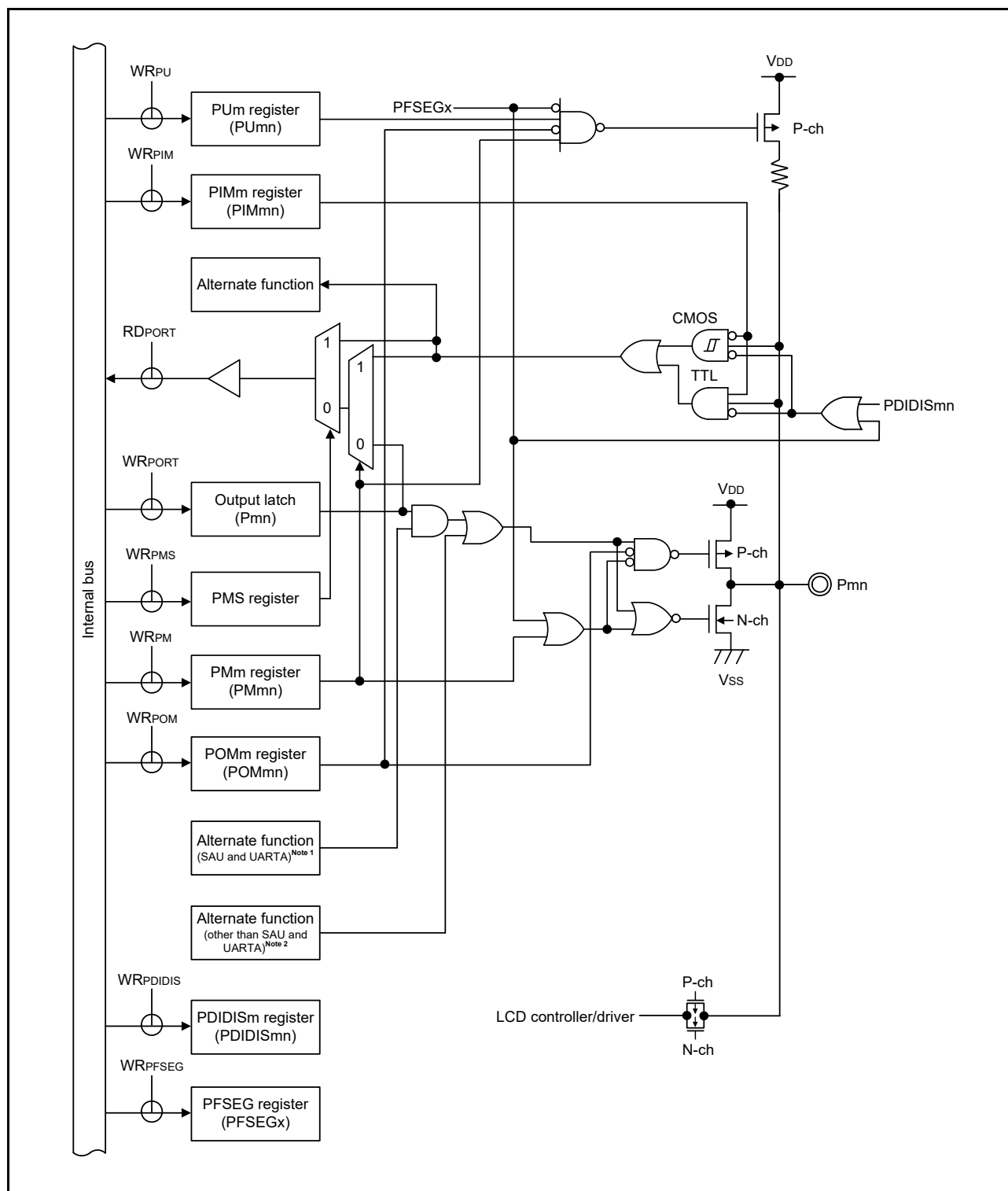
Note 2. This includes the clock output from the UARTA.

Caution When the type 8-5-4 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-5-4 pin due to the configuration of the TTL input buffer. Drive the type 8-5-4 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 - 26 Pin Block Diagram for Pin Type 8-5-17



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Caution 1. The input buffer is enabled even if the type 8-5-17 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 (POMm). This may lead to a through current flowing through the type 8-5-17 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

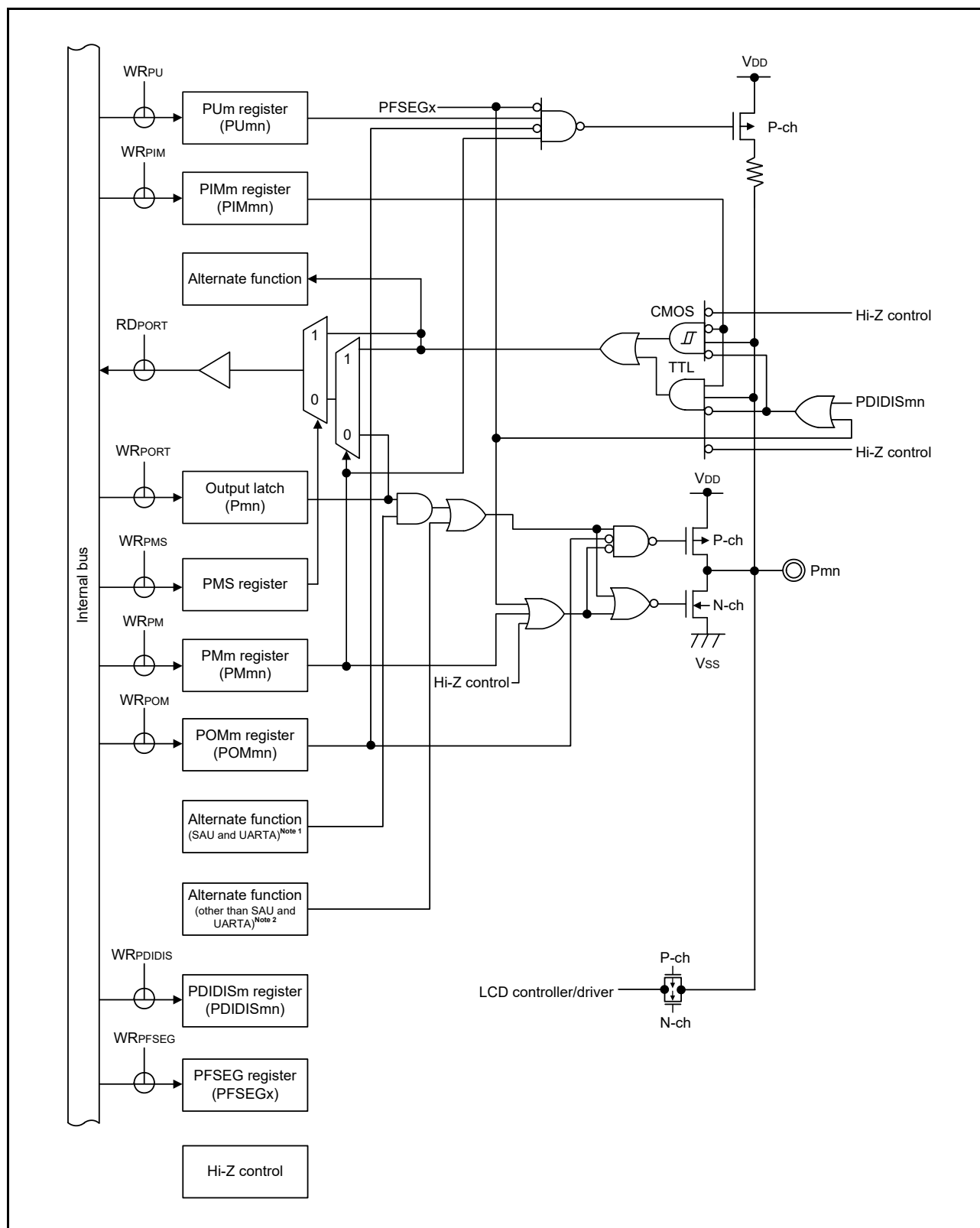
(Caution and Remarks are listed on the next page.)

Caution 2. When the type 8-5-17 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-5-17 pin due to the configuration of the TTL input buffer. Drive the type 8-5-17 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 - 27 Pin Block Diagram for Pin Type 8-5-18



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

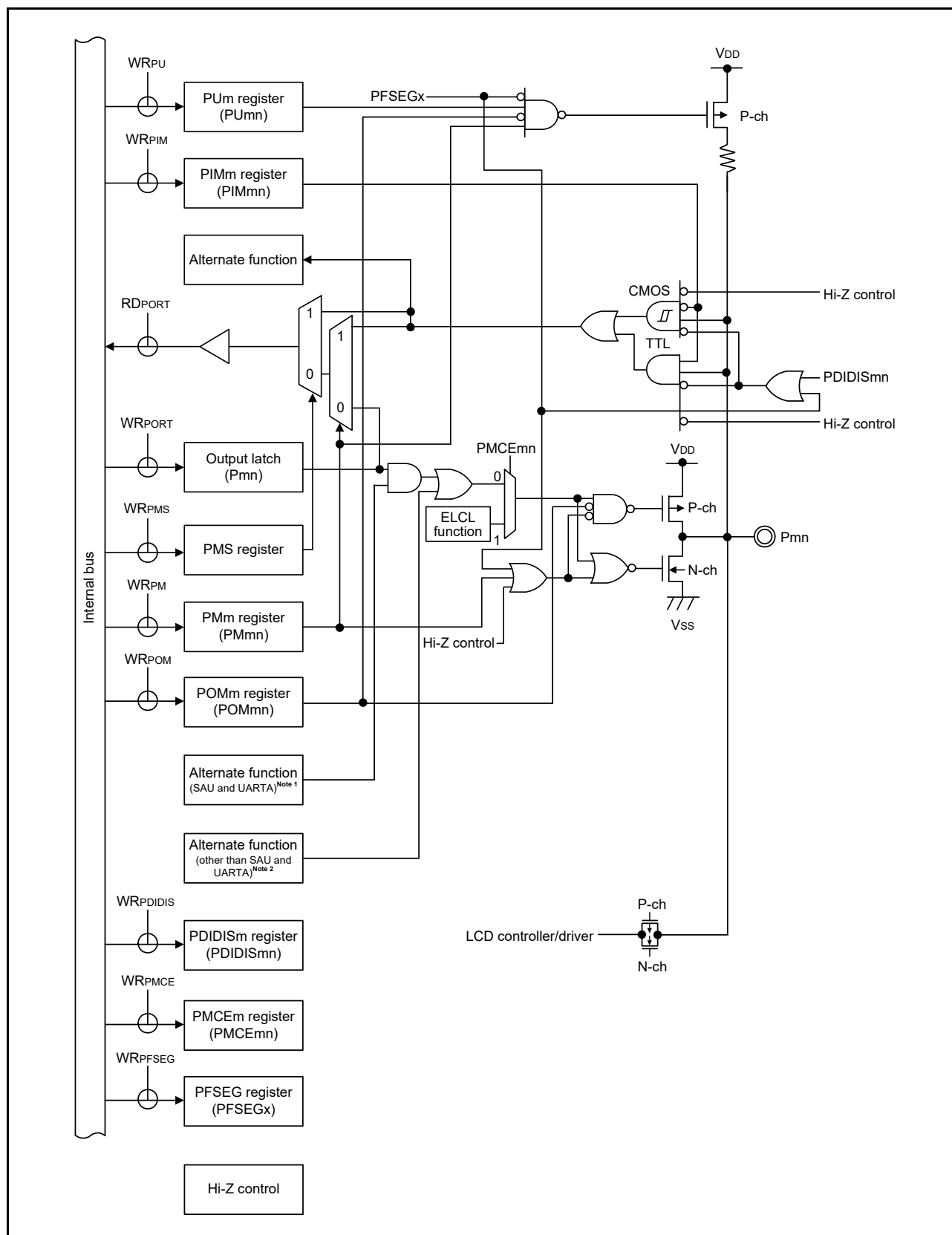
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-5-18 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 (POMm). This may lead to a through current flowing through the type 8-5-18 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-5-18 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-5-18 pin due to the configuration of the TTL input buffer. Drive the type 8-5-18 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 - 28 Pin Block Diagram for Pin Type 8-5-19



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

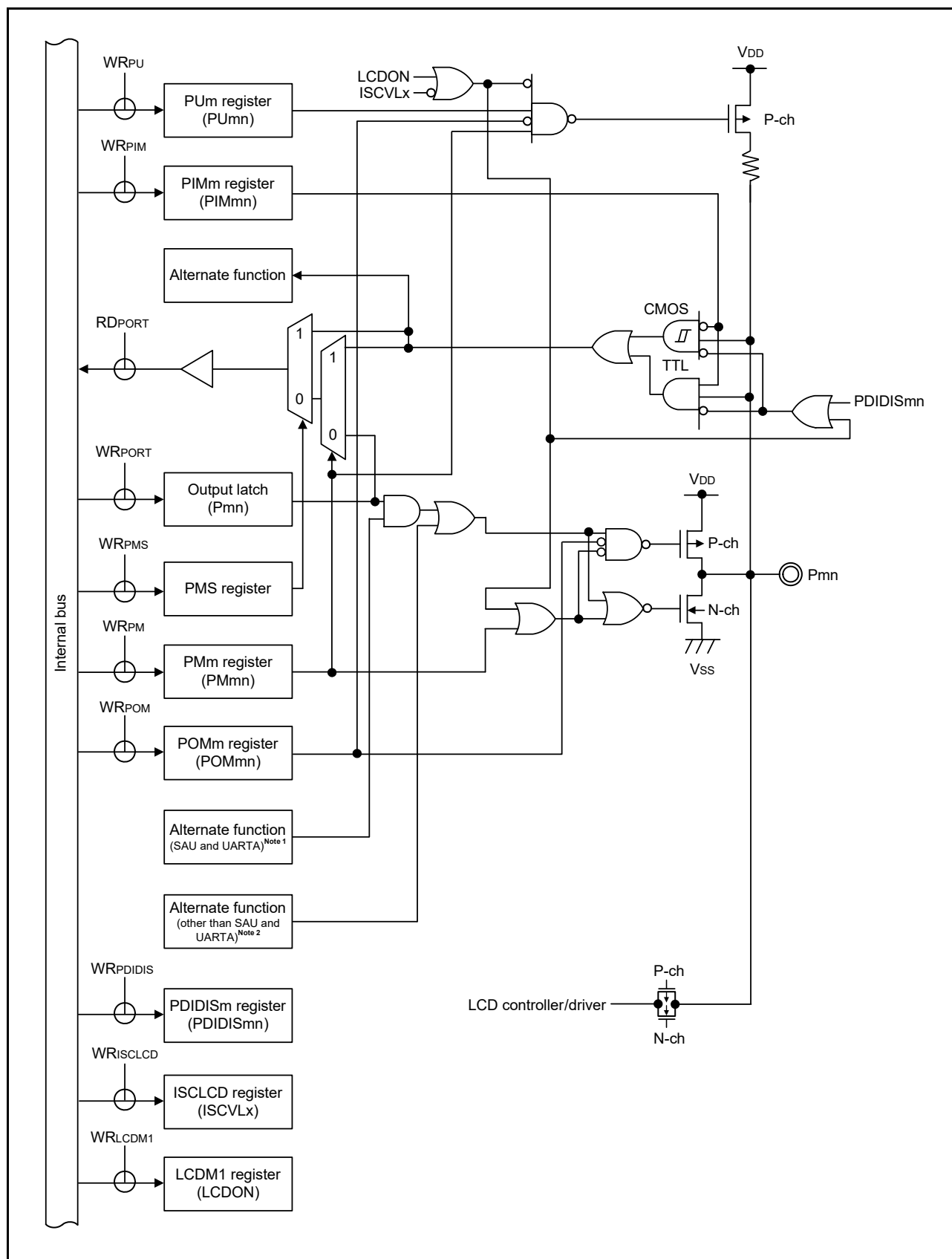
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-5-19 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 (POMm). This may lead to a through current flowing through the type 8-5-19 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-5-19 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-5-19 pin due to the configuration of the TTL input buffer. Drive the type 8-5-19 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

<R> Figure 2 - 29 Pin Block Diagram for Pin Type 8-5-20



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

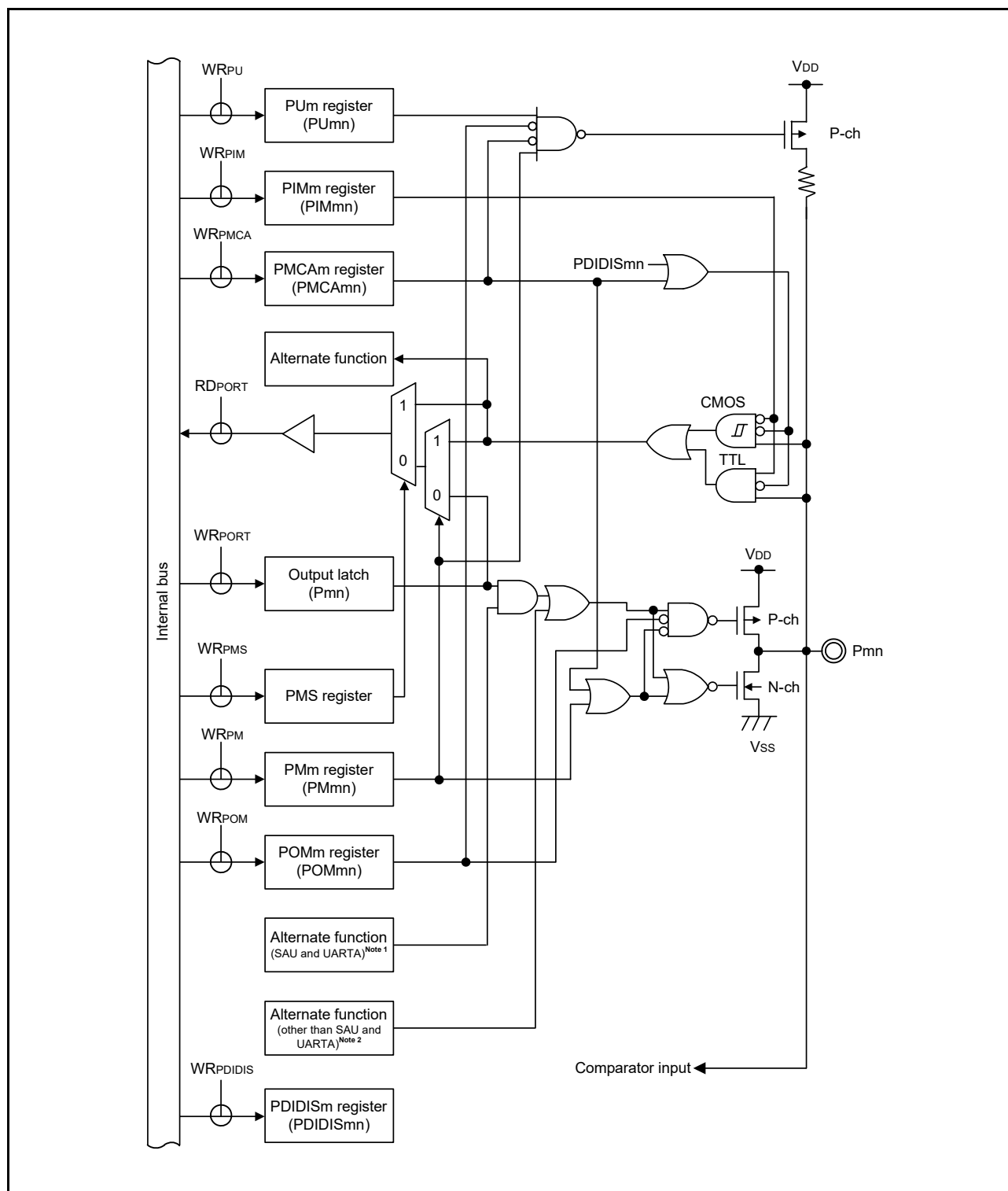
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-5-20 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 (POMm). This may lead to a through current flowing through the type 8-5-20 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-5-20 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-5-20 pin due to the configuration of the TTL input buffer. Drive the type 8-5-20 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 - 30 Pin Block Diagram for Pin Type 8-6-10



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Caution 1. The input buffer is enabled even if the type 8-6-10 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 (POMm). This may lead to a through current flowing through the type 8-6-10 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

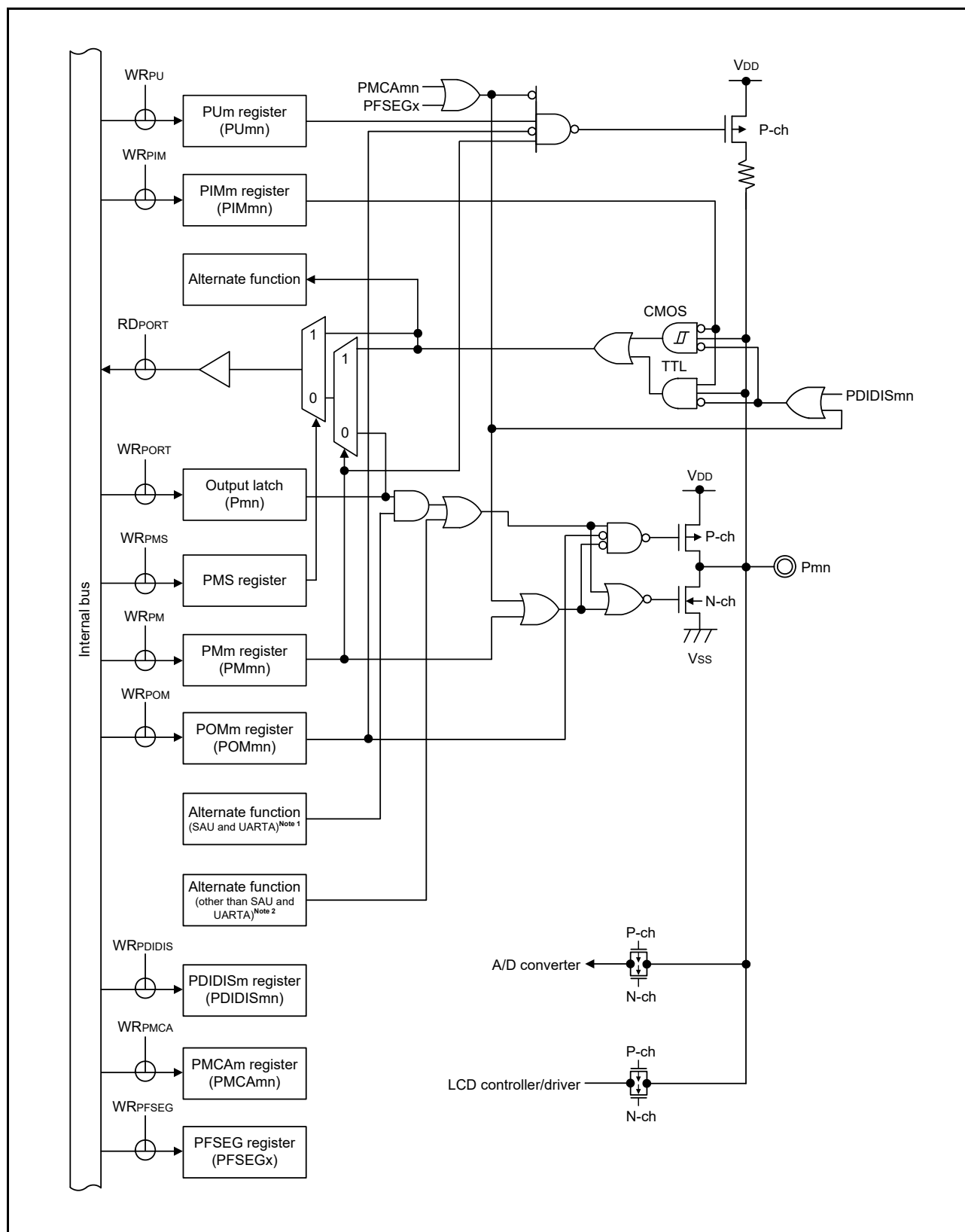
(Caution and Remarks are listed on the next page.)

Caution 2. When the type 8-6-10 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-6-10 pin due to the configuration of the TTL input buffer. Drive the type 8-6-10 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 - 31 Pin Block Diagram for Pin Type 8-10-1



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

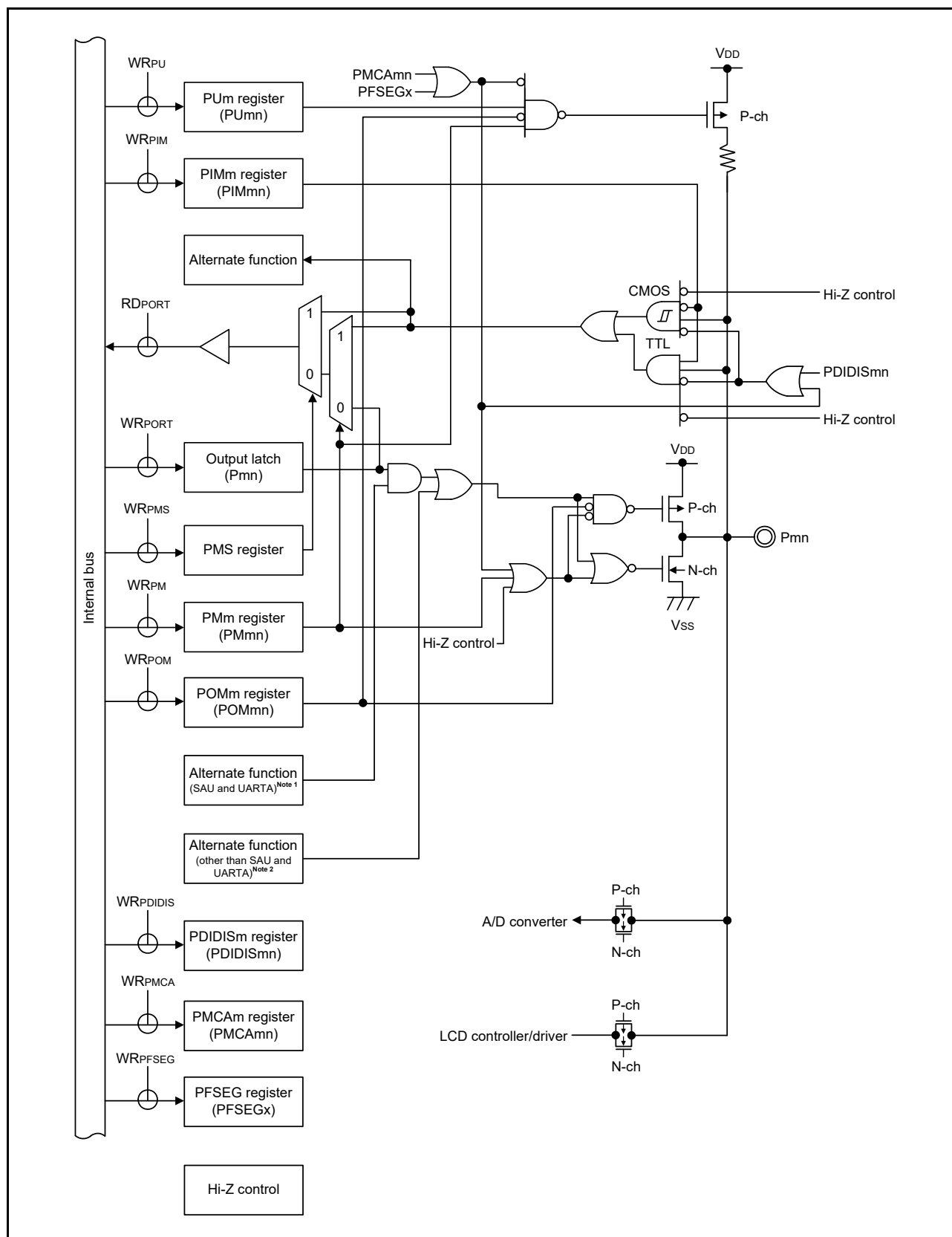
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-10-1 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 (POMm). This may lead to a through current flowing through the type 8-10-1 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-10-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-10-1 pin due to the configuration of the TTL input buffer. Drive the type 8-10-1 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 - 32 Pin Block Diagram for Pin Type 8-10-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

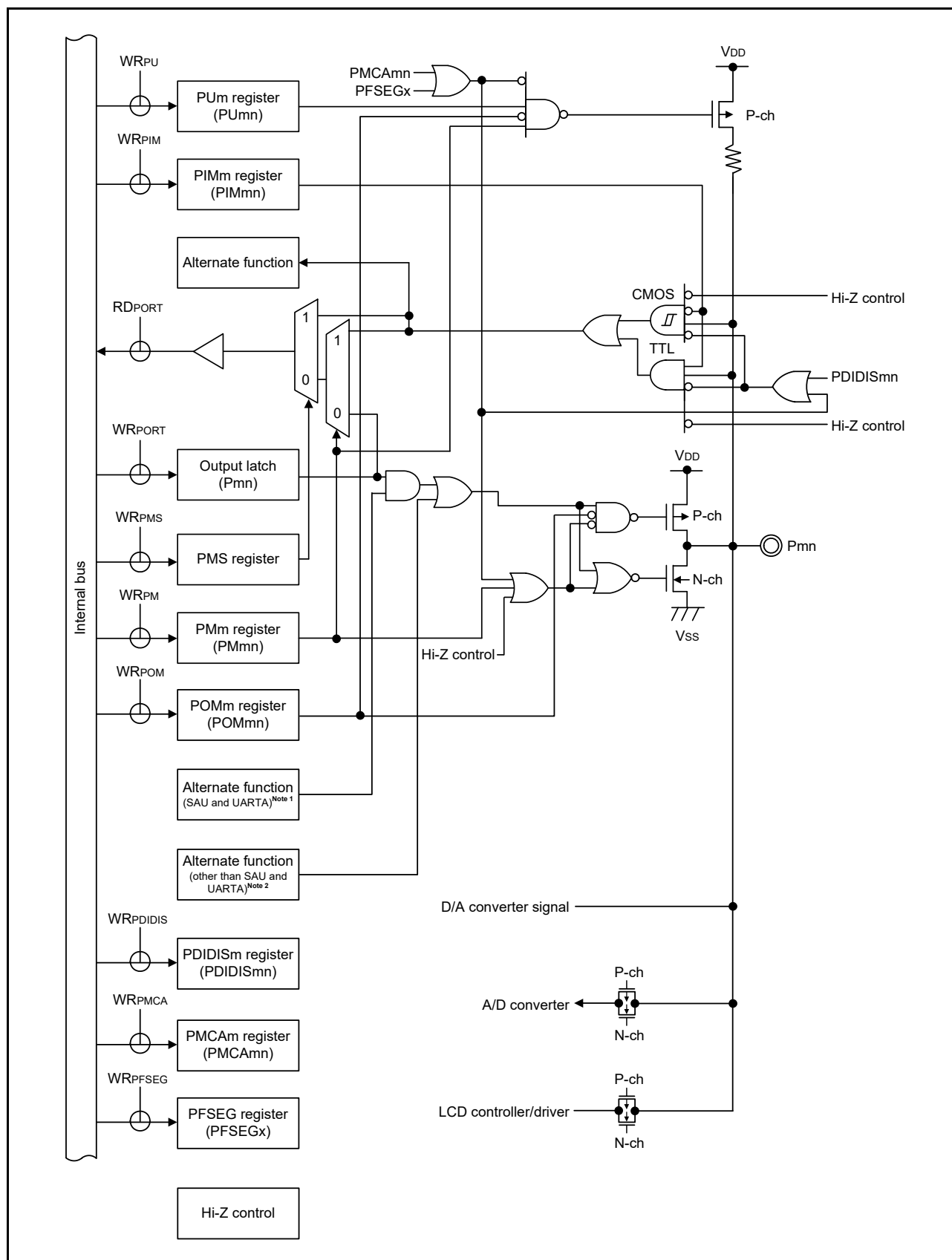
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-10-2 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 (POMm). This may lead to a through current flowing through the type 8-10-2 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-10-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-10-2 pin due to the configuration of the TTL input buffer. Drive the type 8-10-2 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 - 33 Pin Block Diagram for Pin Type 8-43-1



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

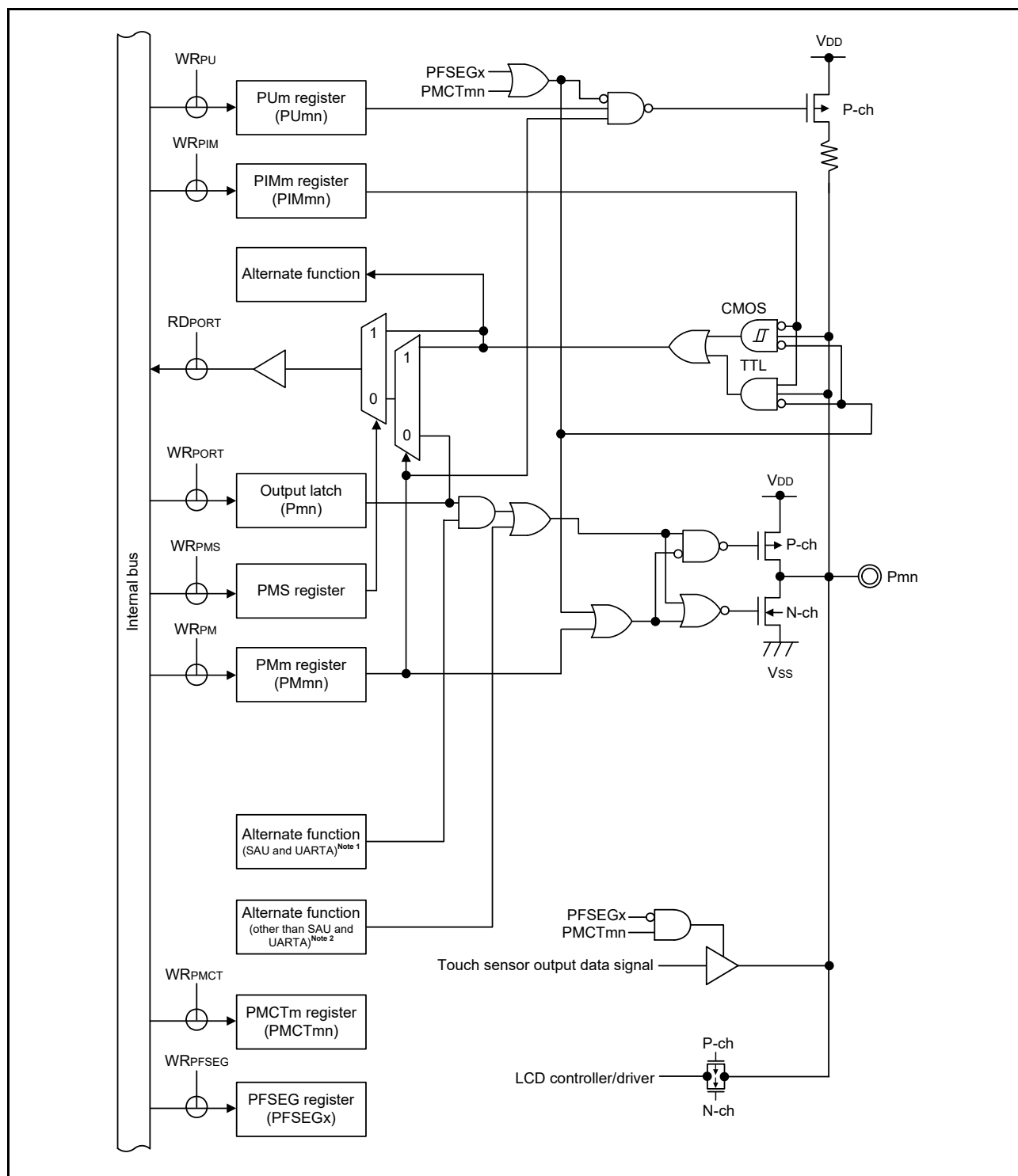
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-43-1 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 (POMm). This may lead to a through current flowing through the type 8-43-1 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-43-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-43-1 pin due to the configuration of the TTL input buffer. Drive the type 8-43-1 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 - 34 Pin Block Diagram for Pin Type 8-44-1



Note 1. This excludes the clock output from the UARTA.

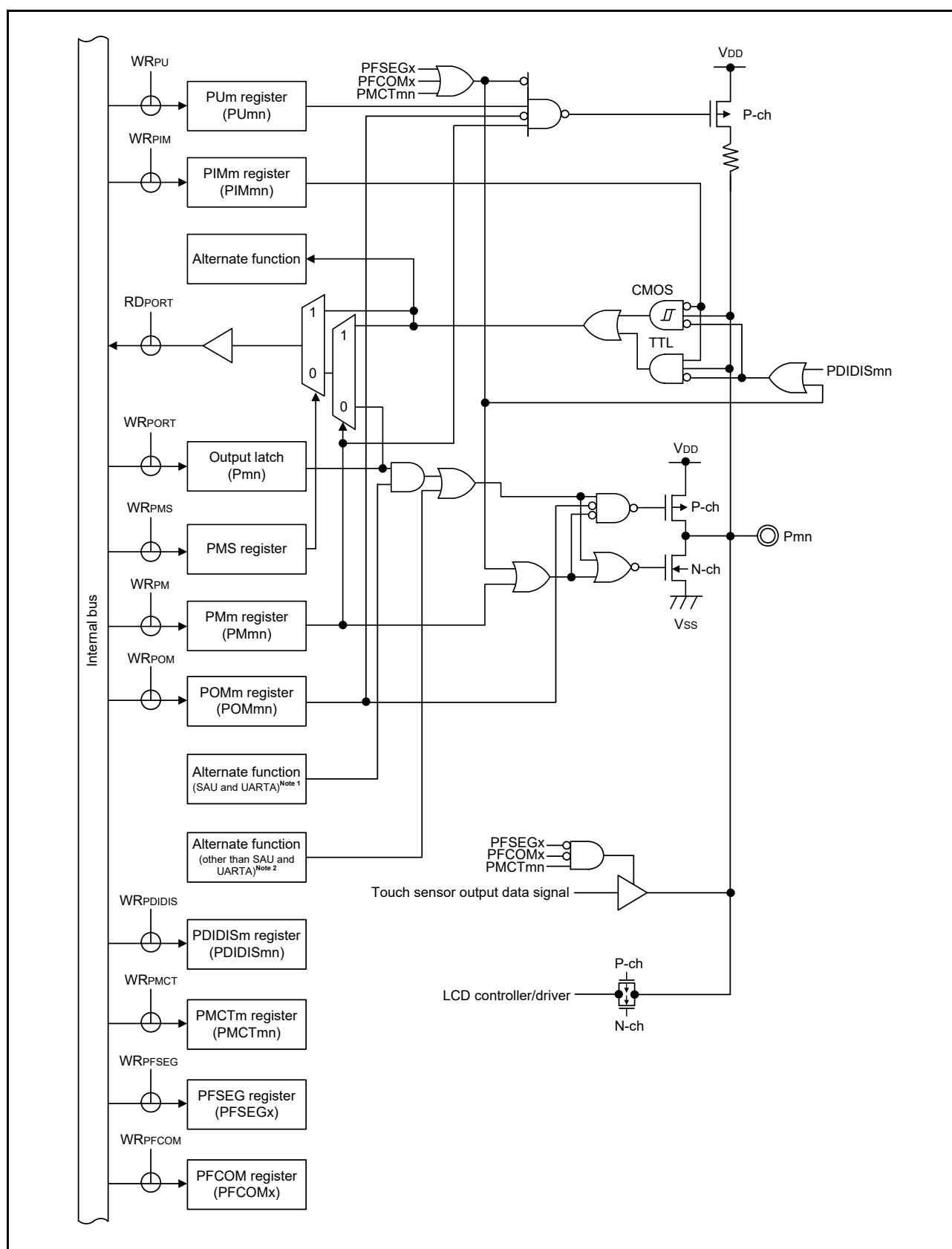
Note 2. This includes the clock output from the UARTA.

Caution When the type 8-44-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-44-1 pin due to the configuration of the TTL input buffer. Drive the type 8-44-1 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 - 35 Pin Block Diagram for Pin Type 8-44-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

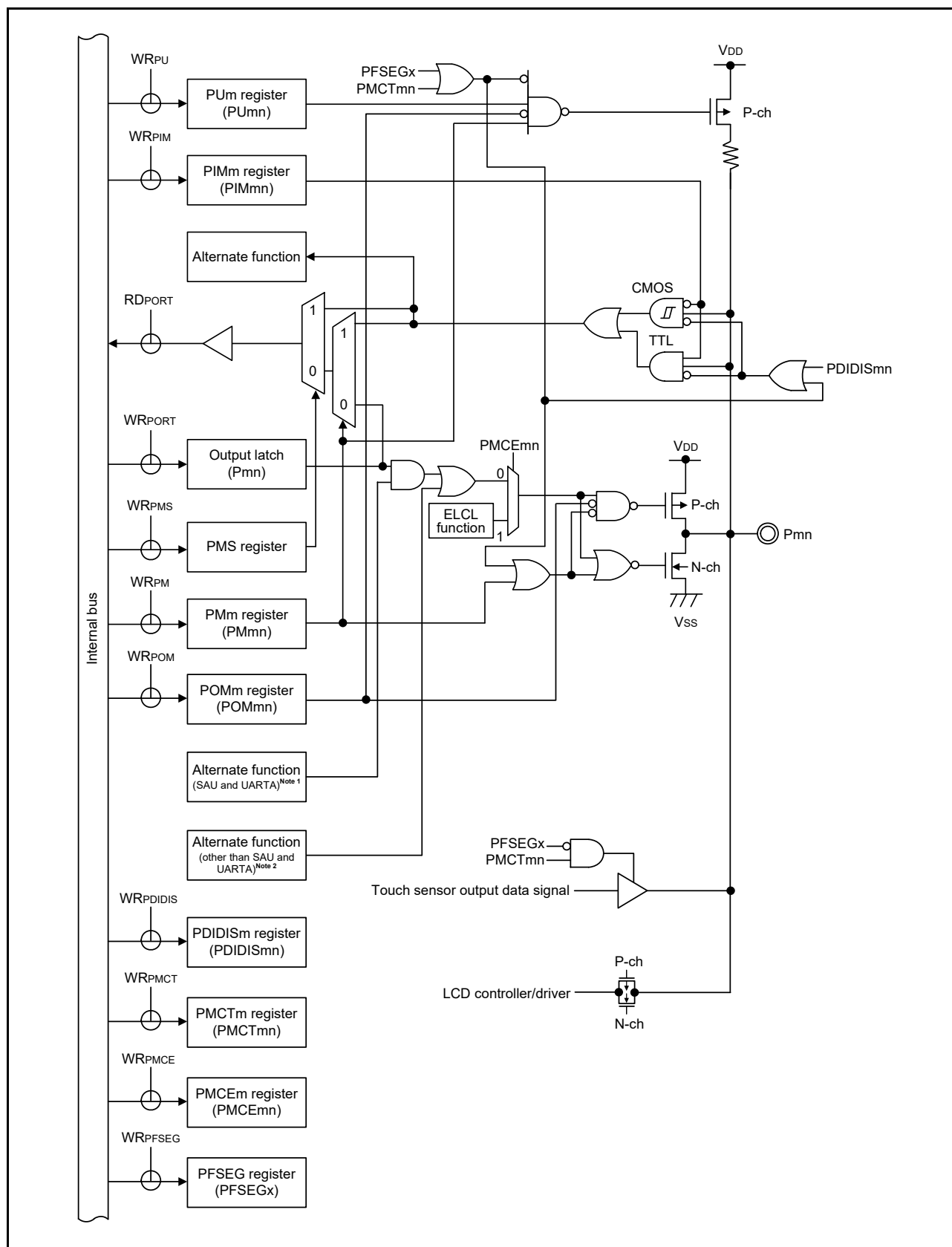
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-44-2 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 (POMm). This may lead to a through current flowing through the type 8-44-2 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-44-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-44-2 pin due to the configuration of the TTL input buffer. Drive the type 8-44-2 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 - 36 Pin Block Diagram for Pin Type 8-44-3



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

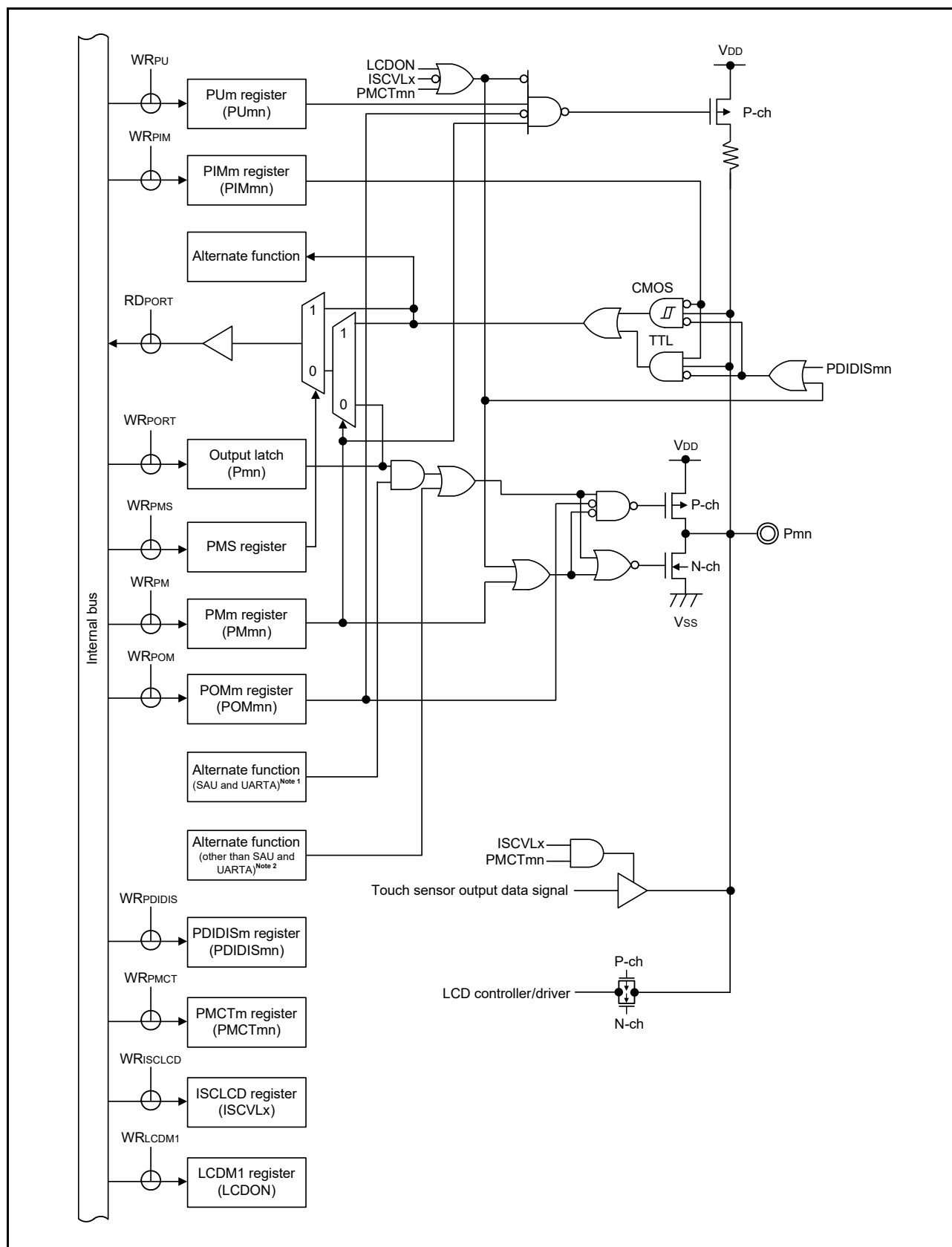
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-44-3 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 (POMm). This may lead to a through current flowing through the type 8-44-3 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-44-3 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-44-3 pin due to the configuration of the TTL input buffer. Drive the type 8-44-3 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

<R> Figure 2 - 37 Pin Block Diagram for Pin Type 8-44-4



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

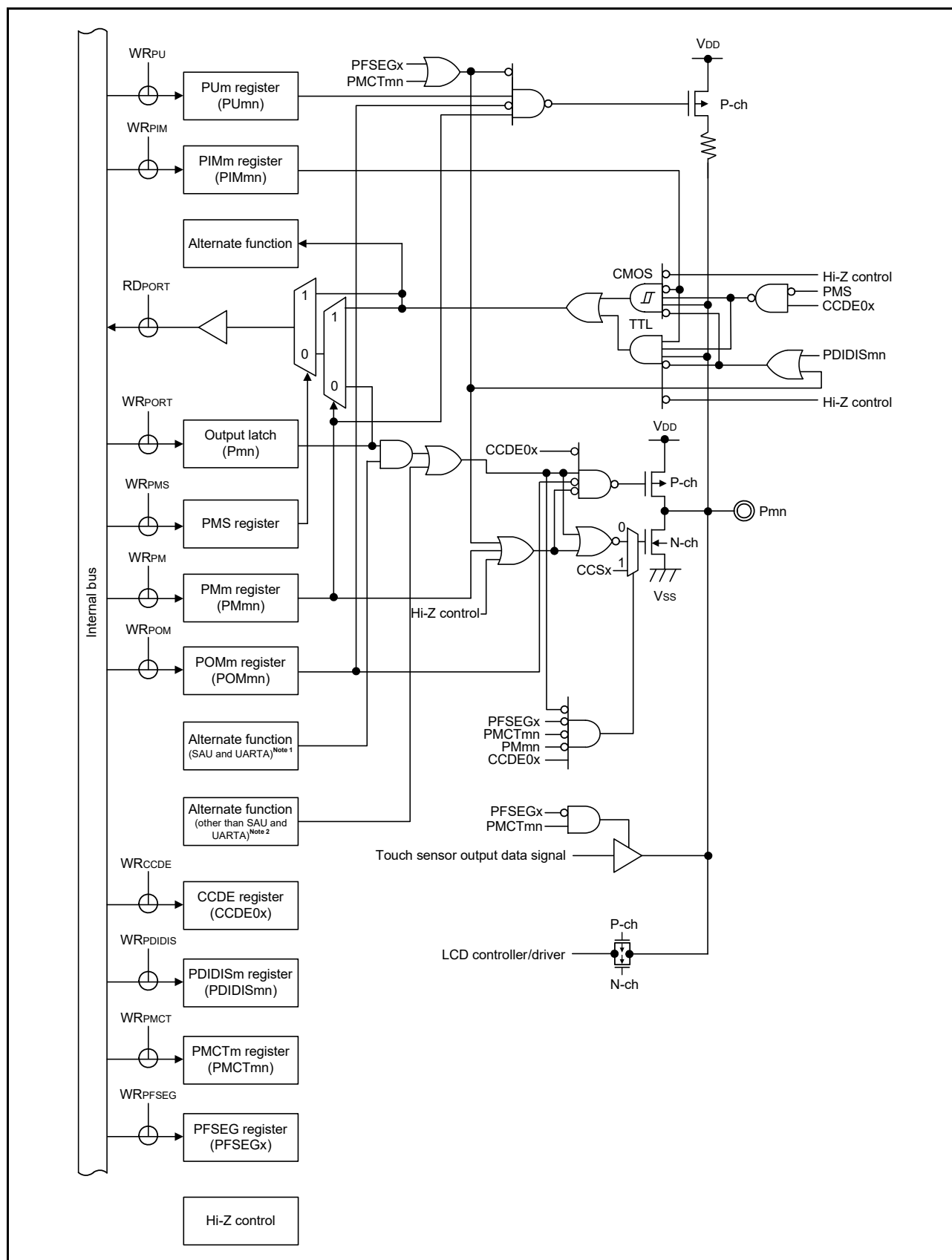
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-44-4 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 (POMm). This may lead to a through current flowing through the type 8-44-4 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-44-4 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-44-4 pin due to the configuration of the TTL input buffer. Drive the type 8-44-4 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

<R> Figure 2 - 38 Pin Block Diagram for Pin Type 8-45-1



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

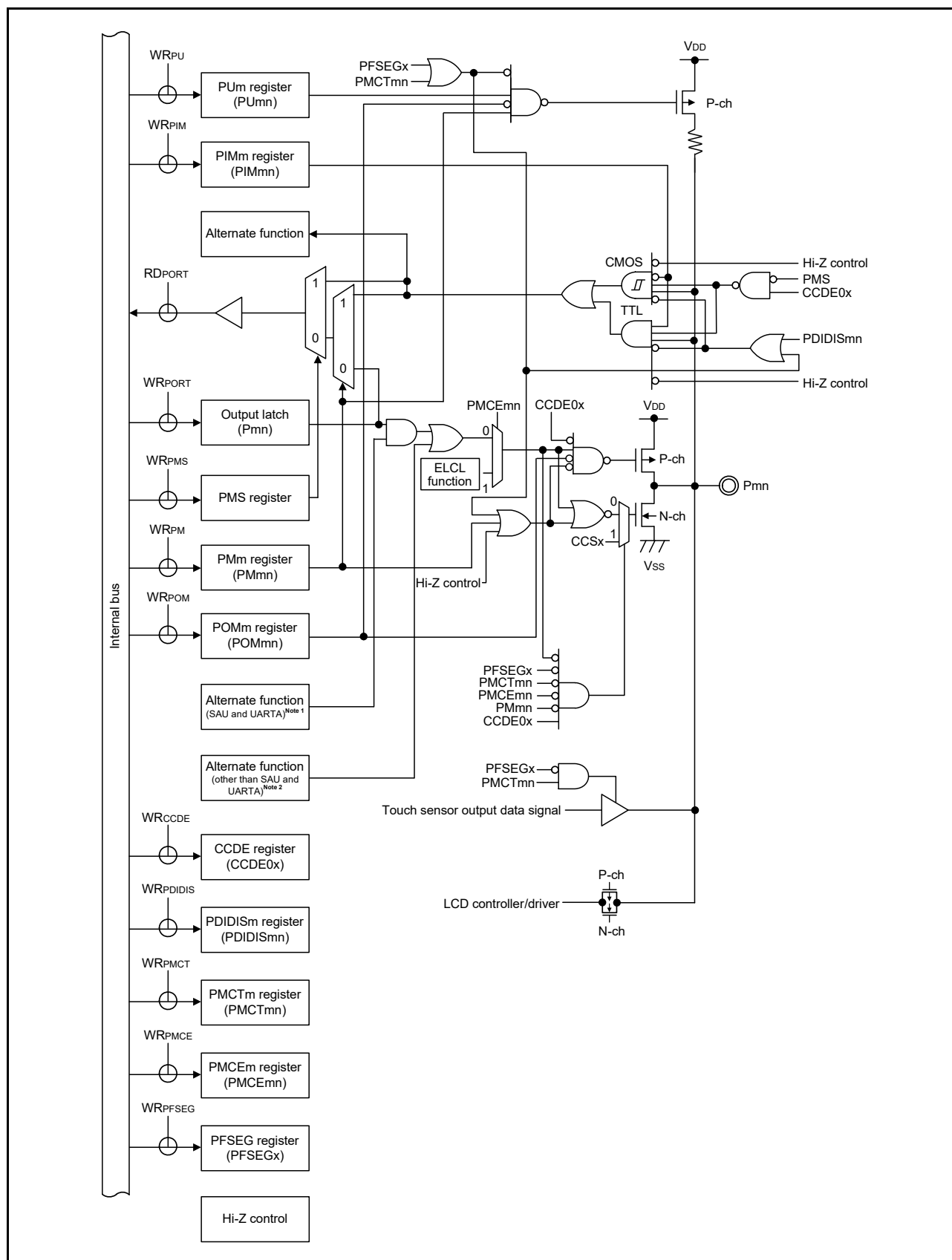
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-45-1 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 (POMm). This may lead to a through current flowing through the type 8-45-1 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-45-1 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-45-1 pin due to the configuration of the TTL input buffer. Drive the type 8-45-1 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

<R> Figure 2 - 39 Pin Block Diagram for Pin Type 8-45-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

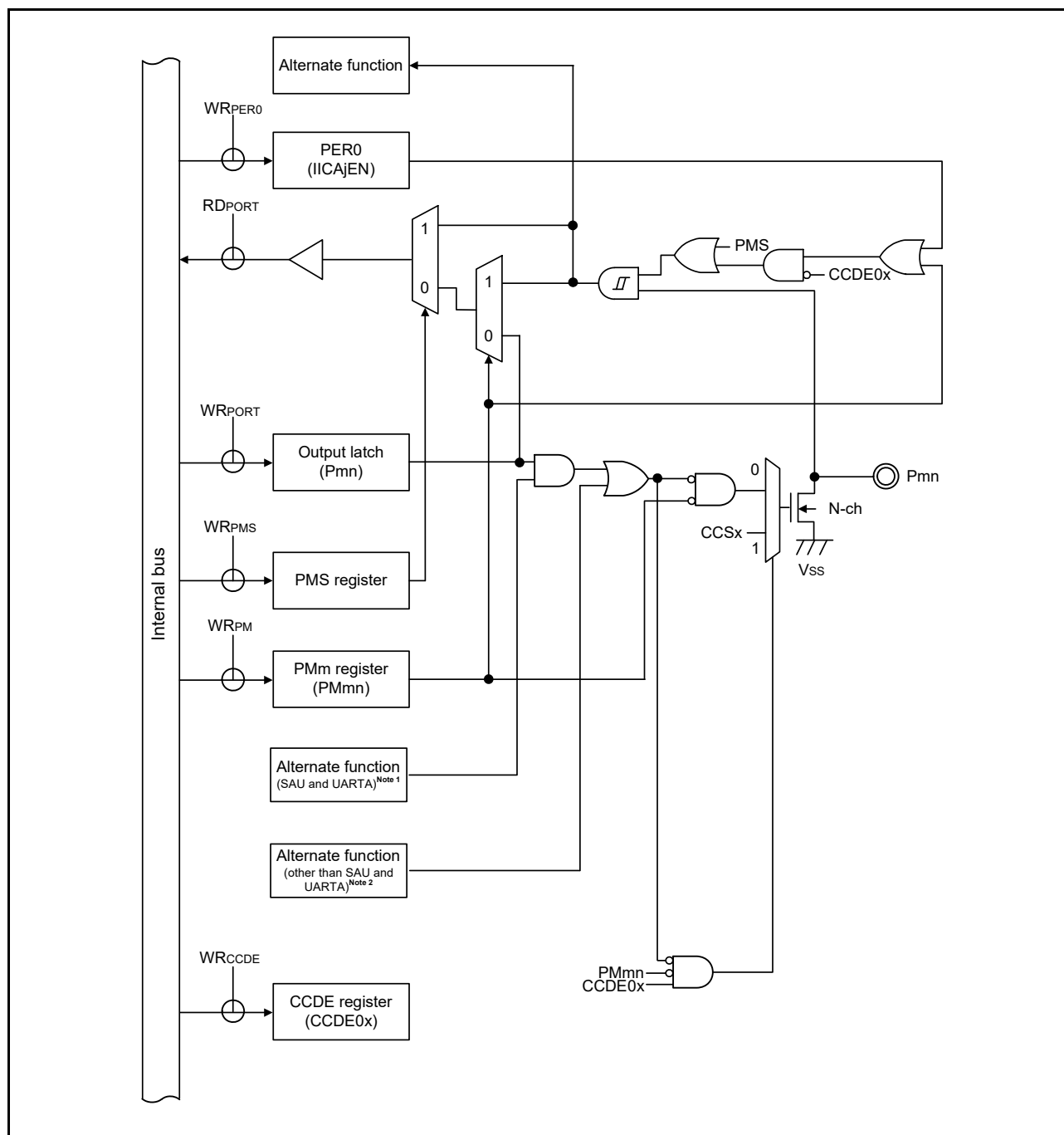
(Cautions and Remarks are listed on the next page.)

- Caution 1.** The input buffer is enabled even if the type 8-45-2 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 (POMm). This may lead to a through current flowing through the type 8-45-2 pin when the voltage level on this pin is intermediate.
However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.
- Caution 2.** When the type 8-45-2 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the type 8-45-2 pin due to the configuration of the TTL input buffer. Drive the type 8-45-2 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

<R> Figure 2 - 40 Pin Block Diagram for Pin Type 12-38-2



Note 1. This excludes the clock output from the UARTA.

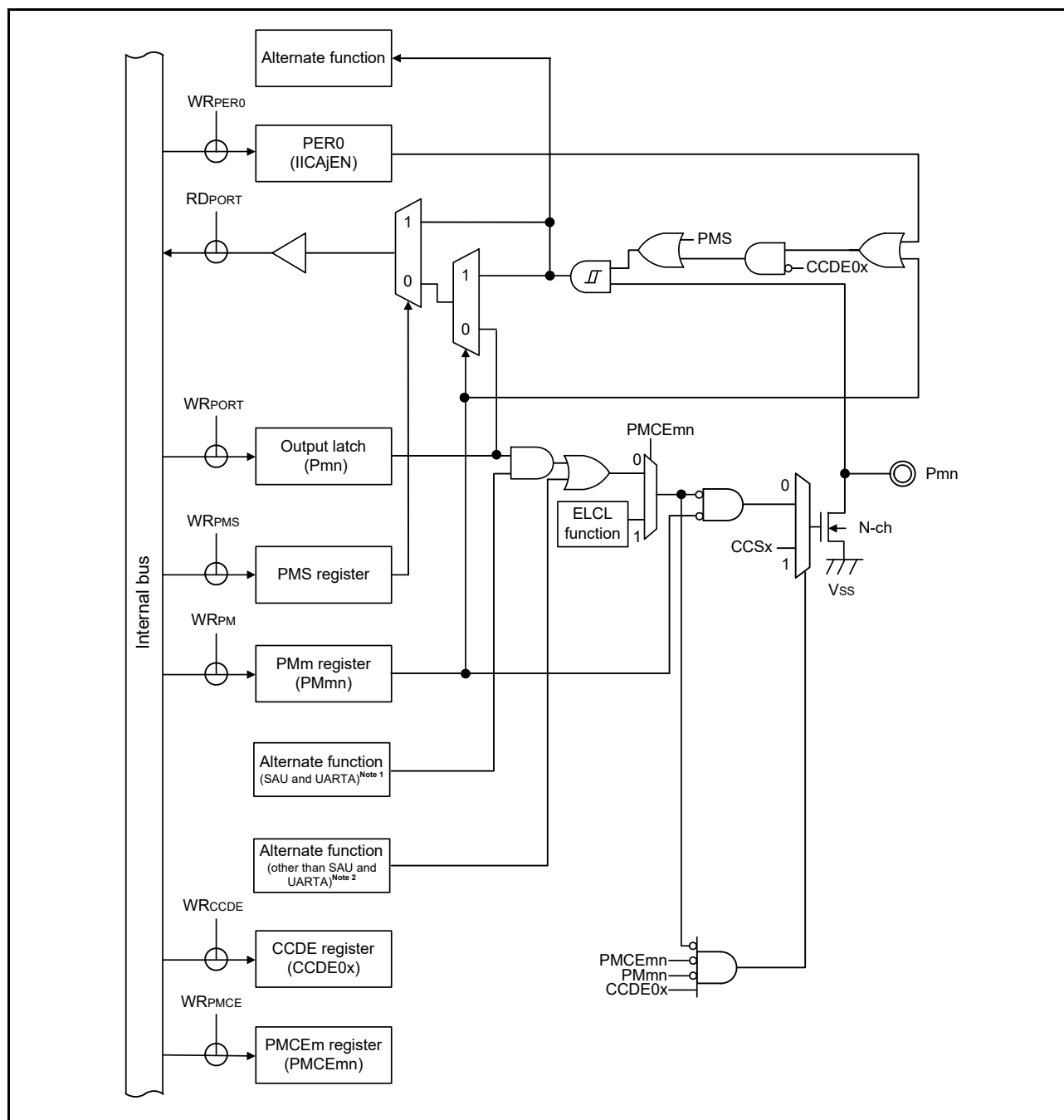
Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled when using the IICA with the IICAjEN bit in the PER0 register being set to 1. This may lead to a through current flowing through the type 12-38-2 pin when the voltage level on this pin is intermediate.

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

Remark 2. SAU: Serial array unit

<R> Figure 2 - 41 Pin Block Diagram for Pin Type 12-38-3



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Caution The input buffer is enabled when using the IICA with the IICAJEN bit in the PER0 register being set to 1. This may lead to a through current flowing through the type 12-38-3 pin when the voltage level on this pin is intermediate.

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

Remark 2. SAU: Serial array unit

Section 3 CPU Architecture

The RL78/L23 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)

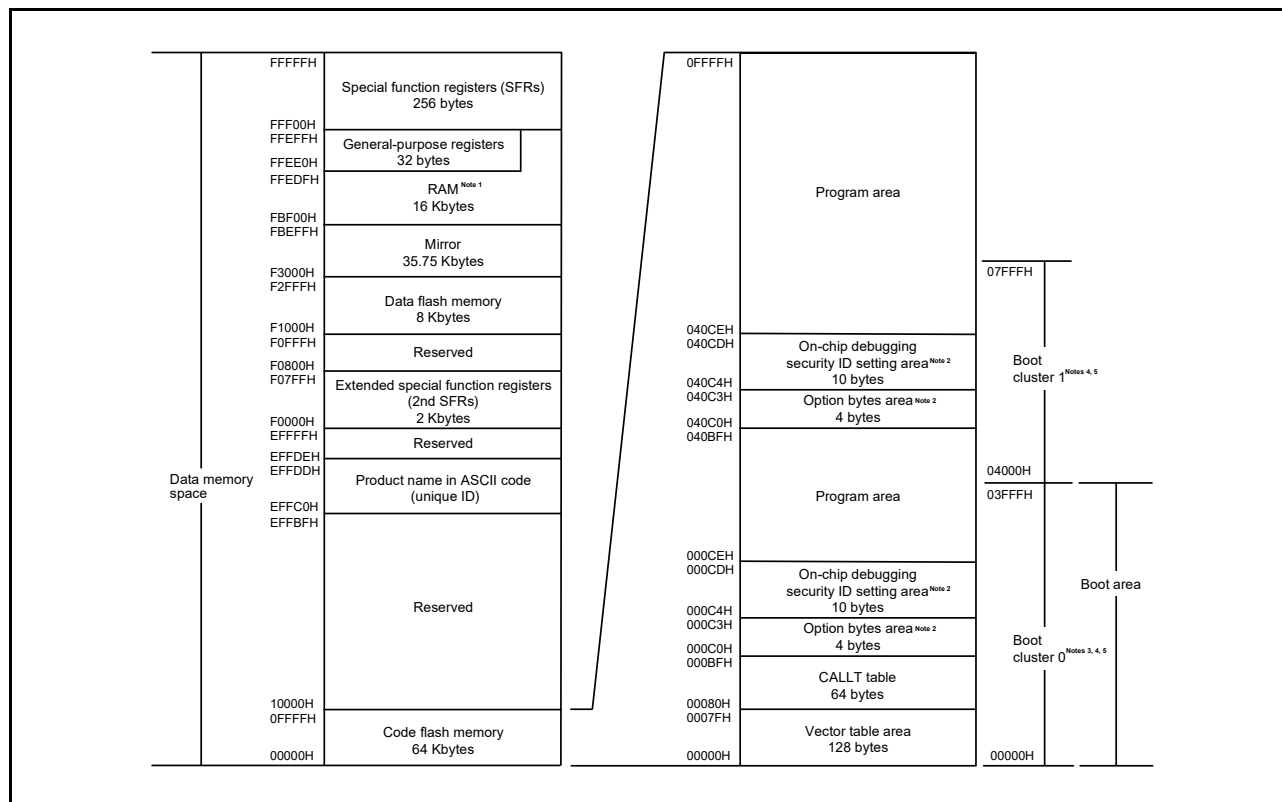
- Data allocation: Little endian

The RL78/L23 supports an OCD trace function.

3.1 Memory Space

Products in the RL78/L23 can access a 1-Mbyte address space. **Figure 3 - 1** to **Figure 3 - 4** show the memory maps.

Figure 3 - 1 Memory Map (R7F100LxE (x = F, G, J, L))



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

Note 2. When boot swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes and on-chip debugging security IDs to the corresponding addresses according to the selected size of the boot cluster as shown below.

- 16 Kbytes: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 040C4H to 040CDH.
- 8 Kbytes: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 020C4H to 020CDH.
- 4 Kbytes: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
- 2 Kbytes: Set the option bytes to 000C0H to 000C3H and 008C0H to 008C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 008C4H to 008CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **38.10 Security Settings**.

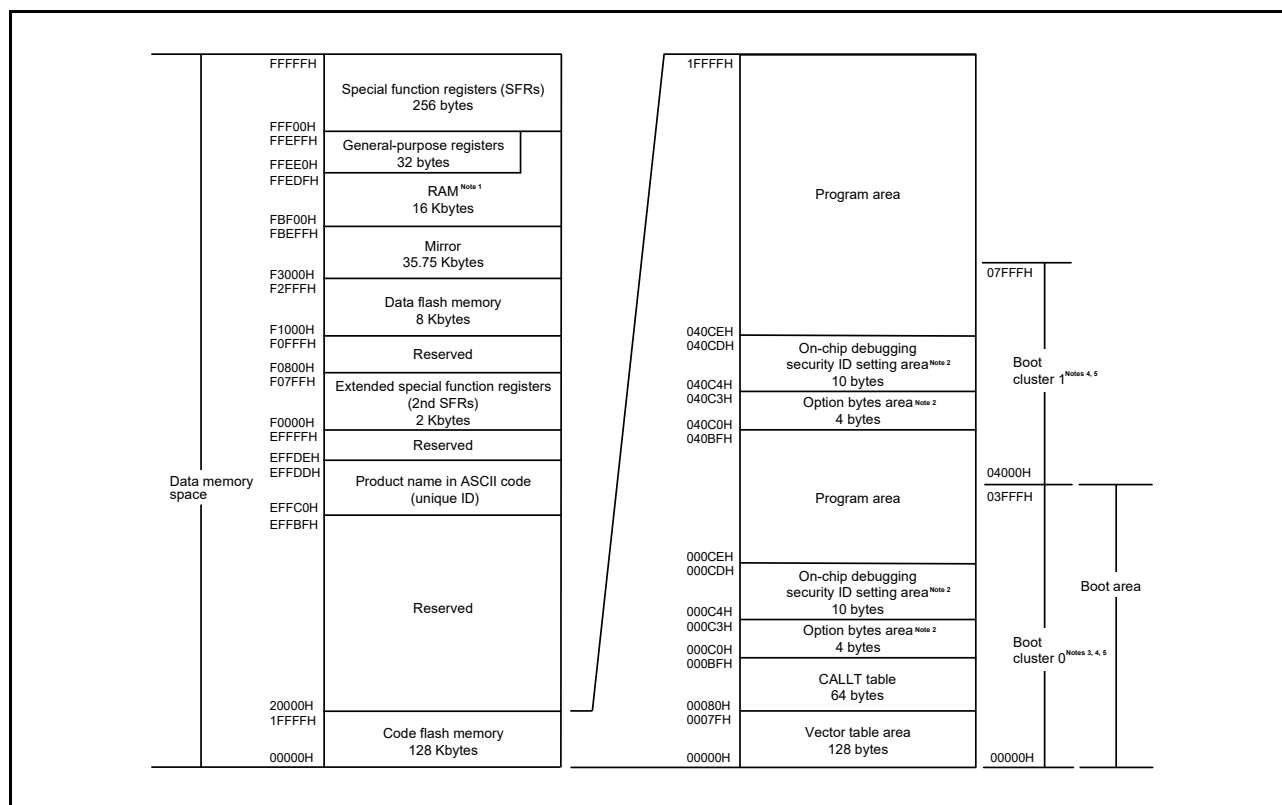
Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **38.7 Boot Swapping**.

Note 5. This applies when 16 Kbytes is selected for the boot cluster size.

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 **32.3.4 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 Addresses and Block Numbers in Flash Memory**.

Figure 3 - 2 Memory Map (R7F100LxG (x = F, G, J, L, M, P))



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

Note 2. When boot swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes and on-chip debugging security IDs to the corresponding addresses according to the selected size of the boot cluster as shown below.

- 32 Kbytes: Set the option bytes to 000C0H to 000C3H and 080C0H to 080C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 080C4H to 080CDH.
- 16 Kbytes: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 040C4H to 040CDH.
- 8 Kbytes: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 020C4H to 020CDH.
- 4 Kbytes: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
- 2 Kbytes: Set the option bytes to 000C0H to 000C3H and 008C0H to 008C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 008C4H to 008CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **38.10 Security Settings**.

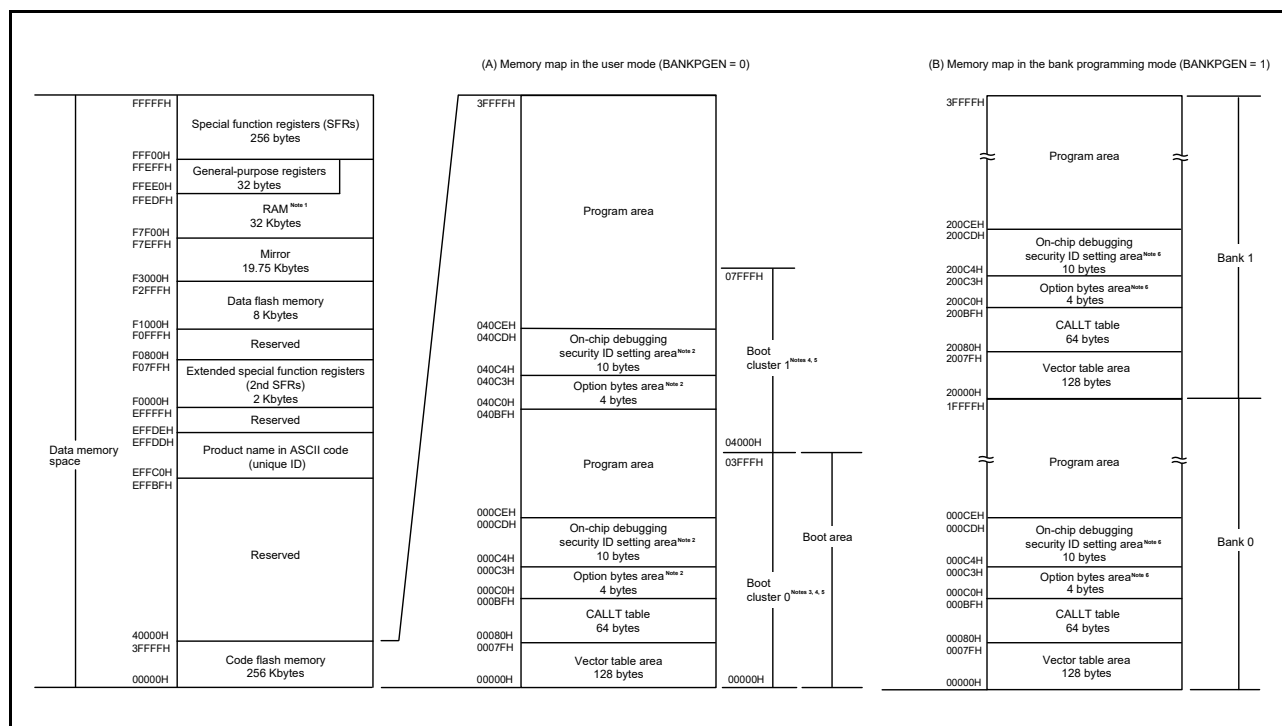
Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **38.7 Boot Swapping**.

Note 5. This applies when 16 Kbytes is selected for the boot cluster size.

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 **32.3.4 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 Addresses and Block Numbers in Flash Memory**.

Figure 3 - 3 Memory Map (R7F100LxJ (x = F, G, J, L, M, P))



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

Note 2. When boot swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes and on-chip debugging security IDs to the corresponding addresses according to the selected size of the boot cluster as shown below.

- 64 Kbytes: Set the option bytes to 000C0H to 000C3H and 100C0H to 100C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 100C4H to 100CDH.
- 32 Kbytes: Set the option bytes to 000C0H to 000C3H and 080C0H to 080C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 080C4H to 080CDH.
- 16 Kbytes: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 040C4H to 040CDH.
- 8 Kbytes: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 020C4H to 020CDH.
- 4 Kbytes: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
- 2 Kbytes: Set the option bytes to 000C0H to 000C3H and 008C0H to 008C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 008C4H to 008CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **38.10 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **38.7 Boot Swapping**.

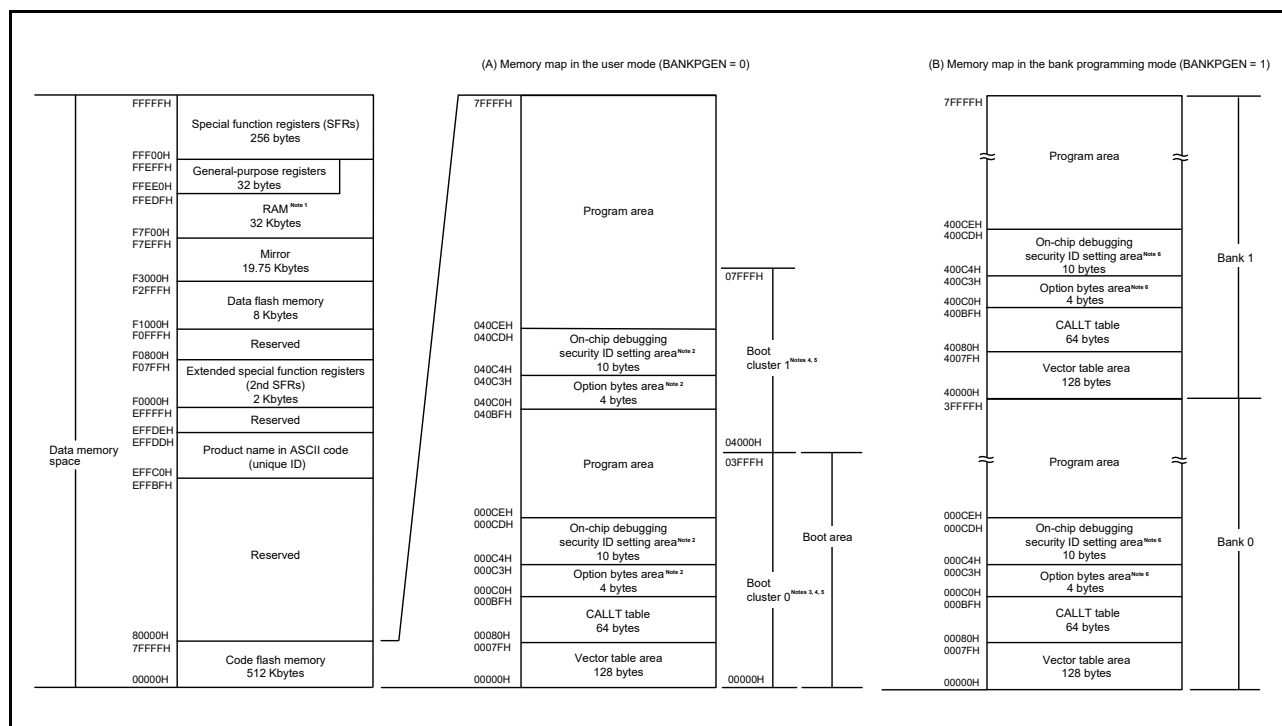
Note 5. This applies when 16 Kbytes is selected for the boot cluster size.

Note 6. When bank swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When bank swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 200C0H to 200C3H, and the on-chip debugging security IDs to 000C4H to 000CDH and 200C4H to 200CDH.

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 **32.3.4 RAM parity error detection**.

Remark The code flash memory area is divided into blocks, with each block being 1 Kbyte. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 4 Memory Map (R7F100LxL (x = F, G, J, L, M, P))



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

Note 2. When boot swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes and on-chip debugging security IDs to the corresponding addresses according to the selected size of the boot cluster as shown below.

- 128 Kbytes: Set the option bytes to 000C0H to 000C3H and 200C0H to 200C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 200C4H to 200CDH.
- 64 Kbytes: Set the option bytes to 000C0H to 000C3H and 100C0H to 100C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 100C4H to 100CDH.
- 32 Kbytes: Set the option bytes to 000C0H to 000C3H and 080C0H to 080C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 080C4H to 080CDH.
- 16 Kbytes: Set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 040C4H to 040CDH.
- 8 Kbytes: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 020C4H to 020CDH.
- 4 Kbytes: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
- 2 Kbytes: Set the option bytes to 000C0H to 000C3H and 008C0H to 008C3H. Set the on-chip debugging security IDs to 000C4H to 000CDH and 008C4H to 008CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **38.10 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **38.7 Boot Swapping**.

Note 5. This applies when 16 Kbytes is selected for the boot cluster size.

Note 6. When bank swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the option bytes to 000C0H to 000C3H, and the on-chip debugging security IDs to 000C4H to 000CDH. When bank swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 400C0H to 400C3H, and the on-chip debugging security IDs to 000C4H to 000CDH and 400C4H to 400CDH.

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 **32.3.4 RAM parity error detection**.

(Remark is listed on the next page.)

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 Addresses and Block Numbers in Flash Memory**.

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 (1/2)

Address	Block Number	Address	Block Number	Address	Block Number	Address	Block Number
00000H to 007FFH	000H	10000H to 107FFH	020H	20000H to 207FFH	040H	30000H to 307FFH	060H
00800H to 00FFFH	001H	10800H to 10FFFH	021H	20800H to 20FFFH	041H	30800H to 30FFFH	061H
01000H to 017FFH	002H	11000H to 117FFH	022H	21000H to 217FFH	042H	31000H to 317FFH	062H
01800H to 01FFFH	003H	11800H to 11FFFH	023H	21800H to 21FFFH	043H	31800H to 31FFFH	063H
02000H to 027FFH	004H	12000H to 127FFH	024H	22000H to 227FFH	044H	32000H to 327FFH	064H
02800H to 02FFFH	005H	12800H to 12FFFH	025H	22800H to 22FFFH	045H	32800H to 32FFFH	065H
03000H to 037FFH	006H	13000H to 137FFH	026H	23000H to 237FFH	046H	33000H to 337FFH	066H
03800H to 03FFFH	007H	13800H to 13FFFH	027H	23800H to 23FFFH	047H	33800H to 33FFFH	067H
04000H to 047FFH	008H	14000H to 147FFH	028H	24000H to 247FFH	048H	34000H to 347FFH	068H
04800H to 04FFFH	009H	14800H to 14FFFH	029H	24800H to 24FFFH	049H	34800H to 34FFFH	069H
05000H to 057FFH	00AH	15000H to 157FFH	02AH	25000H to 257FFH	04AH	35000H to 357FFH	06AH
05800H to 05FFFH	00BH	15800H to 15FFFH	02BH	25800H to 25FFFH	04BH	35800H to 35FFFH	06BH
06000H to 067FFH	00CH	16000H to 167FFH	02CH	26000H to 267FFH	04CH	36000H to 367FFH	06CH
06800H to 06FFFH	00DH	16800H to 16FFFH	02DH	26800H to 26FFFH	04DH	36800H to 36FFFH	06DH
07000H to 077FFH	00EH	17000H to 177FFH	02EH	27000H to 277FFH	04EH	37000H to 377FFH	06EH
07800H to 07FFFH	00FH	17800H to 17FFFH	02FH	27800H to 27FFFH	04FH	37800H to 37FFFH	06FH
08000H to 087FFH	010H	18000H to 187FFH	030H	28000H to 287FFH	050H	38000H to 387FFH	070H
08800H to 08FFFH	011H	18800H to 18FFFH	031H	28800H to 28FFFH	051H	38800H to 38FFFH	071H
09000H to 097FFH	012H	19000H to 197FFH	032H	29000H to 297FFH	052H	39000H to 397FFH	072H
09800H to 09FFFH	013H	19800H to 19FFFH	033H	29800H to 29FFFH	053H	39800H to 39FFFH	073H
0A000H to 0A7FFH	014H	1A000H to 1A7FFH	034H	2A000H to 2A7FFH	054H	3A000H to 3A7FFH	074H
0A800H to 0AFFFH	015H	1A800H to 1AFFFH	035H	2A800H to 2AFFFH	055H	3A800H to 3AFFFH	075H
0B000H to 0B7FFH	016H	1B000H to 1B7FFH	036H	2B000H to 2B7FFH	056H	3B000H to 3B7FFH	076H
0B800H to 0BFFFH	017H	1B800H to 1BFFFH	037H	2B800H to 2BFFFH	057H	3B800H to 3BFFFH	077H
0C000H to 0C7FFH	018H	1C000H to 1C7FFH	038H	2C000H to 2C7FFH	058H	3C000H to 3C7FFH	078H
0C800H to 0CFFFH	019H	1C800H to 1CFFFH	039H	2C800H to 2CFFFH	059H	3C800H to 3CFFFH	079H
0D000H to 0D7FFH	01AH	1D000H to 1D7FFH	03AH	2D000H to 2D7FFH	05AH	3D000H to 3D7FFH	07AH
0D800H to 0DFFFH	01BH	1D800H to 1DFFFH	03BH	2D800H to 2DFFFH	05BH	3D800H to 3DFFFH	07BH
0E000H to 0E7FFH	01CH	1E000H to 1E7FFH	03CH	2E000H to 2E7FFH	05CH	3E000H to 3E7FFH	07CH
0E800H to 0EFFFH	01DH	1E800H to 1EFFFH	03DH	2E800H to 2EFFFH	05DH	3E800H to 3EFFFH	07DH
0F000H to 0F7FFH	01EH	1F000H to 1F7FFH	03EH	2F000H to 2F7FFH	05EH	3F000H to 3F7FFH	07EH
0F800H to 0FFFFH	01FH	1F800H to 1FFFFH	03FH	2F800H to 2FFFFH	05FH	3F800H to 3FFFFH	07FH

Remark R7F100LxE (x = F, G, J, L): Block numbers 000H to 01FH
R7F100LxG (x = F, G, J, L, M, P): Block numbers 000H to 03FH
R7F100LxJ (x = F, G, J, L, M, P): Block numbers 000H to 07FH

Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory (2/2)

Address	Block Number	Address	Block Number	Address	Block Number	Address	Block Number
40000H to 407FFH	080H	50000H to 507FFH	0A0H	60000H to 607FFH	0C0H	70000H to 707FFH	0E0H
40800H to 40FFFH	081H	50800H to 50FFFH	0A1H	60800H to 60FFFH	0C1H	70800H to 70FFFH	0E1H
41000H to 417FFH	082H	51000H to 517FFH	0A2H	61000H to 617FFH	0C2H	71000H to 717FFH	0E2H
41800H to 41FFFH	083H	51800H to 51FFFH	0A3H	61800H to 61FFFH	0C3H	71800H to 71FFFH	0E3H
42000H to 427FFH	084H	52000H to 527FFH	0A4H	62000H to 627FFH	0C4H	72000H to 727FFH	0E4H
42800H to 42FFFH	085H	52800H to 52FFFH	0A5H	62800H to 62FFFH	0C5H	72800H to 72FFFH	0E5H
43000H to 437FFH	086H	53000H to 537FFH	0A6H	63000H to 637FFH	0C6H	73000H to 737FFH	0E6H
43800H to 43FFFH	087H	53800H to 53FFFH	0A7H	63800H to 63FFFH	0C7H	73800H to 73FFFH	0E7H
44000H to 447FFH	088H	54000H to 547FFH	0A8H	64000H to 647FFH	0C8H	74000H to 747FFH	0E8H
44800H to 44FFFH	089H	54800H to 54FFFH	0A9H	64800H to 64FFFH	0C9H	74800H to 74FFFH	0E9H
45000H to 457FFH	08AH	55000H to 557FFH	0AAH	65000H to 657FFH	0CAH	75000H to 757FFH	0EAH
45800H to 45FFFH	08BH	55800H to 55FFFH	0ABH	65800H to 65FFFH	0CBH	75800H to 75FFFH	0EBH
46000H to 467FFH	08CH	56000H to 567FFH	0ACH	66000H to 667FFH	0CCH	76000H to 767FFH	0ECH
46800H to 46FFFH	08DH	56800H to 56FFFH	0ADH	66800H to 66FFFH	0CDH	76800H to 76FFFH	0EDH
47000H to 477FFH	08EH	57000H to 577FFH	0AEH	67000H to 677FFH	0CEH	77000H to 777FFH	0EEH
47800H to 47FFFH	08FH	57800H to 57FFFH	0AFH	67800H to 67FFFH	0CFH	77800H to 77FFFH	0EFH
48000H to 487FFH	090H	58000H to 587FFH	0B0H	68000H to 687FFH	0D0H	78000H to 787FFH	0F0H
48800H to 48FFFH	091H	58800H to 58FFFH	0B1H	68800H to 68FFFH	0D1H	78800H to 78FFFH	0F1H
49000H to 497FFH	092H	59000H to 597FFH	0B2H	69000H to 697FFH	0D2H	79000H to 797FFH	0F2H
49800H to 49FFFH	093H	59800H to 59FFFH	0B3H	69800H to 69FFFH	0D3H	79800H to 79FFFH	0F3H
4A000H to 4A7FFH	094H	5A000H to 5A7FFH	0B4H	6A000H to 6A7FFH	0D4H	7A000H to 7A7FFH	0F4H
4A800H to 4AFFFH	095H	5A800H to 5AFFFH	0B5H	6A800H to 6AFFFH	0D5H	7A800H to 7AFFFH	0F5H
4B000H to 4B7FFH	096H	5B000H to 5B7FFH	0B6H	6B000H to 6B7FFH	0D6H	7B000H to 7B7FFH	0F6H
4B800H to 4BFFFH	097H	5B800H to 5BFFFH	0B7H	6B800H to 6BFFFH	0D7H	7B800H to 7BFFFH	0F7H
4C000H to 4C7FFH	098H	5C000H to 5C7FFH	0B8H	6C000H to 6C7FFH	0D8H	7C000H to 7C7FFH	0F8H
4C800H to 4CFFFH	099H	5C800H to 5CFFFH	0B9H	6C800H to 6CFFFH	0D9H	7C800H to 7CFFFH	0F9H
4D000H to 4D7FFH	09AH	5D000H to 5D7FFH	0BAH	6D000H to 6D7FFH	0DAH	7D000H to 7D7FFH	0FAH
4D800H to 4DFFFH	09BH	5D800H to 5DFFFH	0BBH	6D800H to 6DFFFH	0DBH	7D800H to 7DFFFH	0FBH
4E000H to 4E7FFH	09CH	5E000H to 5E7FFH	0BCH	6E000H to 6E7FFH	0DCH	7E000H to 7E7FFH	0FCH
4E800H to 4EFFFH	09DH	5E800H to 5EFFFH	0BDH	6E800H to 6EFFFH	0DDH	7E800H to 7EFFFH	0FDH
4F000H to 4F7FFH	09EH	5F000H to 5F7FFH	0BEH	6F000H to 6F7FFH	0DEH	7F000H to 7F7FFH	0FEH
4F800H to 4FFFFH	09FH	5F800H to 5FFFFH	0BFH	6F800H to 6FFFFH	0DFH	7F800H to 7FFFFH	0FFH

Remark R7F100LxL (x = F, G, J, L, M, P): Block numbers 000H to 0FFH

3.1.1 Internal program memory space

The internal program memory holds the program and table data. The RL78/L23 products incorporate the internal ROM (flash memory) with the capacity shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F100LxE (x = F, G, J, L)	Flash memory	65536 × 8 bits (00000H to 0FFFFH)
R7F100LxG (x = F, G, J, L, M, P)		131072 × 8 bits (00000H to 1FFFFH)
R7F100LxJ (x = F, G, J, L, M, P)		262144 × 8 bits (00000H to 3FFFFH)
R7F100LxL (x = F, G, J, L, M, P)		524288 × 8 bits (00000H to 7FFFFH)

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. The eight lower-order bits and eight higher-order bits of the 16-bit address are respectively stored at even and odd addresses. When boot swapping is in use, set a vector table also at the following addresses according to the selected boot cluster size (Kbytes): At 20000H to 2007FH for 128 Kbytes, 10000H to 1007FH for 64 Kbytes, 08000H to 0807FH for 32 Kbytes, 04000H to 0407FH for 16 Kbytes, 02000H to 0207FH for 8 Kbytes, 01000H to 0107FH for 4 Kbytes, and 00800H to 0087FH for 2 Kbytes. When bank swapping is in use, set a vector table also at 20000H to 2007FH when 256 Kbytes is selected and at 40000H to 4007FH when 512 Kbytes is selected. **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 **38.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

Table 3 - 3 Vector Table (1/2)

Vector Table Address	Interrupt Source	100-pin	80-pin	64-pin	52-pin	48-pin	44-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	✓	✓	✓	✓	✓	Note 1
	INTIT01	✓	✓	✓	✓	✓	
00016H	INTSR2/INTCSI21/INTIIC21	✓	✓	✓	✓	✓	✓
	INTIT41	✓	✓	✓	✓	✓	✓
00018H	INTSRE2	✓	✓	✓	✓	✓	✓
	INTIT20	✓	✓	✓	✓	✓	✓
0001AH	INTELCL	✓	✓	✓	✓	✓	✓
0001CH	INTSMSE	✓	✓	✓	✓	✓	✓
0001EH	INTST0/INTCSI00/INTIIC00	✓	✓	✓	✓	✓	✓
	INTIT21	✓	✓	✓	✓	✓	✓
00020H	INTTM00	✓	✓	✓	✓	✓	✓
00022H	INTSRE0	✓	✓	✓	✓	✓	✓
	INTTM01H	✓	✓	✓	✓	✓	✓
	INTIT30	✓	✓	✓	✓	✓	✓
00024H	INTST1/INTCSI10/INTIIC10	✓	✓	✓	✓	✓	✓
	INTIT11	✓	✓	✓	✓	✓	✓
00026H	INTSR1/INTCSI11/INTIIC11	✓	✓	✓	✓	✓	✓
	INTIT61	✓	✓	✓	✓	✓	✓
00028H	INTSRE1	✓	✓	✓	✓	✓	✓
	INTTM03H	✓	✓	✓	✓	✓	✓
	INTIT40	✓	✓	✓	✓	✓	✓
0002AH	INTIICA0	✓	✓	✓	✓	✓	✓
0002CH	INTSR0/INTCSI01/INTIIC01	✓	✓	✓	✓	✓	✓
	INTIT51	✓	✓	✓	✓	✓	✓
0002EH	INTTM01	✓	✓	✓	✓	✓	✓
00030H	INTTM02	✓	✓	✓	✓	✓	✓
00032H	INTTM03	✓	✓	✓	✓	✓	✓
00034H	INTAD	✓	✓	✓	✓	✓	✓
00036H	INTRTC	✓	✓	✓	✓	✓	✓
00038H	INTITL	✓	✓	✓	✓	✓	✓
0003AH	INTKR	✓	✓	✓	✓	✓	✓
0003CH	INTST3/INTCSI30/INTIIC30	✓	✓	✓	✓	Note 2	Note 2
	INTIT31	✓	✓	✓	✓		

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
0003EH	INTSR3/INTCSI31/INTIIC31	✓	✓	✓	Note 3	—	—
	INTIT71	✓	✓	✓	✓	✓	✓
00040H	INTIT00	✓	✓	✓	✓	✓	✓
00042H	INTTM04	✓	✓	✓	✓	✓	✓
00044H	INTTM05	✓	✓	✓	✓	✓	✓
	INTURE3	✓	✓	—	—	—	—
00046H	INTTM06	✓	✓	✓	✓	✓	✓
	INTIT50	✓	✓	✓	✓	✓	✓
00048H	INTTM07	✓	✓	✓	✓	✓	✓
	INTIT60	✓	✓	✓	✓	✓	✓
0004AH	INTP6	✓	✓	✓	✓	✓	✓
0004CH	INTP7	✓	✓	✓	✓	✓	✓
0004EH	INTIT10	✓	✓	✓	✓	✓	✓
00050H	INTOSDC	✓	✓	✓	✓	✓	✓
00052H	INTFL	✓	✓	✓	✓	✓	✓
00054H	INTCMP0	✓	✓	✓	✓	✓	✓
00056H	INTCMP1	✓	✓	✓	✓	—	—
00058H	INTURE0	✓	✓	✓	✓	✓	✓
	INTIT70	✓	✓	✓	✓	✓	✓
0005AH	INTURE1	✓	✓	✓	✓	✓	✓
	INTTMKB1	✓	✓	✓	✓	✓	✓
0005CH	INTURE2	✓	✓	✓	✓	✓	—
	INTTMKB2	✓	✓	✓	✓	✓	✓
0005EH	INTSRE3	✓	✓	✓	✓	—	—
	INTTRJ1	✓	✓	✓	✓	✓	✓
00060H	INTCTSUWR	✓	✓	✓	✓	✓	✓
00062H	INTIICA1	✓	✓	✓	✓	✓	✓
00064H	INTCTSURD	✓	✓	✓	✓	✓	✓
00066H	INTCTSUFN	✓	✓	✓	✓	✓	✓
00068H	INTTRJ0	✓	✓	✓	✓	✓	✓
0006AH	INTUT0	✓	✓	✓	✓	✓	✓
0006CH	INTUR0	✓	✓	✓	✓	✓	✓
0006EH	INTUT1	✓	✓	✓	✓	✓	✓
00070H	INTUR1	✓	✓	✓	✓	✓	✓
00072H	INTUT2	✓	✓	✓	✓	✓	—
00074H	INTUR2	✓	✓	✓	✓	✓	—
00076H	INTUT3	✓	✓	—	—	—	—
00078H	INTUR3	✓	✓	—	—	—	—
0007AH	INTTMKB0	✓	✓	✓	✓	✓	✓
0007CH	INTEXSD	✓	✓	✓	✓	✓	✓
0007EH	BRK	✓	✓	✓	✓	✓	✓

Note 1. INTST2 is only present in this product.

Note 2. INTIIC30 is only present in this product.

Note 3. INTSR3 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). When boot swapping is in use, set a CALLT instruction table also at the following addresses according to the selected boot cluster size (Kbytes): At 20080H to 200BFH for 128 Kbytes, 10080H to 100BFH for 64 Kbytes, 08080H to 080BFH for 32 Kbytes, 04080H to 040BFH for 16 Kbytes, 02080H to 020BFH for 8 Kbytes, 01080H to 010BFH for 4 Kbytes, and 00880H to 008BFH for 2 Kbytes. When bank swapping is in use, set a CALLT instruction table also at 20080H to 200BFH when 256 Kbytes is selected and at 40080H to 400BFH when 512 Kbytes is selected.

3. Option bytes area

A 4-byte area of 000C0H to 000C3H can be used as an option bytes area. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option byte also at the following addresses according to the selected boot cluster size (Kbytes): At 200C0H to 200C3H for 128 Kbytes, 100C0H to 100C3H for 64 Kbytes, 080C0H to 080C3H for 32 Kbytes, 040C0H to 040C3H for 16 Kbytes, 020C0H to 020C3H for 8 Kbytes, 010C0H to 010C3H for 4 Kbytes, and 008C0H to 008C3H for 2 Kbytes. When bank swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the option byte also at 200C0H to 200C3H when 256 Kbytes is selected and at 400C0H to 400C3H when 512 Kbytes is selected. For details, see **Section 37 Option Bytes**.

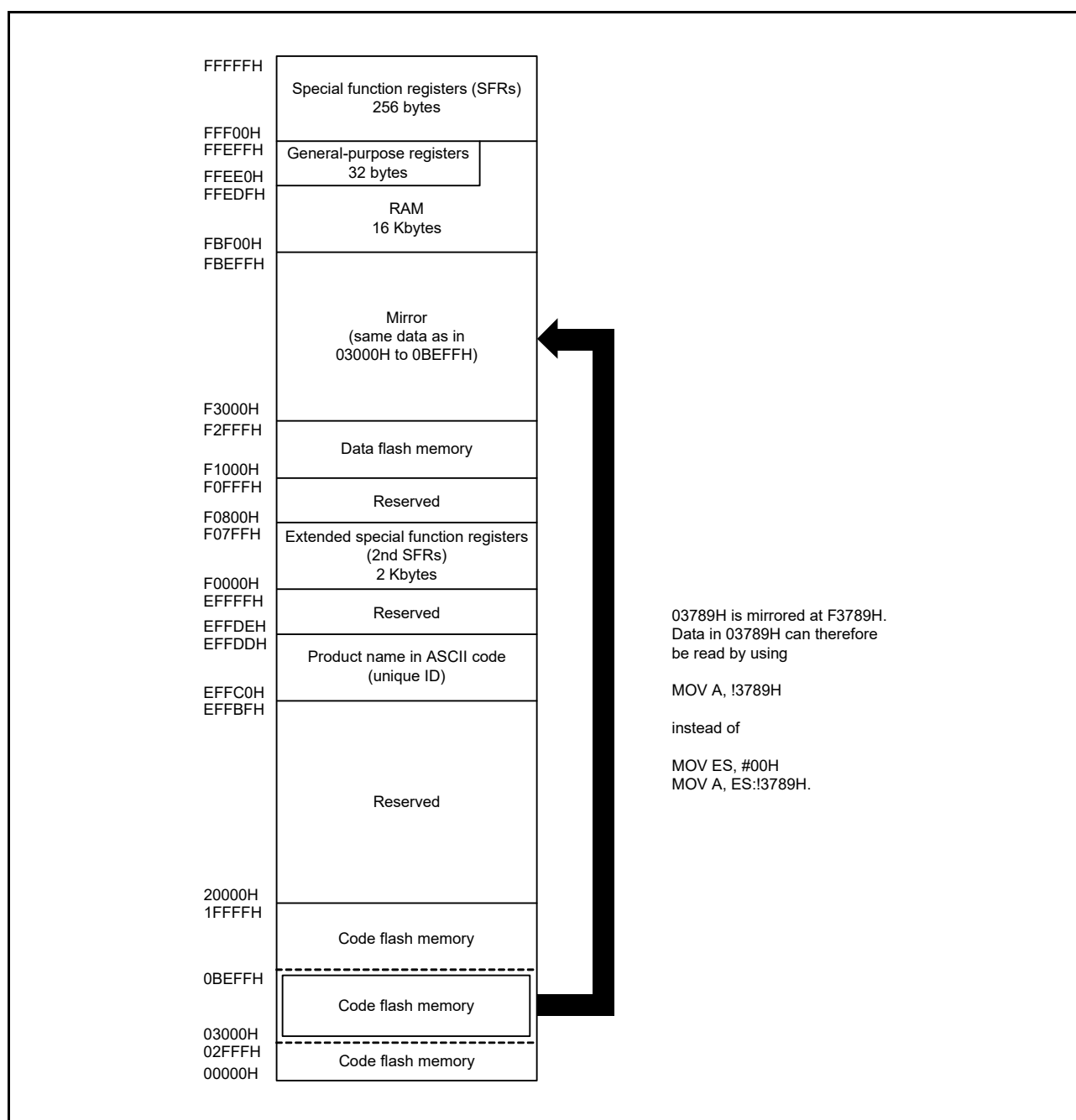
4. On-chip debugging security ID setting area

A 10-byte area of 000C4H to 000CDH and 040C4H to 040CDH can be used as an on-chip debugging security ID setting area. When boot swapping is not in use (the setting of the BTFLG bit in the FLSEC register is 1), set the on-chip debugging security ID of 10 bytes at 000C4H to 000CDH. When boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the on-chip debugging security ID of 10 bytes at the following addresses according to the selected boot cluster size (Kbytes): At 000C4H to 000CDH and 200C4H to 200CDH for 128 Kbytes, 000C4H to 000CDH and 100C4H to 100CDH for 64 Kbytes, 000C4H to 000CDH and 080C4H to 080CDH for 32 Kbytes, 000C4H to 000CDH and 040C4H to 040CDH for 16 Kbytes, 000C4H to 000CDH and 020C4H to 020CDH for 8 Kbytes, 000C4H to 000CDH and 010C4H to 010CDH for 4 Kbytes, and 000C4H to 000CDH and 008C4H to 008CDH for 2 Kbytes. When bank swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0, set the on-chip debugging security ID also at 000C4H to 000CDH and 200C4H to 200CDH when 256 Kbytes is selected and at 000C4H to 000CDH and 400C4H to 400CDH when 512 Kbytes is selected. For details, see **Section 39 On-chip Debugging**.

3.1.2 Mirror area

In products of the RL78/L23, the code flash area from 00000H to 0FFFFH or 10000H to 1FFFFH is mirrored at F0000H to FFFFFH. The setting of the effective bit of the processor mode control register (PMC) determines which range of the code flash area is to be 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 a 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.

Example R7F100LxG (x = F, G, J, L, M, P) (Flash memory: 128 Kbytes, RAM: 16 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 - 5 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	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

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 03000H to 07FFFH are mirrored at F3000H to F7FFFH after boot swapping.

3.1.3 Internal data memory space

The RL78/L23 products incorporate the RAM with the capacity shown below.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R7F100LxE (x = F, G, J, L)	16384 × 8 bits (FBF00H to FFEFFH)
R7F100LxG (x = F, G, J, L, M, P)	16384 × 8 bits (FBF00H to FFEFFH)
R7F100LxJ (x = F, G, J, L, M, P)	32768 × 8 bits (F7F00H to FFEFFH)
R7F100LxL (x = F, G, J, L, M, P)	32768 × 8 bits (F7F00H 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. Some areas in the internal memory can be placed in shutdown mode for low-power operation. Data cannot be retained in the areas placed in shutdown mode. For details, see **27.2.2 Memory power reduction control register (PSMCR)**.

Caution 1. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

<R> **Caution 2.** The use of the RAM area in the following products is prohibited when the trace function of on-chip debugging is in use.

R7F100LxE (x = F, G, J, L): FC300H to FC6FFH

R7F100LxG (x = F, G, J, L, M, P): FC300H to FC6FFH

R7F100LxJ, R7F100LxL (x = F, G, J, L, M, P): F8300H to F86FFH

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 List of Special Function Registers (SFRs)** 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 List of 2nd Special Function Registers** in **3.2.5 Extended special function registers (2nd SFRs)**).

Caution 1. Do not access addresses to which 2nd SFRs are not assigned.

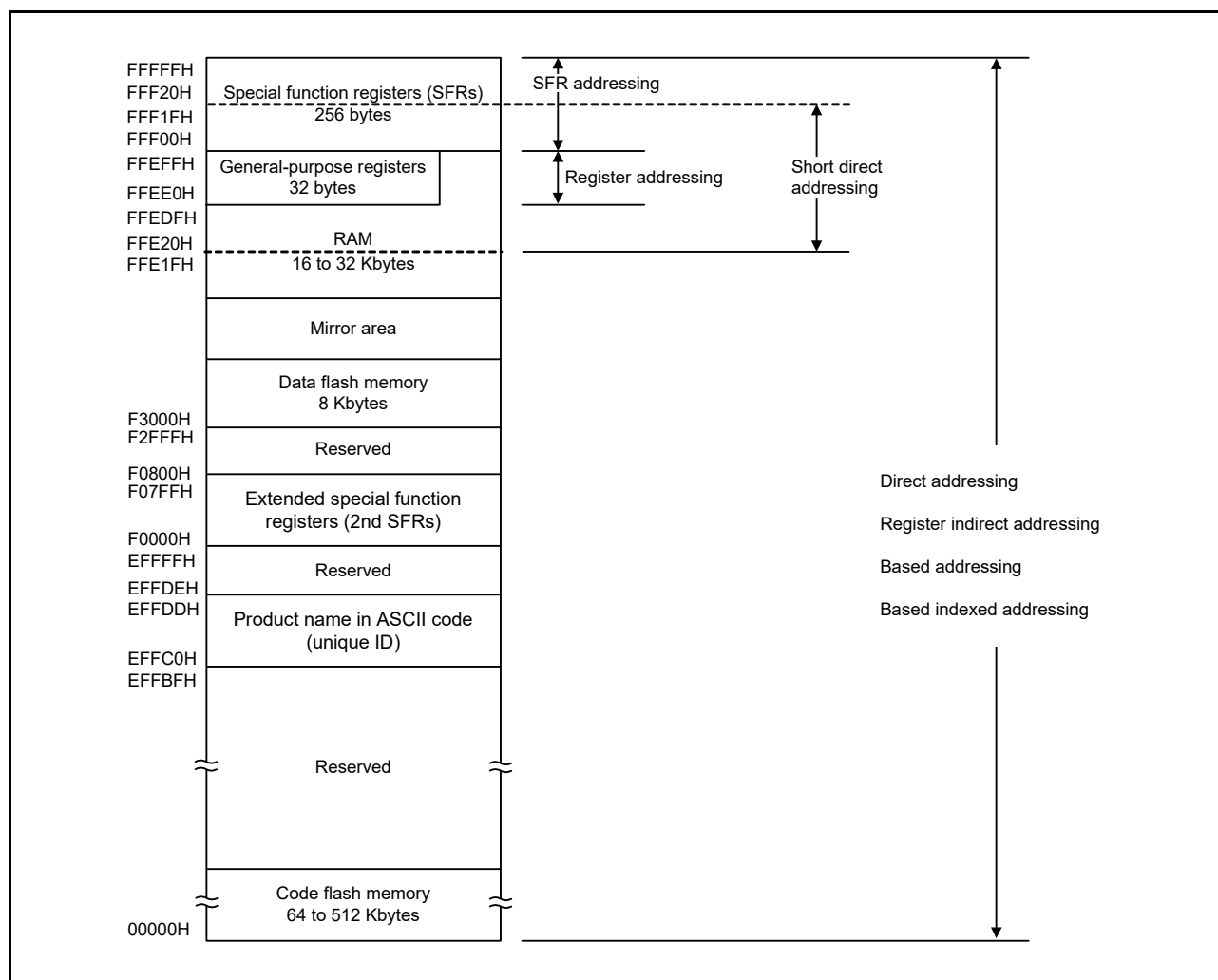
Caution 2. The CPU is placed in the wait state and does not proceed to the next instruction during access to any of the following registers allocated in the areas for the 2nd SFRs: Registers of the capacitive sensing unit (CTSU2La) allocated to F0500H to F0533H, registers of the true random number generator (TRNG) allocated to F0540H and F0542H, registers of the 8-bit interval timers allocated to F0550H to F0589H, timer RJ counter register 0 (TRJ0) allocated to F0598H and F0599H, and timer RJ counter register 1 (TRJ1) allocated to F059AH and F059BH. 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 any of the registers above: Registers of the 8-bit interval timers, the TRJn register of the timer RJn, registers of the capacitive sensing unit (CTSU2La), and registers of the true random number generator (TRNG).

Remark n: Channel number (n = 0, 1)

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/L23, 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 - 6** shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 6 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/L23 products incorporate the following processor registers.

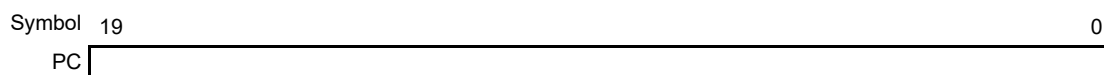
3.2.1 Control registers

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.

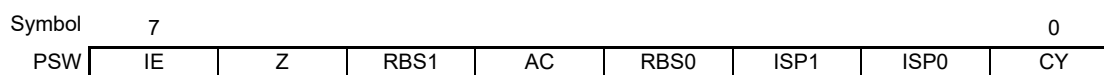
Figure 3 - 7 Format of Program Counter



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 - 8 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 to 0 upon DI instruction execution or interrupt acknowledgment and is set to 1 upon EI instruction execution.

b) Zero flag (Z)

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

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

This flag manages 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 **25.3.3 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 - 9 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.

Caution 3. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.

R7F100LxE (x = F, G, J, L): FC300H to FC6FFH

R7F100LxG (x = F, G, J, L, M, P): FC300H to FC6FFH

R7F100LxJ, R7F100LxL (x = F, G, J, L, M, P): F8300H to F86FFH

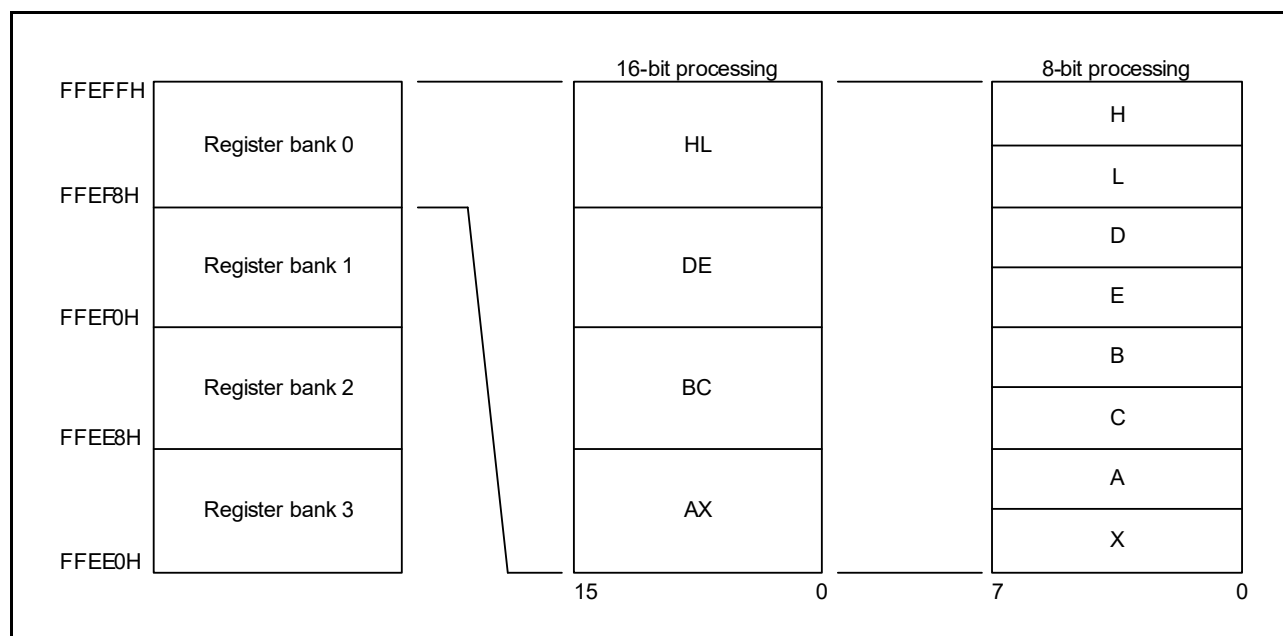
3.2.2 General-purpose registers

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 - 10 Configuration of General-Purpose Registers

a) Function name



3.2.3 ES and CS registers

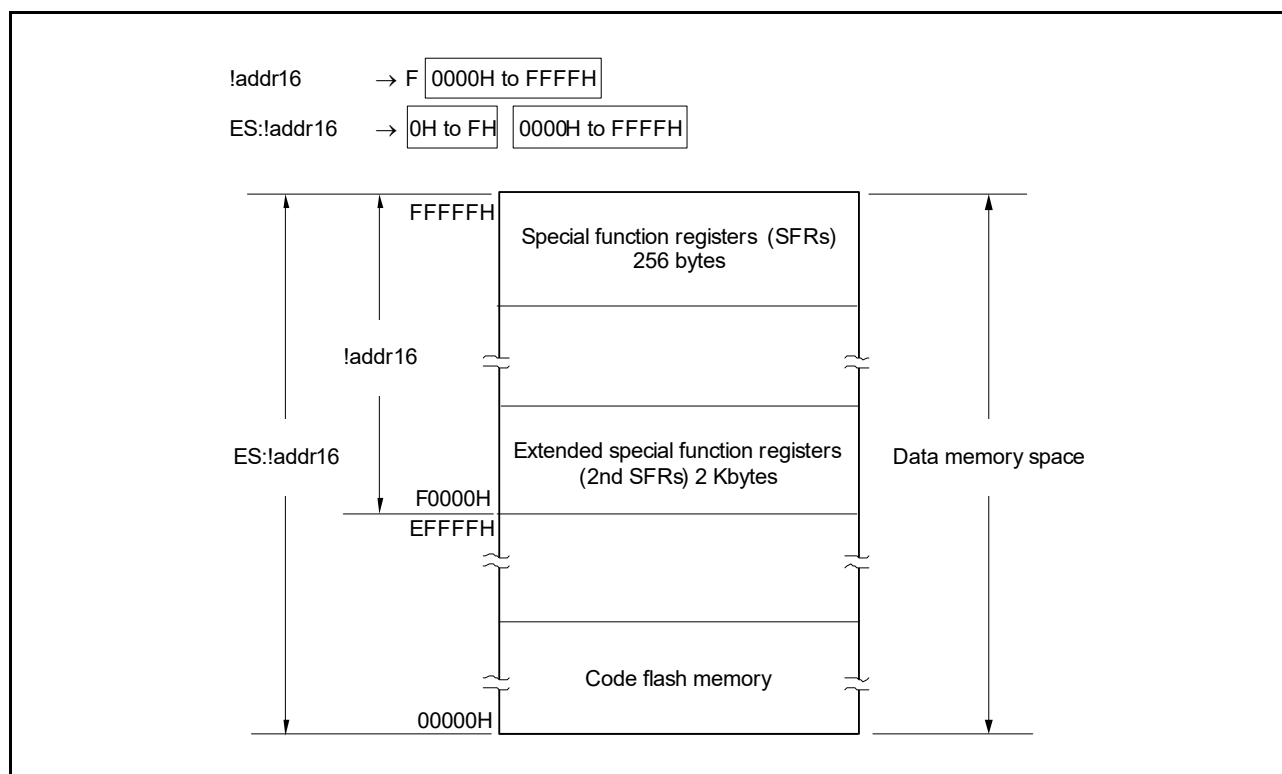
The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register indirect addressing), respectively. The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3 - 11 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 - 12 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each special function register (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 lists 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 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-bit	16-bit	
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
FFF08H	Port register 8	P8		R/W	✓	✓	—	00H
FFF09H	Port register 9	P9		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		—			—	—		
FFF14H	Serial data register 12	TXD3/ SIO30	SDR12	R/W	—	✓	✓	0000H
FFF15H		—			—	—		
FFF16H	Serial data register 13	RXD3/ SIO31	SDR13	R/W	—	✓	✓	0000H
FFF17H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	✓	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	✓	✓	00H
FFF1BH		TDR01H			—	✓		00H
FFF1EH	12-bit/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
FFF28H	Port mode register 8	PM8		R/W	✓	✓	—	FFH
FFF29H	Port mode register 9	PM9		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

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-bit	16-bit	
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	✓	✓	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	✓	✓	—	00H
FFF40H	LCD mode register 0	LCDM0		R/W	—	✓	—	00H
FFF41H	LCD mode register 1	LCDM1		R/W	✓	✓	—	00H
FFF42H	LCD clock control register 0	LCDC0		R/W	—	✓	—	00H
FFF43H	LCD boost level control register	VLCD		R/W	—	✓	—	04H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	✓	✓	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	✓	✓	0000H
FFF47H		—			—	—		
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
FFF54H	IICA shift register 1	IICA1		R/W	—	✓	—	00H
FFF55H	IICA status register 1	IICS1		R	✓	✓	—	00H
FFF56H	IICA flag register 1	IICF1		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

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-bit	16-bit	
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		IF3H		R/W	✓	✓		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		MK3H		R/W	✓	✓		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
FFFDBH		PR03H		R/W	✓	✓		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	✓	✓	✓	FFH
FFDDDH		PR12H		R/W	✓	✓		FFH
FFFDEH	Priority specification flag register 13	PR13L	PR13	R/W	✓	✓	✓	FFH
FFDFH		PR13H		R/W	✓	✓		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
FFFEDH		PR10H		R/W	✓	✓		FFH
FFFEH	Priority specification flag register 11	PR11L	PR11	R/W	✓	✓	✓	FFH
FFFEFH		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								
FFFEH	Processor mode control register	PMC		R/W	✓	✓	—	00H

Note 1. For the reset resources, see **Section 28 Reset Function**.

Note 2. The initial value depends on the source of the reset. See **30.3.1 Voltage detection register (LVIM)**.

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

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

3.2.5 Extended special function registers (2nd SFRs)

Unlike a general-purpose register, each extended special function register (2nd SFR) has a special function. 2nd 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 2nd SFR area, however, is 1 byte longer than an instruction that accesses the SFR area. 2nd 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 2nd 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 lists the 2nd SFRs. The meanings of items in the table are as follows.

- Symbol
This item indicates the address of a 2nd 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 2nd 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 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 2nd Special Function Registers (1/19)

Address	Extended Special Function Register (2nd 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
F0020H	12-bit/10-bit A/D conversion result register 0	—	ADCR0	R	—	—	✓	0000H
F0021H	8-bit A/D conversion result register 0	ADCR0H		R	—	✓	—	00H
F0030H	Pull-up resistor option register 0	PU0		R/W	✓	✓	—	00H
F0031H	Pull-up resistor option register 1	PU1		R/W	✓	✓	—	00H
F0032H	Pull-up resistor option register 2	PU2		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
F0036H	Pull-up resistor option register 6	PU6		R/W	✓	✓	—	00H
F0037H	Pull-up resistor option register 7	PU7		R/W	✓	✓	—	00H
F0038H	Pull-up resistor option register 8	PU8		R/W	✓	✓	—	00H
F0039H	Pull-up resistor option register 9	PU9		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
F0042H	Port input mode register 2	PIM2		R/W	✓	✓	—	00H
F0043H	Port input mode register 3	PIM3		R/W	✓	✓	—	00H
F0044H	Port input mode register 4	PIM4		R/W	✓	✓	—	00H
F0045H	Port input mode register 5	PIM5		R/W	✓	✓	—	00H
F0046H	Port input mode register 6	PIM6		R/W	✓	✓	—	00H
F0047H	Port input mode register 7	PIM7		R/W	✓	✓	—	00H
F0048H	Port input mode register 8	PIM8		R/W	✓	✓	—	00H
F0049H	Port input mode register 9	PIM9		R/W	✓	✓	—	00H
F004CH	Port input mode register 12	PIM12		R/W	✓	✓	—	00H
F004EH	Port input mode register 14	PIM14		R/W	✓	✓	—	00H
F0050H	Port output mode register 0	POM0		R/W	✓	✓	—	00H
F0051H	Port output mode register 1	POM1		R/W	✓	✓	—	00H
F0052H	Port output mode register 2	POM2		R/W	✓	✓	—	00H
F0053H	Port output mode register 3	POM3		R/W	✓	✓	—	00H
F0054H	Port output mode register 4	POM4		R/W	✓	✓	—	00H
F0055H	Port output mode register 5	POM5		R/W	✓	✓	—	00H
F0056H	Port output mode register 6	POM6		R/W	✓	✓	—	00H
F0057H	Port output mode register 7	POM7		R/W	✓	✓	—	00H
F0058H	Port output mode register 8	POM8		R/W	✓	✓	—	00H
F0059H	Port output mode register 9	POM9		R/W	✓	✓	—	00H
F005CH	Port output mode register 12	POM12		R/W	✓	✓	—	00H
F005DH	Port output mode register 13	POM13		R/W	✓	✓	—	00H
F005EH	Port output mode register 14	POM14		R/W	✓	✓	—	00H
F0061H	Port mode control A register 1	PMCA1		R/W	✓	✓	—	FFH
F0062H	Port mode control A register 2	PMCA2		R/W	✓	✓	—	FFH
F0064H	Port mode control A register 4	PMCA4		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

Table 3 - 6 List of 2nd Special Function Registers (2/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0075H	Timer input select register 1	TIS1	R/W	—	✓	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	✓	—	00H
F0079H	UART loopback select register	ULBS	R/W	✓	✓	—	00H
F007AH	Timer output select register	TOS	R/W	✓	✓	—	00H
F007BH	Port mode select register	PMS	R/W	✓	✓	—	00H
F0090H	Data flash control register	DFLCTL	R/W	✓	✓	—	00H
F009EH	Main oscillator oscillation mode select register	MODRV	R/W	—	✓	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	—	✓	—	Note 1
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	✓	—	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
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 Note 4
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
F00FCH	Peripheral enable register 2	PER2	R/W	✓	✓	—	00H
F00FDH	Peripheral reset control register 2	PRR2	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				—	—	—	

Table 3 - 6 List of 2nd Special Function Registers (3/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		—			—	—		
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		—			—	—		
F0144H	Serial status register 12	SSR12L	SSR12	R	—	✓	✓	0000H
F0145H		—			—	—		
F0146H	Serial status register 13	SSR13L	SSR13	R	—	✓	✓	0000H
F0147H		—			—	—		
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		—			—	—		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	—	✓	✓	0000H
F014DH		—			—	—		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	—	✓	✓	0000H
F014FH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	✓	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	✓	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	—	—	✓	0020H
F0155H								

Table 3 - 6 List of 2nd Special Function Registers (4/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0156H	Serial mode register 13	SMR13		R/W	—	—	✓	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	✓	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	✓	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	—	—	✓	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	—	—	✓	0087H
F015FH								
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		—			—	—		
F0178H	Serial standby control register 1	SSC1L	SSC1	R/W	—	✓	✓	0000H
F0179H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	✓	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	✓	FFFFH
F0183H								
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								

Table 3 - 6 List of 2nd Special Function Registers (5/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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	TSR00	R	—	✓	✓	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	✓	✓	0000H
F01A3H		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	✓	✓	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	✓	✓	0000H
F01A7H		—			—	—		
F01A8H	Timer status register 04	TSR04L	TSR04	R	—	✓	✓	0000H
F01A9H		—			—	—		
F01AAH	Timer status register 05	TSR05L	TSR05	R	—	✓	✓	0000H
F01ABH		—			—	—		
F01ACH	Timer status register 06	TSR06L	TSR06	R	—	✓	✓	0000H
F01ADH		—			—	—		
F01AEH	Timer status register 07	TSR07L	TSR07	R	—	✓	✓	0000H
F01AFH		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	✓	✓	✓	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	✓	✓	✓	0000H
F01B3H		—			—	—		
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	Memory power reduction control register	PSMCR		R/W	✓	✓	—	00H
F0217H	XT1 oscillator margin checking control register	SOMRG		R/W	—	✓	—	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

Table 3 - 6 List of 2nd Special Function Registers (6/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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
F0238H	IICA control register 10	IICCTL10	R/W	✓	✓	—	00H
F0239H	IICA control register 11	IICCTL11	R/W	✓	✓	—	00H
F023AH	IICA low-level width setting register 1	IICWL1	R/W	—	✓	—	FFH
F023BH	IICA high-level width setting register 1	IICWH1	R/W	—	✓	—	FFH
F023CH	Slave address register 1	SVA1	R/W	—	✓	—	00H
F0240H	Timer RJ control register 0	TRJCR0	R/W	—	✓	—	00H
F0241H	Timer RJ I/O control register 0	TRJIOC0	R/W	✓	✓	—	00H
F0242H	Timer RJ mode register 0	TRJMR0	R/W	✓	✓	—	00H
F0243H	Timer RJ event pin select register 0	TRJISR0	R/W	✓	✓	—	00H
F0244H	Timer RJ control register 1	TRJCR1	R/W	—	✓	—	00H
F0245H	Timer RJ I/O control register 1	TRJIOC1	R/W	✓	✓	—	00H
F0246H	Timer RJ mode register 1	TRJMR1	R/W	✓	✓	—	00H
F0247H	Timer RJ event pin select register 1	TRJISR1	R/W	✓	✓	—	00H
F025AH	External signal sampler control register 1	EXSDM1	R/W	—	✓	—	00H
F025BH	External signal sampler control register 0	EXSDM0	R/W	✓	✓	—	00H
F0260H	Port mode control T register 0	PMCT0	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
F0269H	Port mode control T register 9	PMCT9	R/W	✓	✓	—	00H
F026CH	Port mode control T register 12	PMCT12	R/W	✓	✓	—	00H
F0280H	Port mode control E register 0	PMCE0	R/W	✓	✓	—	00H
F0281H	Port mode control E register 1	PMCE1	R/W	✓	✓	—	00H
F0286H	Port mode control E register 6	PMCE6	R/W	✓	✓	—	00H
F0287H	Port mode control E register 7	PMCE7	R/W	✓	✓	—	00H
F0290H	LCD port function register 0	PFSEG0	R/W	✓	✓	—	0FH
F0291H	LCD port function register 1	PFSEG1	R/W	✓	✓	—	FFH
F0292H	LCD port function register 2	PFSEG2	R/W	✓	✓	—	FFH
F0293H	LCD port function register 3	PFSEG3	R/W	✓	✓	—	FFH
F0294H	LCD port function register 4	PFSEG4	R/W	✓	✓	—	EFH
F0295H	LCD port function register 5	PFSEG5	R/W	✓	✓	—	FFH
F0296H	LCD port function register 6	PFSEG6	R/W	✓	✓	—	FFH
F0297H	LCD port function register 7	PFSEG7	R/W	✓	✓	—	FFH
F0298H	LCD input switch control register	ISCLCD	R/W	✓	✓	—	00H
F0299H	LCD mode register 2	LCDM2	R/W	✓	✓	—	00H
F029EH	A/D converter mode extension register 1	ADM1EX	R/W	✓	✓	—	00H
F02A0H	Output current select register 0	CCS0	R/W	—	✓	—	00H
F02A4H	Output current select register 4	CCS4	R/W	—	✓	—	00H
F02A5H	Output current select register 5	CCS5	R/W	—	✓	—	00H
F02A6H	Output current select register 6	CCS6	R/W	—	✓	—	00H
F02A7H	Output current select register 7	CCS7	R/W	—	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (7/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02A8H	Output current control enable register	CCDE	R/W	✓	✓	—	00H
F02A9H	40-mA port output control register	PTDC	R/W	✓	✓	—	00H
F02AAH	Port function output enable register 0	PFOE0	R/W	✓	✓	—	FFH
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
F02B2H	Port digital input disable register 2	PDIDIS2	R/W	✓	✓	—	00H
F02B3H	Port digital input disable register 3	PDIDIS3	R/W	✓	✓	—	00H
F02B4H	Port digital input disable register 4	PDIDIS4	R/W	✓	✓	—	00H
F02B5H	Port digital input disable register 5	PDIDIS5	R/W	✓	✓	—	00H
F02B6H	Port digital input disable register 6	PDIDIS6	R/W	✓	✓	—	00H
F02B7H	Port digital input disable register 7	PDIDIS7	R/W	✓	✓	—	00H
F02B8H	Port digital input disable register 8	PDIDIS8	R/W	✓	✓	—	00H
F02B9H	Port digital input disable register 9	PDIDIS9	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
F02BEH	Port digital input disable register 14	PDIDIS14	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/80H
F02CBH	Flash memory sequencer status register H	FSASTH	R	✓	✓	—	00H/04H
F02CCH	Flash write buffer register L	FLWL	R/W	—	—	✓	0000H
F02CEH	Flash write buffer register H	FLWH	R/W	—	—	✓	0000H
F02D0H	Oscillation stop detection control register	OSDC	R/W	—	—	✓	0FFFH
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
F02EDH	DTC activation enable register 5	DTCEN5	R/W	✓	✓	—	00H
F02EEH	DTC activation enable register 6	DTCEN6	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
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
F0330H	D/A conversion value setting register 0	DACS0	R/W	—	✓	—	00H
F0331H	D/A conversion value setting register 1	DACS1	R/W	—	✓	—	00H
F0332H	D/A converter mode register 0	DAM0	R/W	✓	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (8/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0338H	D/A conversion value setting register 2	DACS2	R/W	—	✓	—	00H
F033AH	D/A converter mode register 1	DAM1	R/W	✓	✓	—	00H
F0340H	Comparator mode setting register	COMPMDR	R/W	✓	✓	—	00H
F0341H	Comparator filter control register	COMPFIR	R/W	✓	✓	—	00H
F0342H	Comparator output control register	COMPOCR	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
F0370H	External interrupt control register 0	INTPCTL0	R/W	—	✓	—	00H
F0371H	External interrupt control register 1	INTPCTL1	R/W	—	✓	—	00H
F0372H	External interrupt control register 2	INTPCTL2	R/W	—	✓	—	00H
F0373H	External interrupt control register 3	INTPCTL3	R/W	—	✓	—	00H
F0374H	External interrupt control register 4	INTPCTL4	R/W	—	✓	—	00H
F0375H	External interrupt control register 5	INTPCTL5	R/W	—	✓	—	00H
F0376H	External interrupt control register 6	INTPCTL6	R/W	—	✓	—	00H
F0377H	External interrupt control register 7	INTPCTL7	R/W	—	✓	—	00H
F037FH	Timer clock select register 2	TPS2	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							

Table 3 - 6 List of 2nd Special Function Registers (9/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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							
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							

Table 3 - 6 List of 2nd Special Function Registers (10/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F03CCH	Sequencer general-purpose register 6	MSG6	R/W	—	—	✓	0000H
F03CDH				—	—	✓	
F03CEH	Sequencer general-purpose register 7	MSG7	R/W	—	—	✓	0000H
F03CFH				—	—	✓	
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
F0400H	LCD display data register 0	SEG0	R/W	—	✓	—	00H
F0401H	LCD display data register 1	SEG1	R/W	—	✓	—	00H
F0402H	LCD display data register 2	SEG2	R/W	—	✓	—	00H
F0403H	LCD display data register 3	SEG3	R/W	—	✓	—	00H
F0404H	LCD display data register 4	SEG4	R/W	—	✓	—	00H
F0405H	LCD display data register 5	SEG5	R/W	—	✓	—	00H
F0406H	LCD display data register 6	SEG6	R/W	—	✓	—	00H
F0407H	LCD display data register 7	SEG7	R/W	—	✓	—	00H
F0408H	LCD display data register 8	SEG8	R/W	—	✓	—	00H
F0409H	LCD display data register 9	SEG9	R/W	—	✓	—	00H
F040AH	LCD display data register 10	SEG10	R/W	—	✓	—	00H
F040BH	LCD display data register 11	SEG11	R/W	—	✓	—	00H
F040CH	LCD display data register 12	SEG12	R/W	—	✓	—	00H
F040DH	LCD display data register 13	SEG13	R/W	—	✓	—	00H
F040EH	LCD display data register 14	SEG14	R/W	—	✓	—	00H
F040FH	LCD display data register 15	SEG15	R/W	—	✓	—	00H
F0410H	LCD display data register 16	SEG16	R/W	—	✓	—	00H
F0411H	LCD display data register 17	SEG17	R/W	—	✓	—	00H
F0412H	LCD display data register 18	SEG18	R/W	—	✓	—	00H
F0413H	LCD display data register 19	SEG19	R/W	—	✓	—	00H
F0414H	LCD display data register 20	SEG20	R/W	—	✓	—	00H
F0415H	LCD display data register 21	SEG21	R/W	—	✓	—	00H
F0416H	LCD display data register 22	SEG22	R/W	—	✓	—	00H
F0417H	LCD display data register 23	SEG23	R/W	—	✓	—	00H
F0418H	LCD display data register 24	SEG24	R/W	—	✓	—	00H
F0419H	LCD display data register 25	SEG25	R/W	—	✓	—	00H
F041AH	LCD display data register 26	SEG26	R/W	—	✓	—	00H
F041BH	LCD display data register 27	SEG27	R/W	—	✓	—	00H
F041CH	LCD display data register 28	SEG28	R/W	—	✓	—	00H
F041DH	LCD display data register 29	SEG29	R/W	—	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (11/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F041EH	LCD display data register 30	SEG30	R/W	—	✓	—	00H
F041FH	LCD display data register 31	SEG31	R/W	—	✓	—	00H
F0420H	LCD display data register 32	SEG32	R/W	—	✓	—	00H
F0421H	LCD display data register 33	SEG33	R/W	—	✓	—	00H
F0422H	LCD display data register 34	SEG34	R/W	—	✓	—	00H
F0423H	LCD display data register 35	SEG35	R/W	—	✓	—	00H
F0424H	LCD display data register 36	SEG36	R/W	—	✓	—	00H
F0425H	LCD display data register 37	SEG37	R/W	—	✓	—	00H
F0426H	LCD display data register 38	SEG38	R/W	—	✓	—	00H
F0427H	LCD display data register 39	SEG39	R/W	—	✓	—	00H
F0428H	LCD display data register 40	SEG40	R/W	—	✓	—	00H
F0429H	LCD display data register 41	SEG41	R/W	—	✓	—	00H
F042AH	LCD display data register 42	SEG42	R/W	—	✓	—	00H
F042BH	LCD display data register 43	SEG43	R/W	—	✓	—	00H
F042CH	LCD display data register 44	SEG44	R/W	—	✓	—	00H
F042DH	LCD display data register 45	SEG45	R/W	—	✓	—	00H
F042EH	LCD display data register 46	SEG46	R/W	—	✓	—	00H
F042FH	LCD display data register 47	SEG47	R/W	—	✓	—	00H
F0430H	LCD display data register 48	SEG48	R/W	—	✓	—	00H
F0431H	LCD display data register 49	SEG49	R/W	—	✓	—	00H
F0432H	LCD display data register 50	SEG50	R/W	—	✓	—	00H
F0433H	LCD display data register 51	SEG51	R/W	—	✓	—	00H
F0434H	LCD display data register 52	SEG52	R/W	—	✓	—	00H
F0435H	LCD display data register 53	SEG53	R/W	—	✓	—	00H
F0436H	LCD display data register 54	SEG54	R/W	—	✓	—	00H
F0437H	LCD display data register 55	SEG55	R/W	—	✓	—	00H
F0440H	8-bit interval timer compare register 00	TRTCMP00	TRTCMP0	R/W	—	✓	FFH
F0441H	8-bit interval timer compare register 01	TRTCMP01		R/W	—	✓	FFH
F0442H	8-bit interval timer control register 0	TRTCR0		R/W	✓	✓	00H
F0443H	8-bit interval timer frequency division register 0	TRTMD0		R/W	—	✓	00H
F0448H	8-bit interval timer compare register 10	TRTCMP10	TRTCMP1	R/W	—	✓	FFH
F0449H	8-bit interval timer compare register 11	TRTCMP11		R/W	—	✓	FFH
F044AH	8-bit interval timer control register 1	TRTCR1		R/W	✓	✓	00H
F044BH	8-bit interval timer frequency division register 1	TRTMD1		R/W	—	✓	00H
F0450H	8-bit interval timer compare register 20	TRTCMP20	TRTCMP2	R/W	—	✓	FFH
F0451H	8-bit interval timer compare register 21	TRTCMP21		R/W	—	✓	FFH
F0452H	8-bit interval timer control register 2	TRTCR2		R/W	✓	✓	00H
F0453H	8-bit interval timer frequency division register 2	TRTMD2		R/W	—	✓	00H
F0458H	8-bit interval timer compare register 30	TRTCMP30	TRTCMP3	R/W	—	✓	FFH
F0459H	8-bit interval timer compare register 31	TRTCMP31		R/W	—	✓	FFH
F045AH	8-bit interval timer control register 3	TRTCR3		R/W	✓	✓	00H
F045BH	8-bit interval timer frequency division register 3	TRTMD3		R/W	—	✓	00H
F0460H	8-bit interval timer compare register 40	TRTCMP40	TRTCMP4	R/W	—	✓	FFH
F0461H	8-bit interval timer compare register 41	TRTCMP41		R/W	—	✓	FFH
F0462H	8-bit interval timer control register 4	TRTCR4		R/W	✓	✓	00H
F0463H	8-bit interval timer frequency division register 4	TRTMD4		R/W	—	✓	00H
F0468H	8-bit interval timer compare register 50	TRTCMP50	TRTCMP5	R/W	—	✓	FFH
F0469H	8-bit interval timer compare register 51	TRTCMP51		R/W	—	✓	FFH
F046AH	8-bit interval timer control register 5	TRTCR5		R/W	✓	✓	00H
F046BH	8-bit interval timer frequency division register 5	TRTMD5		R/W	—	✓	00H
F0470H	8-bit interval timer compare register 60	TRTCMP60	TRTCMP6	R/W	—	✓	FFH
F0471H	8-bit interval timer compare register 61	TRTCMP61		R/W	—	✓	FFH

Table 3 - 6 List of 2nd Special Function Registers (12/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0472H	8-bit interval timer control register 6	TRTCR6	R/W	✓	✓	—	00H
F0473H	8-bit interval timer frequency division register 6	TRTMD6	R/W	—	✓	—	00H
F0478H	8-bit interval timer compare register 70	TRTCMP70	TRTCMP7	—	✓	✓	FFH
F0479H	8-bit interval timer compare register 71	TRTCMP71		—	✓		FFH
F047AH	8-bit interval timer control register 7	TRTCR7	R/W	✓	✓	—	00H
F047BH	8-bit interval timer frequency division register 7	TRTMD7	R/W	—	✓	—	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							
F0490H	16-bit timer KB skipping control register 0	TKBTCTL0	R/W	—	✓	—	00H
F0491H	16-bit timer KB times-of-skipping setting register 0	TKBTCMP0	R/W	—	✓	—	00H
F0492H	16-bit timer KB skipping control register 1	TKBTCTL1	R/W	—	✓	—	00H
F0493H	16-bit timer KB times-of-skipping setting register 1	TKBTCMP1	R/W	—	✓	—	00H
F0494H	16-bit timer KB skipping control register 2	TKBTCTL2	R/W	—	✓	—	00H
F0495H	16-bit timer KB times-of-skipping setting register 2	TKBTCMP2	R/W	—	✓	—	00H
F04A0H	16-bit timer KB compare register 20	TKBCR20	R/W	—	—	✓	0000H
F04A1H							
F04A2H	16-bit timer KB compare register 21	TKBCR21	R/W	—	—	✓	0000H
F04A3H							
F04A4H	16-bit timer KB compare register 22	TKBCR22	R/W	—	—	✓	0000H
F04A5H							
F04A6H	16-bit timer KB compare register 23	TKBCR23	R/W	—	—	✓	0000H
F04A7H							
F04A8H	16-bit timer KB trigger compare register 2	TKBTGCR2	R/W	—	—	✓	0000H
F04A9H							
F04AAH	16-bit timer KB smooth start initial duty register 20	TKBSIR20	R/W	—	—	✓	0000H
F04ABH							
F04ACH	16-bit timer KB smooth start initial duty register 21	TKBSIR21	R/W	—	—	✓	0000H
F04ADH							
F04AEH	16-bit timer KB dithering count register 20	TKBDNR20	R/W	—	✓	—	00H
F04AFH	16-bit timer KB smooth start step width register 20	TKBSSR20	R/W	—	✓	—	00H
F04B0H	16-bit timer KB dithering count register 21	TKBDNR21	R/W	—	✓	—	00H
F04B1H	16-bit timer KB smooth start step width register 21	TKBSSR21	R/W	—	✓	—	00H
F04B2H	16-bit timer KB trigger register 2	TKBTRG2	W	✓	✓	—	00H
F04B3H	16-bit timer KB flag register 2	TKBFLG2	R	✓	✓	—	00H
F04B4H	16-bit timer KB compare 1L & dithering count register 20	TKBCRLD20	R/W	—	—	✓	0000H
F04B5H							
F04B6H	16-bit timer KB compare 3L & dithering count register 21	TKBCRLD21	R/W	—	—	✓	0000H
F04B7H							
F04C0H	16-bit timer counter KB2	TKBCNT2	R	—	—	✓	FFFFH
F04C1H							
F04C2H	16-bit timer KB operation control register 20	TKBCTL20	R/W	—	—	✓	0040H
F04C3H							

Table 3 - 6 List of 2nd Special Function Registers (13/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F04C4H F04C5H	16-bit timer KB maximum frequency limit setting register 2	TKBMFR2	R/W	—	—	✓	0000H
F04C6H	16-bit timer KB output control register 20	TKBIOC20	R/W	✓	✓	—	00H
F04C7H	16-bit timer KB flag clear trigger register 2	TKBCLR2	W	✓	✓	—	00H
F04C8H	16-bit timer KB output control register 21	TKBIOC21	R/W	✓	✓	—	00H
F04C9H	16-bit timer KB operation control register 21	TKBCTL21	R/W	✓	✓	—	00H
F04CAH F04CBH	16-bit timer KB operation control register 22	TKBCTL22	R/W	—	—	✓	0000H
F04D0H F04D1H	Forced output stop function control register 20	TKBPACTL20	R/W	—	—	✓	0000H
F04D2H F04D3H	Forced output stop function control register 21	TKBPACTL21	R/W	—	—	✓	0000H
F04D4H	Forced output stop function start trigger register 2	TKBPAHFS2	R/W	✓	✓	—	00H
F04D5H	Forced output stop function cancel trigger register 2	TKBPAHFT2	R/W	✓	✓	—	00H
F04D6H	Forced output stop function flag register 2	TKBPAFLG2	R	✓	✓	—	00H
F04D7H	Forced output stop function control register 22	TKBPACTL22	R/W	✓	✓	—	00H
F04D8H F04D9H	Forced output stop function 1 input select register 20	TKBPAHXS20	R/W	—	—	✓	0000H
F04DAH F04DBH	Forced output stop function 1 input select register 21	TKBPAHXS21	R/W	—	—	✓	0000H
F04DCH F04DDH	Forced output stop function 2 input select register 20	TKBPAFXS20	R/W	—	—	✓	0000H
F04DEH F04DFH	Forced output stop function 2 input select register 21	TKBPAFXS21	R/W	—	—	✓	0000H
F04E0H	Peripheral I/O redirection register 0	PIOR0	R/W	—	✓	—	00H
F04E1H	Peripheral I/O redirection register 1	PIOR1	R/W	—	✓	—	00H
F04E2H	Peripheral I/O redirection register 2	PIOR2	R/W	—	✓	—	00H
F04E3H	Peripheral I/O redirection register 3	PIOR3	R/W	—	✓	—	00H
F04E4H	Peripheral I/O redirection register 4	PIOR4	R/W	—	✓	—	00H
F04E5H	Peripheral I/O redirection register 5	PIOR5	R/W	—	✓	—	00H
F04E6H	Peripheral I/O redirection register 6	PIOR6	R/W	—	✓	—	00H
F04E7H	Peripheral I/O redirection register 7	PIOR7	R/W	—	✓	—	00H
F04E8H	Peripheral I/O redirection register 8	PIOR8	R/W	—	✓	—	00H
F04E9H	Peripheral I/O redirection register 9	PIOR9	R/W	—	✓	—	00H
F04EAH	Peripheral I/O redirection register 10	PIOR10	R/W	—	✓	—	00H
F04EBH	Peripheral I/O redirection register 11	PIOR11	R/W	—	✓	—	00H
F04ECH	Peripheral I/O redirection register 12	PIOR12	R/W	—	✓	—	00H
F04EDH	Peripheral I/O redirection register 13	PIOR13	R/W	—	✓	—	00H
F0500H F0501H	CTSUC control register AL	CTSUCR0 CTSUCR1	CTSUCRAL	✓	✓	✓	0000H
F0502H F0503H	CTSUC control register AH	CTSUCR2 CTSUCR3		✓	✓		
F0504H F0505H	CTSUC control register BL	CTSUSDP RS CTSUSST	CTSUCRBL	✓	✓	✓	0000H
F0506H F0507H	CTSUC control register BH	— CTSUDCLKC		—	—		
F0508H F0509H	CTSUC measurement channel register L	CTSUCMCH 0 CTSUCMCH 1	CTSUCMCHL	—	✓	✓	3F3FH
				—	✓		

Table 3 - 6 List of 2nd Special Function Registers (14/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F050AH	CTSU measurement channel register H	CTSUMFA F	CTSUMC HH	R/W	✓	✓	✓	0000H
F050BH		—			—	—		
F050CH	CTSU channel enable control register AL	CTSUCHA C0	CTSUCHA CAL	R/W	✓	✓	✓	0000H
F050DH		CTSUCHA C1			✓	✓		
F050EH	CTSU channel enable control register AH	CTSUCHA C2	CTSUCHA CAH	R/W	✓	✓	✓	0000H
F050FH		CTSUCHA C3			✓	✓		
F0510H	CTSU channel enable control register BL	CTSUCHA C4	CTSUCHA CBL	R/W	✓	✓	✓	0000H
F0511H		CTSUCHA C5			✓	✓		
F0514H	CTSU channel transmit/receive control register AL	CTSUCHT RC0	CTSUCHT RCAL	R/W	✓	✓	✓	0000H
F0515H		CTSUCHT RC1			✓	✓		
F0516H	CTSU channel transmit/receive control register AH	CTSUCHT RC2	CTSUCHT CAH	R/W	✓	✓	✓	0000H
F0517H		CTSUCHT RC3			✓	✓		
F0518H	CTSU channel transmit/receive control register BL	CTSUCHT RC4	CTSUCHT RCBL	R/W	✓	✓	✓	0000H
F0519H		CTSUCHT RC5			✓	✓		
F051CH	CTSU status register L	CTSUST1	CTSUSRL	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	CTSUSC		R	—	—	✓	0000H
F0525H					—	—		
F0526H	CTSU sensor counter register	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 seed command register 0	TRNGSCR0		R/W	✓	✓	—	00H
F0550H	8-bit interval timer counter register 00	TRT00	TRT0	R	—	✓	✓	00H
F0551H	8-bit interval timer counter register 01	TRT01		R	—	✓		00H
F0558H	8-bit interval timer counter register 10	TRT10	TRT1	R	—	✓	✓	00H
F0559H	8-bit interval timer counter register 11	TRT11		R	—	✓		00H

Table 3 - 6 List of 2nd Special Function Registers (15/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0560H	8-bit interval timer counter register 20	TRT20	TRT2	R	—	✓	✓	00H
F0561H	8-bit interval timer counter register 21	TRT21		R	—	✓		00H
F0568H	8-bit interval timer counter register 30	TRT30	TRT3	R	—	✓	✓	00H
F0569H	8-bit interval timer counter register 31	TRT31		R	—	✓		00H
F0570H	8-bit interval timer counter register 40	TRT40	TRT4	R	—	✓	✓	00H
F0571H	8-bit interval timer counter register 41	TRT41		R	—	✓		00H
F0578H	8-bit interval timer counter register 50	TRT50	TRT5	R	—	✓	✓	00H
F0579H	8-bit interval timer counter register 51	TRT51		R	—	✓		00H
F0580H	8-bit interval timer counter register 60	TRT60	TRT6	R	—	✓	✓	00H
F0581H	8-bit interval timer counter register 61	TRT61		R	—	✓		00H
F0588H	8-bit interval timer counter register 70	TRT70	TRT7	R	—	✓	✓	00H
F0589H	8-bit interval timer counter register 71	TRT71		R	—	✓		00H
F0598H	Timer RJ counter register 0	TRJ0		R/W	—	—	✓	FFFFH
F0599H								
F059AH	Timer RJ counter register 1	TRJ1		R/W	—	—	✓	FFFFH
F059BH								
F0600H	CTSU trimming register AL	RTRIM	CTSUTRI M0	R/W	—	✓	✓	Note 1
F0601H		DACTRIM			—	✓		
F0602H	CTSU trimming register AH	SUADJD	CTSUTRI M1	R/W	—	✓	✓	Note 1
F0603H		TRESULT4			—	✓		
F0604H	CTSU trimming register BL	TRESULT0	CTSUTRI M2	R/W	—	✓	✓	Note 1
F0605H		TRESULT1			—	✓		
F0606H	CTSU trimming register BH	TRESULT2	CTSUTRI M3	R/W	—	✓	✓	Note 1
F0607H		TRESULT3			—	✓		
F0610H	Transmit buffer register 1	TXBA1		R/W	—	✓	—	FFH
F0611H	Receive buffer register 1	RXBA1		R	—	✓	—	FFH
F0612H	Operation mode setting register 10	ASIMA10		R/W	✓	✓	—	01H
F0613H	Operation mode setting register 11	ASIMA11		R/W	✓	✓	—	1AH
F0614H	Baud rate generator control register 1	BRGCA1		R/W	—	✓	—	FFH
F0615H	Status register 1	ASISA1		R	—	✓	—	00H
F0616H	Status clear trigger register 1	ASCTA1		R/W	✓	✓	—	00H
F0620H	UARTA clock select register 1	UTA1CK		R/W	✓	✓	—	00H
F0630H	Transmit buffer register 2	TXBA2		R/W	—	✓	—	FFH
F0631H	Receive buffer register 2	RXBA2		R	—	✓	—	FFH
F0632H	Operation mode setting register 20	ASIMA20		R/W	✓	✓	—	01H
F0633H	Operation mode setting register 21	ASIMA21		R/W	✓	✓	—	1AH
F0634H	Baud rate generator control register 2	BRGCA2		R/W	—	✓	—	FFH
F0635H	Status register 2	ASISA2		R	—	✓	—	00H
F0636H	Status clear trigger register 2	ASCTA2		R/W	✓	✓	—	00H
F0640H	UARTA clock select register 2	UTA2CK		R/W	✓	✓	—	00H
F0650H	Transmit buffer register 3	TXBA3		R/W	—	✓	—	FFH
F0651H	Receive buffer register 3	RXBA3		R	—	✓	—	FFH
F0652H	Operation mode setting register 30	ASIMA30		R/W	✓	✓	—	01H
F0653H	Operation mode setting register 31	ASIMA31		R/W	✓	✓	—	1AH
F0654H	Baud rate generator control register 3	BRGCA3		R/W	—	✓	—	FFH
F0655H	Status register 3	ASISA3		R	—	✓	—	00H
F0656H	Status clear trigger register 3	ASCTA3		R/W	✓	✓	—	00H
F0660H	UARTA clock select register 3	UTA3CK		R/W	✓	✓	—	00H
F0680H	Input signal select register 0	ELISEL0		R/W	—	✓	—	00H
F0681H	Input signal select register 1	ELISEL1		R/W	—	✓	—	00H
F0682H	Input signal select register 2	ELISEL2		R/W	—	✓	—	00H
F0683H	Input signal select register 3	ELISEL3		R/W	—	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (16/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0684H	Input signal select register 4	ELISEL4	R/W	—	✓	—	00H
F0685H	Input signal select register 5	ELISEL5	R/W	—	✓	—	00H
F0686H	Input signal select register 6	ELISEL6	R/W	—	✓	—	00H
F0687H	Input signal select register 7	ELISEL7	R/W	—	✓	—	00H
F0688H	Input signal select register 8	ELISEL8	R/W	—	✓	—	00H
F0689H	Input signal select register 9	ELISEL9	R/W	—	✓	—	00H
F068AH	Input signal select register 10	ELISEL10	R/W	—	✓	—	00H
F068BH	Input signal select register 11	ELISEL11	R/W	—	✓	—	00H
F0690H	Event link L1 signal select register 0	ELL1SEL0	R/W	—	✓	—	00H
F0691H	Event link L1 signal select register 1	ELL1SEL1	R/W	—	✓	—	00H
F0692H	Event link L1 signal select register 2	ELL1SEL2	R/W	—	✓	—	00H
F0693H	Event link L1 signal select register 3	ELL1SEL3	R/W	—	✓	—	00H
F0694H	Event link L1 signal select register 4	ELL1SEL4	R/W	—	✓	—	00H
F0695H	Event link L1 signal select register 5	ELL1SEL5	R/W	—	✓	—	00H
F0696H	Event link L1 signal select register 6	ELL1SEL6	R/W	—	✓	—	00H
F0697H	Logic cell block L1 control register	ELL1CTL	R/W	—	✓	—	00H
F0698H	Event link L1 output select register 0	ELL1LNK0	R/W	—	✓	—	00H
F0699H	Event link L1 output select register 1	ELL1LNK1	R/W	—	✓	—	00H
F069AH	Event link L1 output select register 2	ELL1LNK2	R/W	—	✓	—	00H
F069BH	Event link L1 output select register 3	ELL1LNK3	R/W	—	✓	—	00H
F069CH	Event link L1 output select register 4	ELL1LNK4	R/W	—	✓	—	00H
F069DH	Event link L1 output select register 5	ELL1LNK5	R/W	—	✓	—	00H
F069EH	Event link L1 output select register 6	ELL1LNK6	R/W	—	✓	—	00H
F06A0H	Event link L2 signal select register 0	ELL2SEL0	R/W	—	✓	—	00H
F06A1H	Event link L2 signal select register 1	ELL2SEL1	R/W	—	✓	—	00H
F06A2H	Event link L2 signal select register 2	ELL2SEL2	R/W	—	✓	—	00H
F06A3H	Event link L2 signal select register 3	ELL2SEL3	R/W	—	✓	—	00H
F06A4H	Event link L2 signal select register 4	ELL2SEL4	R/W	—	✓	—	00H
F06A5H	Event link L2 signal select register 5	ELL2SEL5	R/W	—	✓	—	00H
F06A6H	Event link L2 signal select register 6	ELL2SEL6	R/W	—	✓	—	00H
F06A7H	Logic cell block L2 control register	ELL2CTL	R/W	—	✓	—	00H
F06A8H	Event link L2 output select register 0	ELL2LNK0	R/W	—	✓	—	00H
F06A9H	Event link L2 output select register 1	ELL2LNK1	R/W	—	✓	—	00H
F06AAH	Event link L2 output select register 2	ELL2LNK2	R/W	—	✓	—	00H
F06ABH	Event link L2 output select register 3	ELL2LNK3	R/W	—	✓	—	00H
F06ACH	Event link L2 output select register 4	ELL2LNK4	R/W	—	✓	—	00H
F06ADH	Event link L2 output select register 5	ELL2LNK5	R/W	—	✓	—	00H
F06AEH	Event link L2 output select register 6	ELL2LNK6	R/W	—	✓	—	00H
F06B0H	Event link L3 signal select register 0	ELL3SEL0	R/W	—	✓	—	00H
F06B1H	Event link L3 signal select register 1	ELL3SEL1	R/W	—	✓	—	00H
F06B2H	Event link L3 signal select register 2	ELL3SEL2	R/W	—	✓	—	00H
F06B3H	Event link L3 signal select register 3	ELL3SEL3	R/W	—	✓	—	00H
F06B4H	Event link L3 signal select register 4	ELL3SEL4	R/W	—	✓	—	00H
F06B5H	Event link L3 signal select register 5	ELL3SEL5	R/W	—	✓	—	00H
F06B6H	Event link L3 signal select register 6	ELL3SEL6	R/W	—	✓	—	00H
F06B7H	Logic cell block L3 control register	ELL3CTL	R/W	—	✓	—	00H
F06B8H	Event link L3 output select register 0	ELL3LNK0	R/W	—	✓	—	00H
F06B9H	Event link L3 output select register 1	ELL3LNK1	R/W	—	✓	—	00H
F06BAH	Event link L3 output select register 2	ELL3LNK2	R/W	—	✓	—	00H
F06BBH	Event link L3 output select register 3	ELL3LNK3	R/W	—	✓	—	00H
F06BCH	Event link L3 output select register 4	ELL3LNK4	R/W	—	✓	—	00H
F06BDH	Event link L3 output select register 5	ELL3LNK5	R/W	—	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (17/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F06BEH	Event link L3 output select register 6	ELL3LNK6	R/W	—	✓	—	00H
F06C0H	Output signal select register 0	ELOSEL0	R/W	—	✓	—	00H
F06C1H	Output signal select register 1	ELOSEL1	R/W	—	✓	—	00H
F06C2H	Output signal select register 2	ELOSEL2	R/W	—	✓	—	00H
F06C3H	Output signal select register 3	ELOSEL3	R/W	—	✓	—	00H
F06C4H	Output signal select register 4	ELOSEL4	R/W	—	✓	—	00H
F06C5H	Output signal select register 5	ELOSEL5	R/W	—	✓	—	00H
F06C6H	Output signal select register 6	ELOSEL6	R/W	—	✓	—	00H
F06C7H	Output signal select register 7	ELOSEL7	R/W	—	✓	—	00H
F06C8H	Output signal enable register	ELOENCTL	R/W	—	✓	—	00H
F06C9H	Output signal monitor register	ELOMONI	R	—	✓	—	00H
F0740H	16-bit timer KB compare register 00	TKBCR00	R/W	—	—	✓	0000H
F0741H							
F0742H	16-bit timer KB compare register 01	TKBCR01	R/W	—	—	✓	0000H
F0743H							
F0744H	16-bit timer KB compare register 02	TKBCR02	R/W	—	—	✓	0000H
F0745H							
F0746H	16-bit timer KB compare register 03	TKBCR03	R/W	—	—	✓	0000H
F0747H							
F0748H	16-bit timer KB trigger compare register 0	TKBTGCR0	R/W	—	—	✓	0000H
F0749H							
F074AH	16-bit timer KB smooth start initial duty register 00	TKBSIR00	R/W	—	—	✓	0000H
F074BH							
F074CH	16-bit timer KB smooth start initial duty register 01	TKBSIR01	R/W	—	—	✓	0000H
F074DH							
F074EH	16-bit timer KB dithering count register 00	TKBDNR00	R/W	—	✓	—	00H
F074FH	16-bit timer KB smooth start step width register 00	TKBSSR00	R/W	—	✓	—	00H
F0750H	16-bit timer KB dithering count register 01	TKBDNR01	R/W	—	✓	—	00H
F0751H	16-bit timer KB smooth start step width register 01	TKBSSR01	R/W	—	✓	—	00H
F0752H	16-bit timer KB trigger register 0	TKBTRG0	W	✓	✓	—	00H
F0753H	16-bit timer KB flag register 0	TKBFLG0	R	✓	✓	—	00H
F0754H	16-bit timer KB compare 1L & dithering count register 00	TKBCRLD00	R/W	—	—	✓	0000H
F0755H							
F0756H	16-bit timer KB compare 3L & dithering count register 01	TKBCRLD01	R/W	—	—	✓	0000H
F0757H							
F0760H	16-bit timer counter KB0	TKBCNT0	R	—	—	✓	FFFFH
F0761H							
F0762H	16-bit timer KB operation control register 00	TKBCTL00	R/W	—	—	✓	0040H
F0763H							
F0764H	16-bit timer KB maximum frequency limit setting register 0	TKBMFR0	R/W	—	—	✓	0000H
F0765H							
F0766H	16-bit timer KB output control register 00	TKBIOC00	R/W	✓	✓	—	00H
F0767H	16-bit timer KB flag clear trigger register 0	TKBCLR0	W	✓	✓	—	00H
F0768H	16-bit timer KB output control register 01	TKBIOC01	R/W	✓	✓	—	00H
F0769H	16-bit timer KB operation control register 01	TKBCTL01	R/W	✓	✓	—	00H
F076AH	16-bit timer KB operation control register 02	TKBCTL02	R/W	—	—	✓	0000H
F076BH							
F0770H	Forced output stop function control register 00	TKBPACTL00	R/W	—	—	✓	0000H
F0771H							
F0772H	Forced output stop function control register 01	TKBPACTL01	R/W	—	—	✓	0000H
F0773H							
F0774H	Forced output stop function start trigger register 0	TKBPAHFS0	R/W	✓	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (18/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0775H	Forced output stop function cancel trigger register 0	TKBPAHFT0	R/W	✓	✓	—	00H
F0776H	Forced output stop function flag register 0	TKBP AFLG0	R	✓	✓	—	00H
F0777H	Forced output stop function control register 02	TKBPACTL02	R/W	✓	✓	—	00H
F0778H	Forced output stop function 1 input select register 00	TKBPAHZS00	R/W	—	—	✓	0000H
F0779H							
F077AH	Forced output stop function 1 input select register 01	TKBPAHZS01	R/W	—	—	✓	0000H
F077BH							
F077CH	Forced output stop function 2 input select register 00	TKBPAFXS00	R/W	—	—	✓	0000H
F077DH							
F077EH	Forced output stop function 2 input select register 01	TKBPAFXS01	R/W	—	—	✓	0000H
F077FH							
F0780H	16-bit timer KB compare register 10	TKBCR10	R/W	—	—	✓	0000H
F0781H							
F0782H	16-bit timer KB compare register 11	TKBCR11	R/W	—	—	✓	0000H
F0783H							
F0784H	16-bit timer KB compare register 12	TKBCR12	R/W	—	—	✓	0000H
F0785H							
F0786H	16-bit timer KB compare register 13	TKBCR13	R/W	—	—	✓	0000H
F0787H							
F0788H	16-bit timer KB trigger compare register 1	TKBTGCR1	R/W	—	—	✓	0000H
F0789H							
F078AH	16-bit timer KB smooth start initial duty register 10	TKBSIR10	R/W	—	—	✓	0000H
F078BH							
F078CH	16-bit timer KB smooth start initial duty register 11	TKBSIR11	R/W	—	—	✓	0000H
F078DH							
F078EH	16-bit timer KB dithering count register 10	TKBDNR10	R/W	—	✓	—	00H
F078FH	16-bit timer KB smooth start step width register 10	TKBSSR10	R/W	—	✓	—	00H
F0790H	16-bit timer KB dithering count register 11	TKBDNR11	R/W	—	✓	—	00H
F0791H	16-bit timer KB smooth start step width register 11	TKBSSR11	R/W	—	✓	—	00H
F0792H	16-bit timer KB trigger register 1	TKBTRG1	W	✓	✓	—	00H
F0793H	16-bit timer KB flag register 1	TKBFLG1	R	✓	✓	—	00H
F0794H	16-bit timer KB compare 1L & dithering count register 10	TKBCRLD10	R/W	—	—	✓	0000H
F0795H							
F0796H	16-bit timer KB compare 3L & dithering count register 11	TKBCRLD11	R/W	—	—	✓	0000H
F0797H							
F07A0H	16-bit timer counter KB1	TKBCNT1	R	—	—	✓	FFFFH
F07A1H							
F07A2H	16-bit timer KB operation control register 10	TKBCTL10	R/W	—	—	✓	0040H
F07A3H							
F07A4H	16-bit timer KB maximum frequency limit setting register 1	TKBMFR1	R/W	—	—	✓	0000H
F07A5H							
F07A6H	16-bit timer KB output control register 10	TKBIOC10	R/W	✓	✓	—	00H
F07A7H	16-bit timer KB flag clear trigger register 1	TKBCLR1	W	✓	✓	—	00H
F07A8H	16-bit timer KB output control register 11	TKBIOC11	R/W	✓	✓	—	00H
F07A9H	16-bit timer KB operation control register 11	TKBCTL11	R/W	✓	✓	—	00H
F07AAH	16-bit timer KB operation control register 12	TKBCTL12	R/W	—	—	✓	0000H
F07ABH							
F07B0H	Forced output stop function control register 10	TKBPACTL10	R/W	—	—	✓	0000H
F07B1H							
F07B2H	Forced output stop function control register 11	TKBPACTL11	R/W	—	—	✓	0000H
F07B3H							
F07B4H	Forced output stop function start trigger register 1	TKBPAHFS1	R/W	✓	✓	—	00H

Table 3 - 6 List of 2nd Special Function Registers (19/19)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F07B5H	Forced output stop function cancel trigger register 1	TKBPAHFT1	R/W	✓	✓	—	00H
F07B6H	Forced output stop function flag register 1	TKBP AFLG1	R	✓	✓	—	00H
F07B7H	Forced output stop function control register 12	TKBPACTL12	R/W	✓	✓	—	00H
F07B8H	Forced output stop function 1 input select register 00	TKBPAHZS10	R/W	—	—	✓	0000H
F07B9H							
F07BAH	Forced output stop function 1 input select register 01	TKBPAHZS11	R/W	—	—	✓	0000H
F07BBH							
F07BCH	Forced output stop function 2 input select register 00	TKBP AFXS10	R/W	—	—	✓	0000H
F07BDH							
F07BEH	Forced output stop function 2 input select register 01	TKBP AFXS11	R/W	—	—	✓	0000H
F07BFH							

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.

Note 4. In the initial value of the OSMC register, the value of the RTCLPC and WUTMMCK0 bits is 0 and that of the HIPREC bit is 1.

Remark For SFRs in the SFR area, see **Table 3 - 5 List of Special Function Registers (SFRs)**.

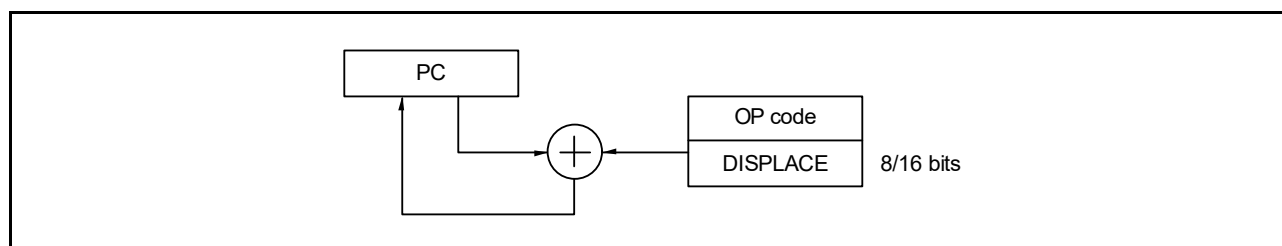
3.3 Instruction Address Addressing

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 - 13 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 - 14 Example of CALL !!addr20/BR !!addr20

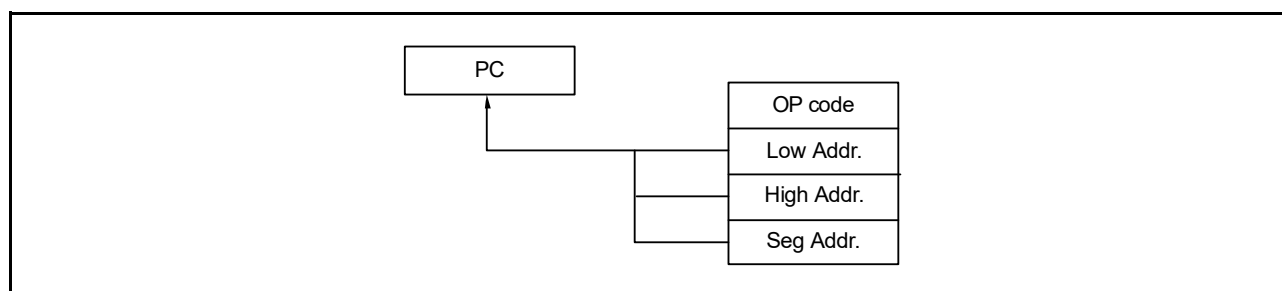
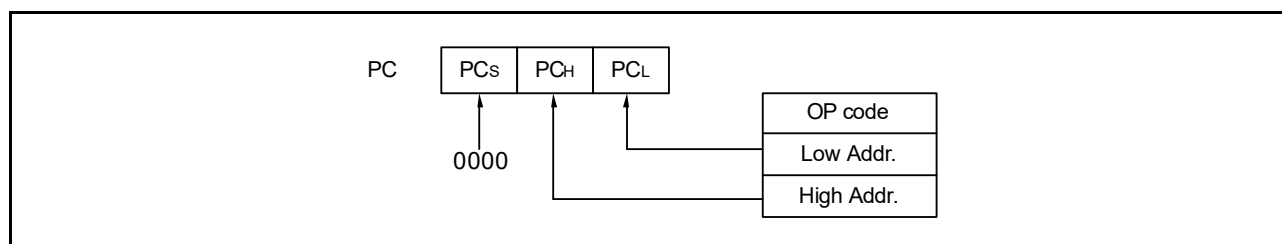


Figure 3 - 15 Example of CALL !addr16/BR !addr16

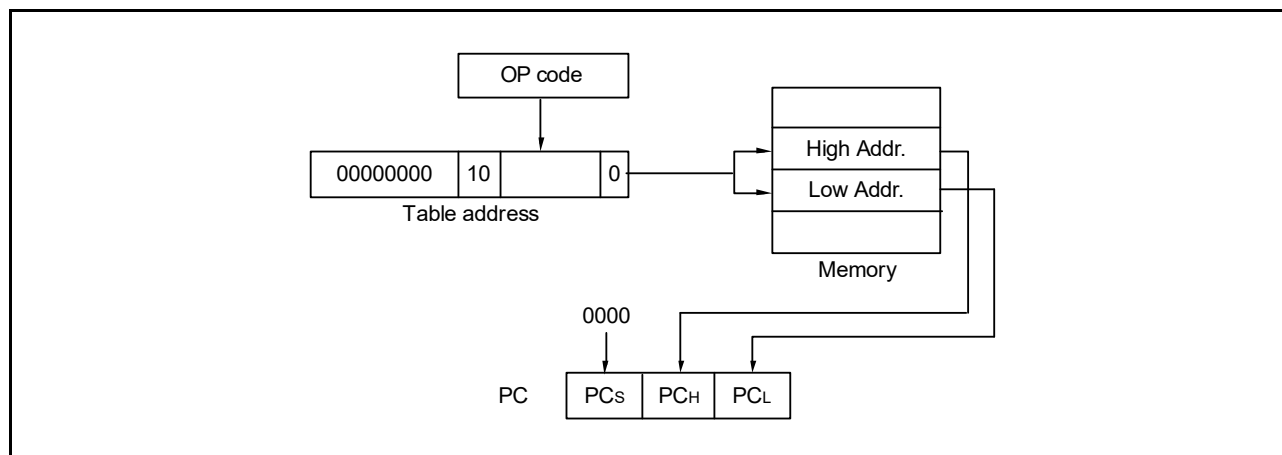


3.3.3 Table indirect addressing

[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 - 16 Outline of Table Indirect Addressing

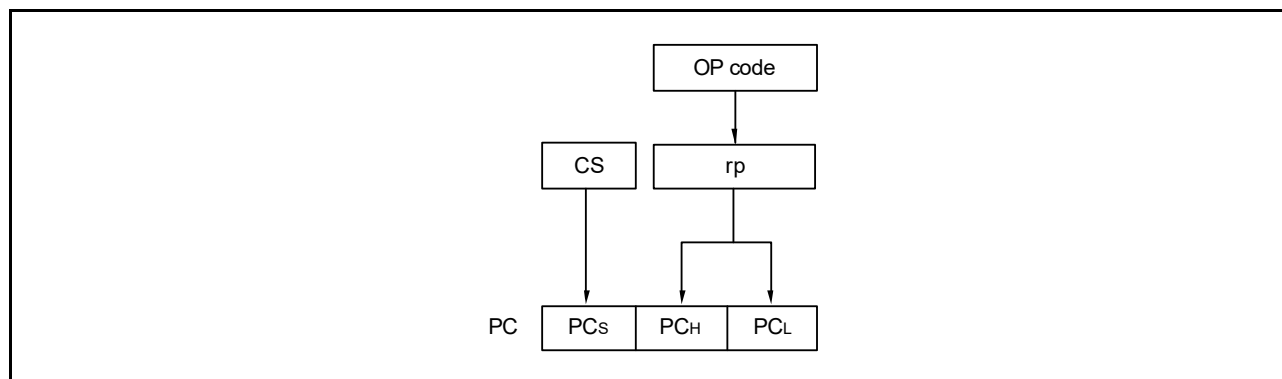


3.3.4 Register indirect addressing

[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 - 17 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

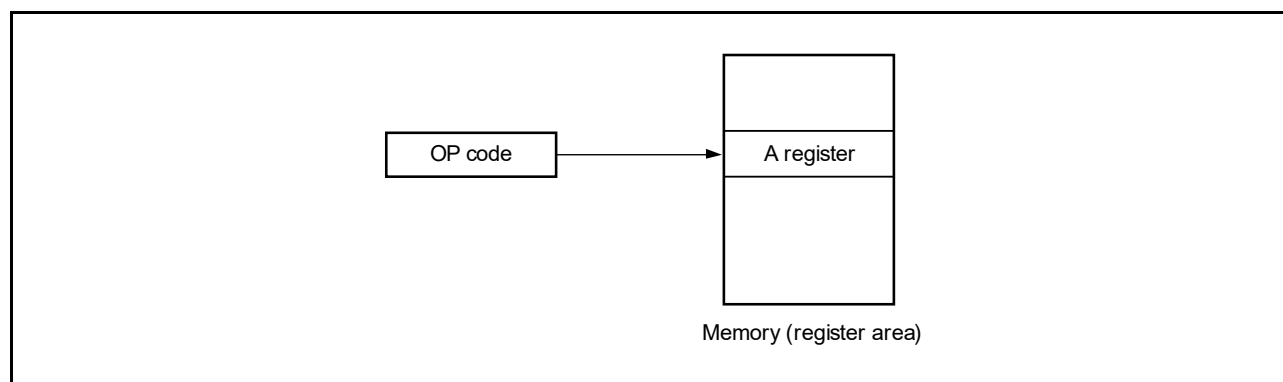
[Function]

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 - 18 Outline of Implied Addressing



3.4.2 Register addressing

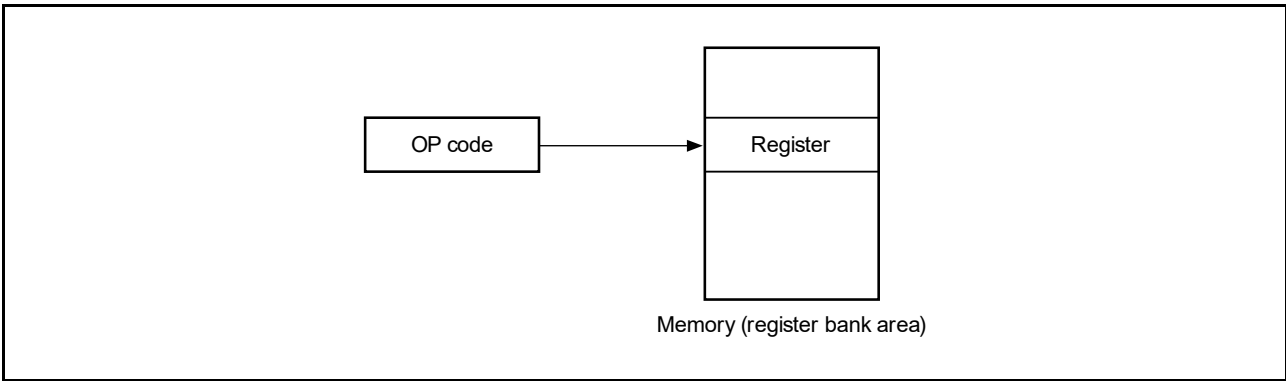
[Function]

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 - 19 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 - 20 Example of !addr16

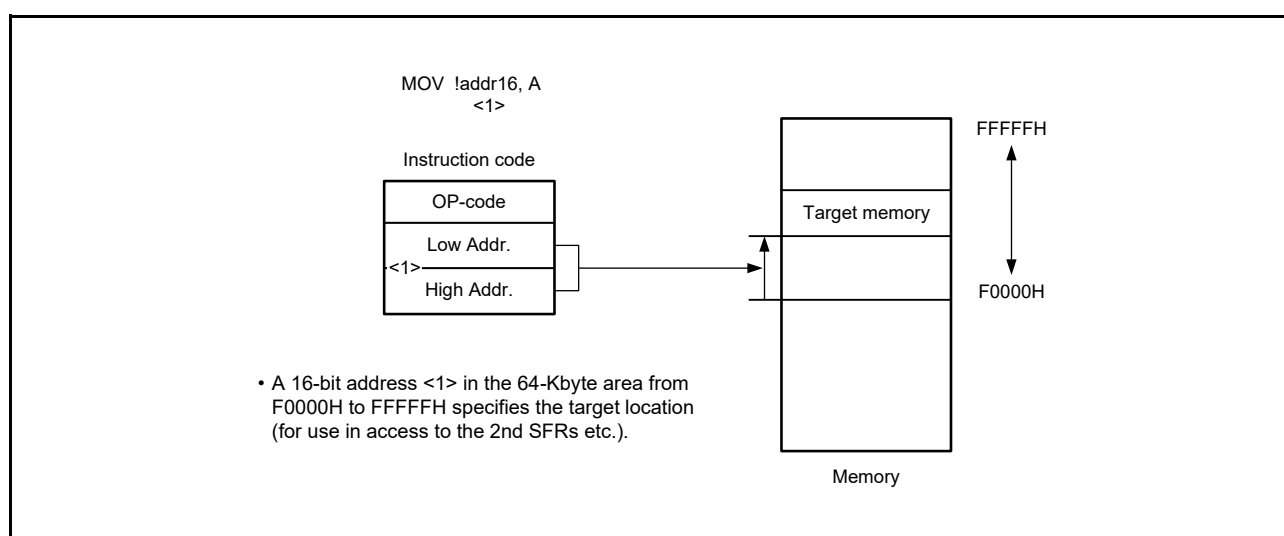
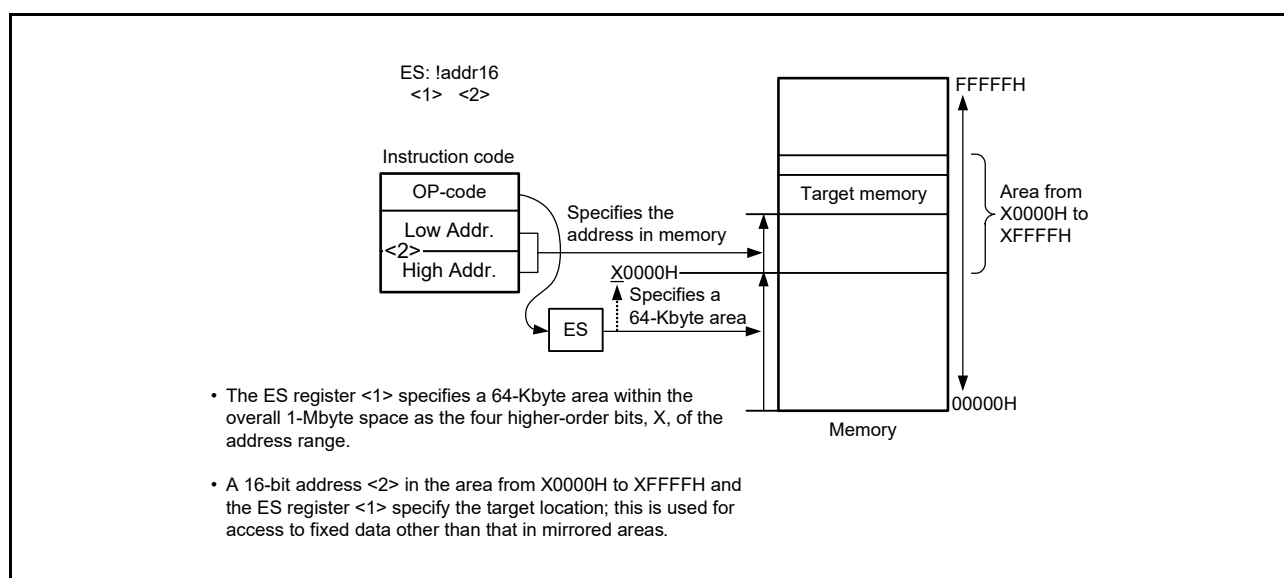


Figure 3 - 21 Example of ES:!addr16



3.4.4 Short direct addressing

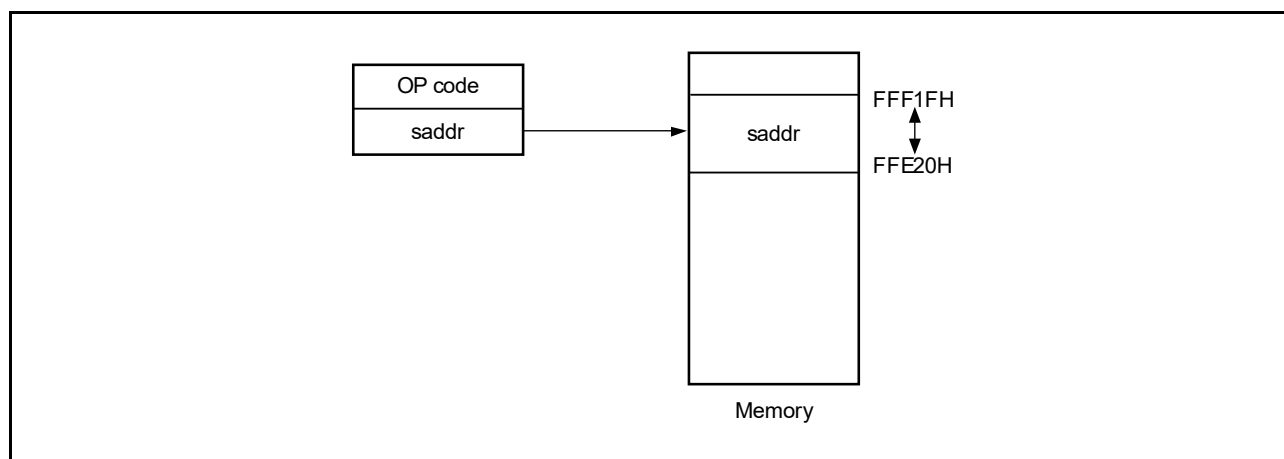
[Function]

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 - 22 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

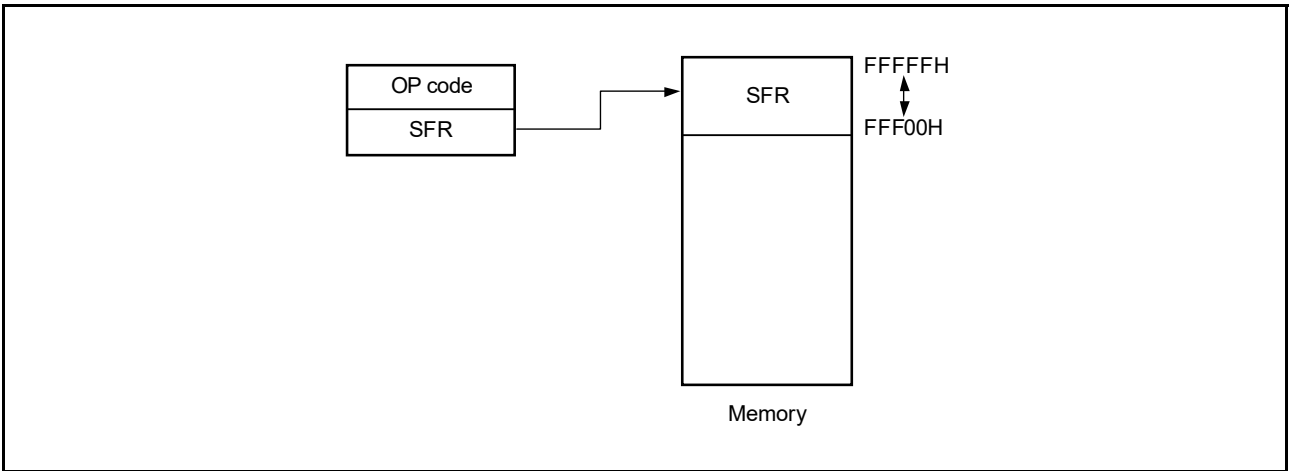
[Function]

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 - 23 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 - 24 Example of [DE], [HL]

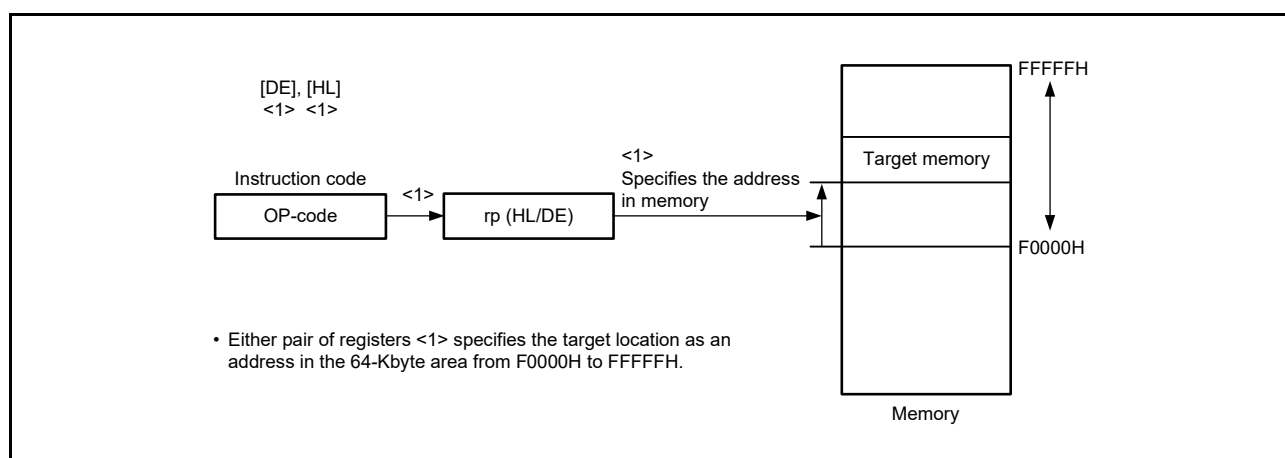
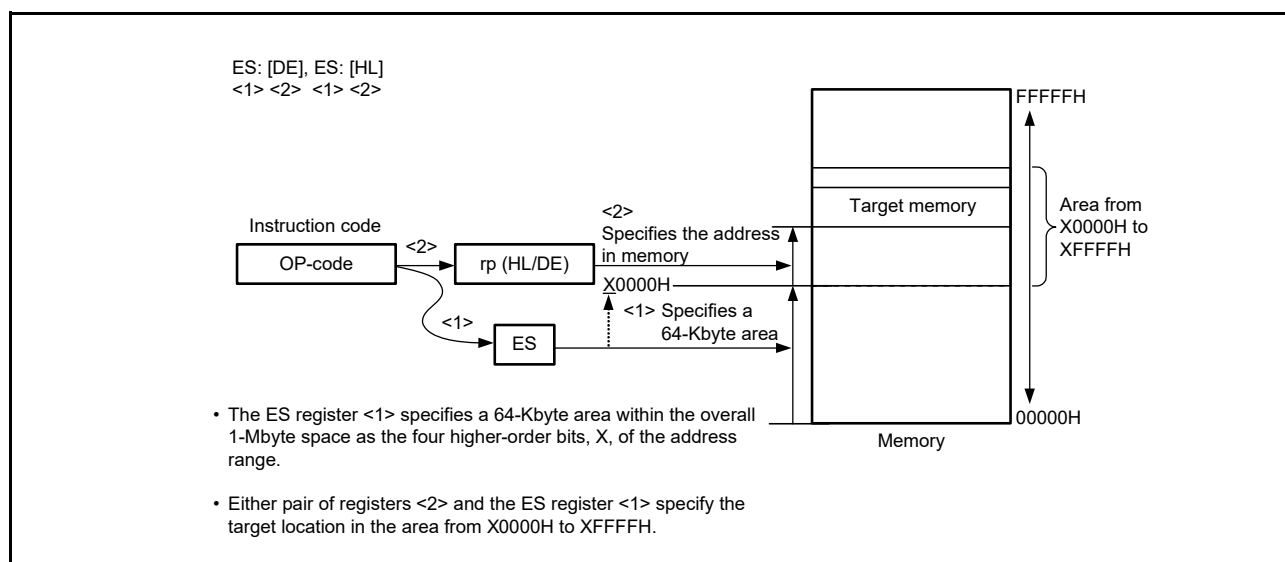


Figure 3 - 25 Example of ES:[DE], ES:[HL]



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 - 26 Example of [SP + byte]

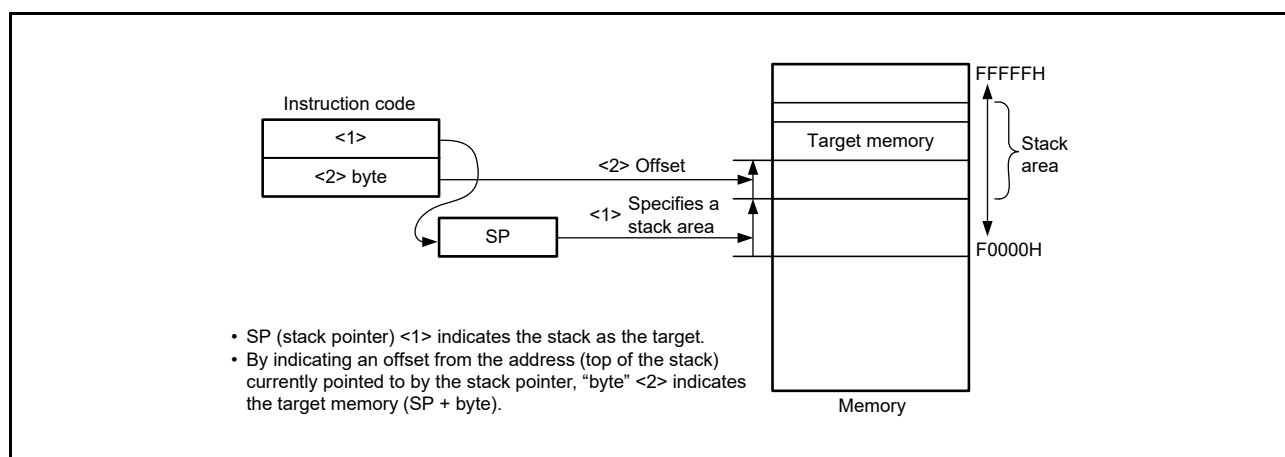


Figure 3 - 27 Example of [HL + byte], [DE + byte]

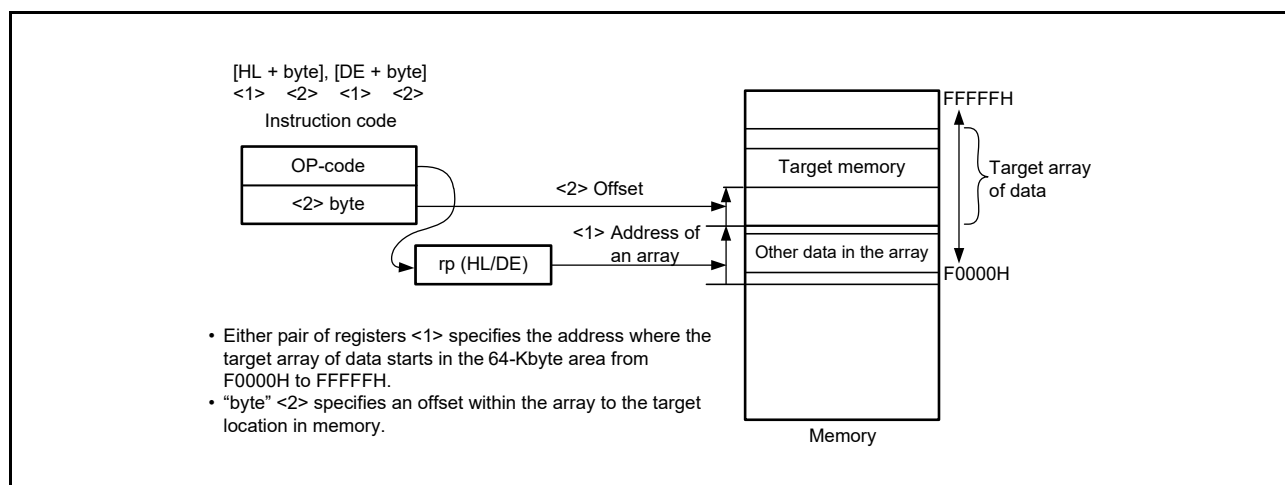


Figure 3 - 28 Example of word[B], word[C]

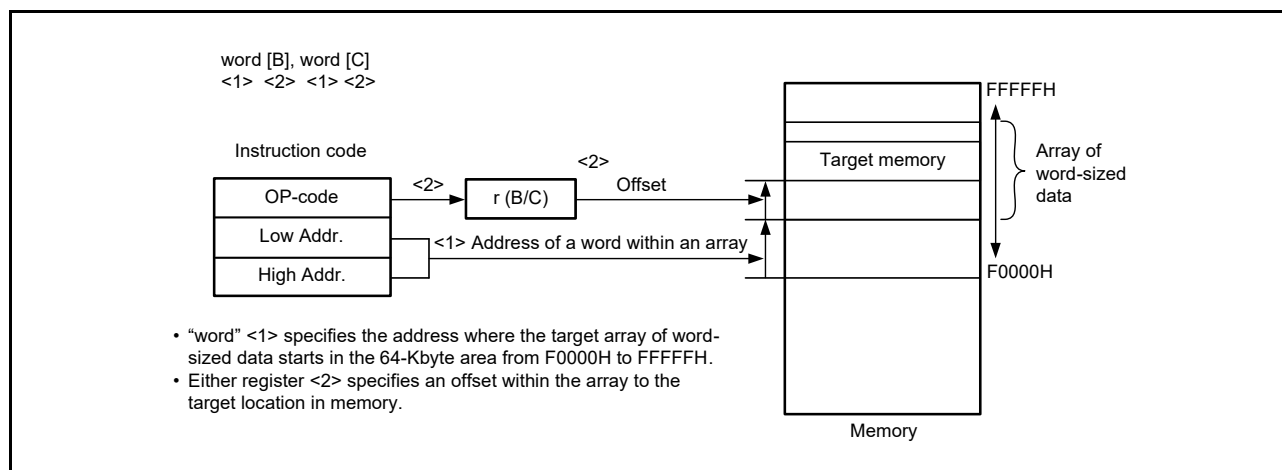


Figure 3 - 29 Example of word[BC]

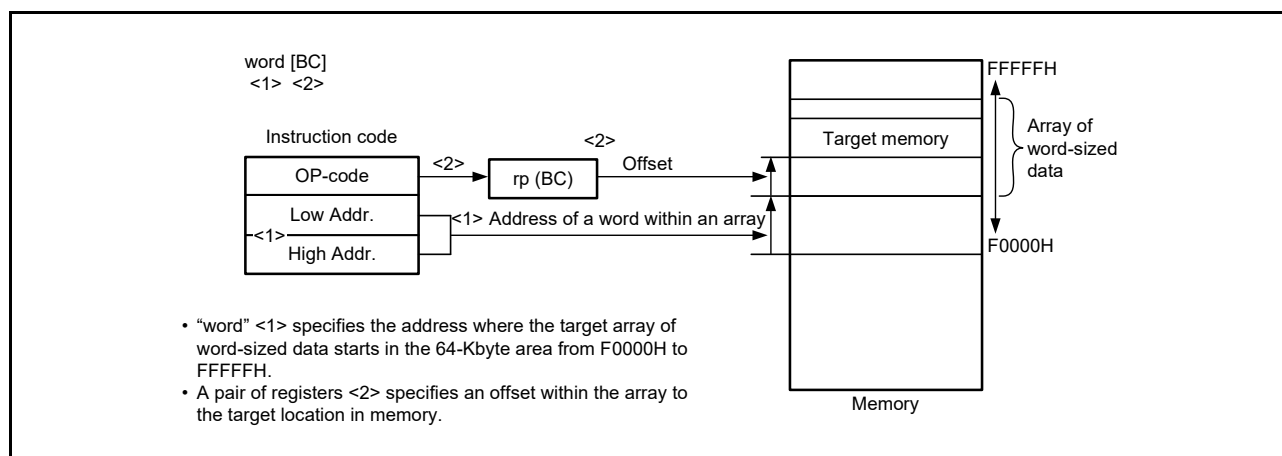


Figure 3 - 30 Example of ES:[HL + byte], ES:[DE + byte]

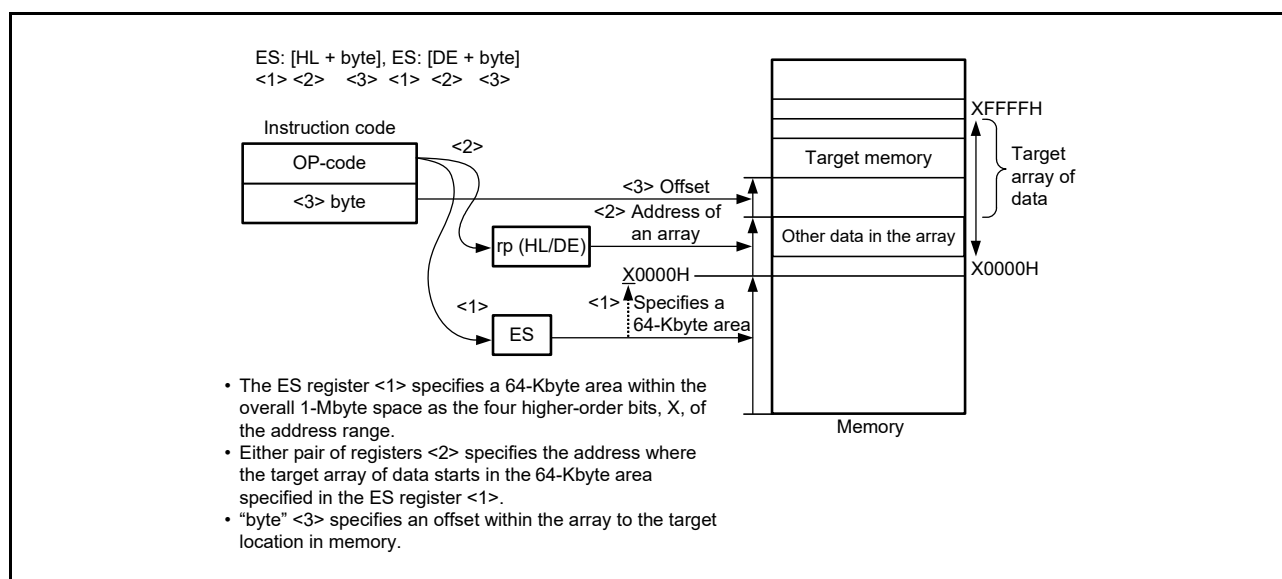


Figure 3 - 31 Example of ES:word[B], ES:word[C]

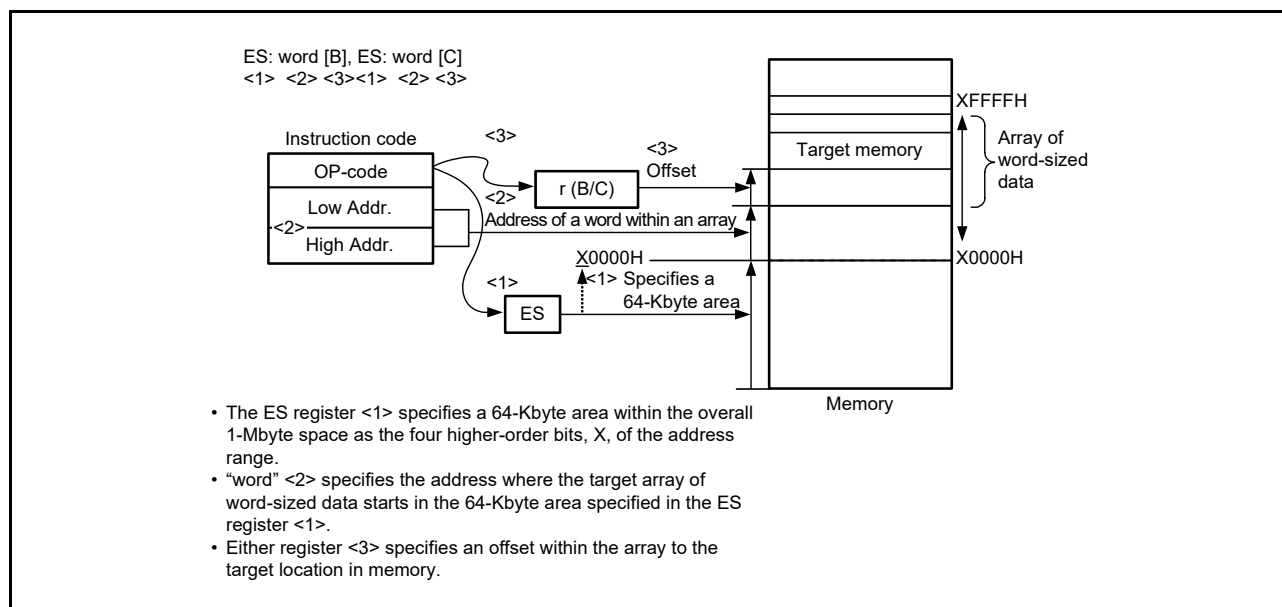
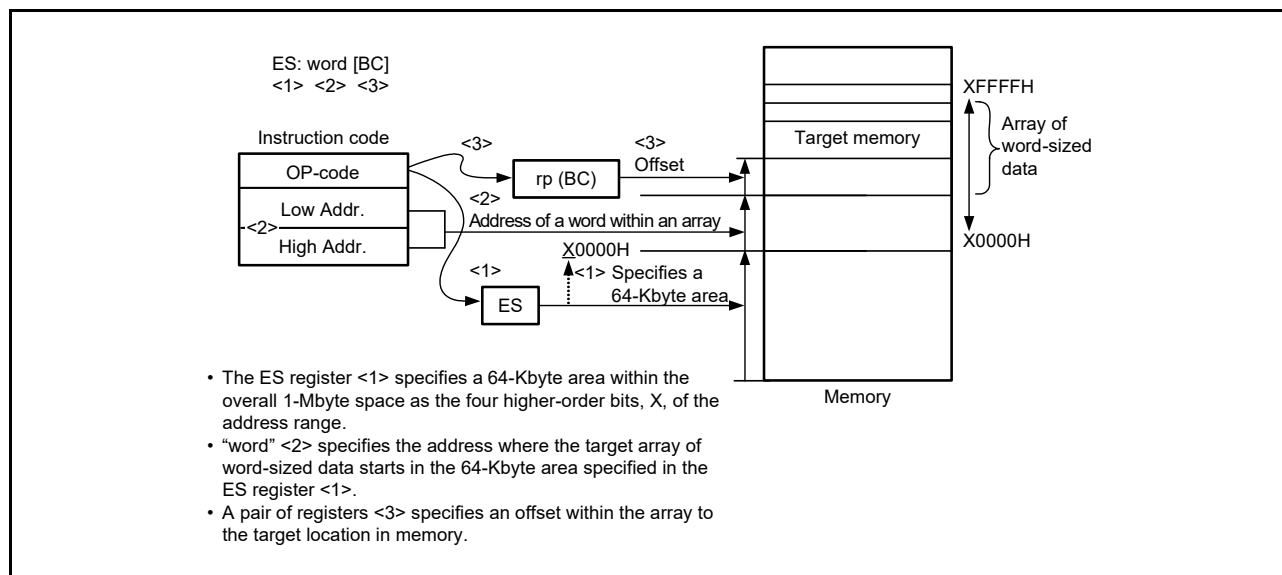


Figure 3 - 32 Example of ES:word[BC]



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 - 33 Example of [HL + B], [HL + C]

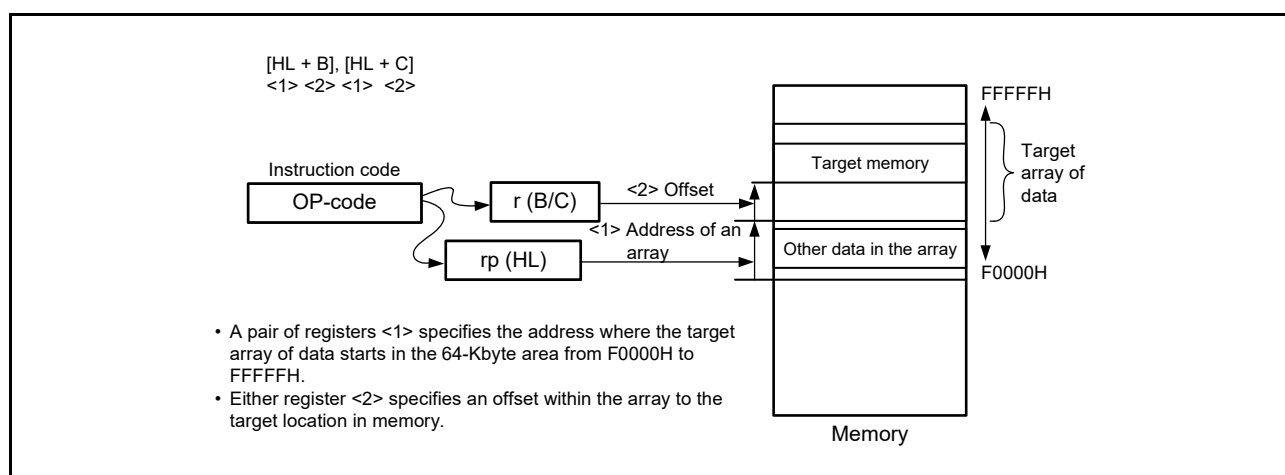
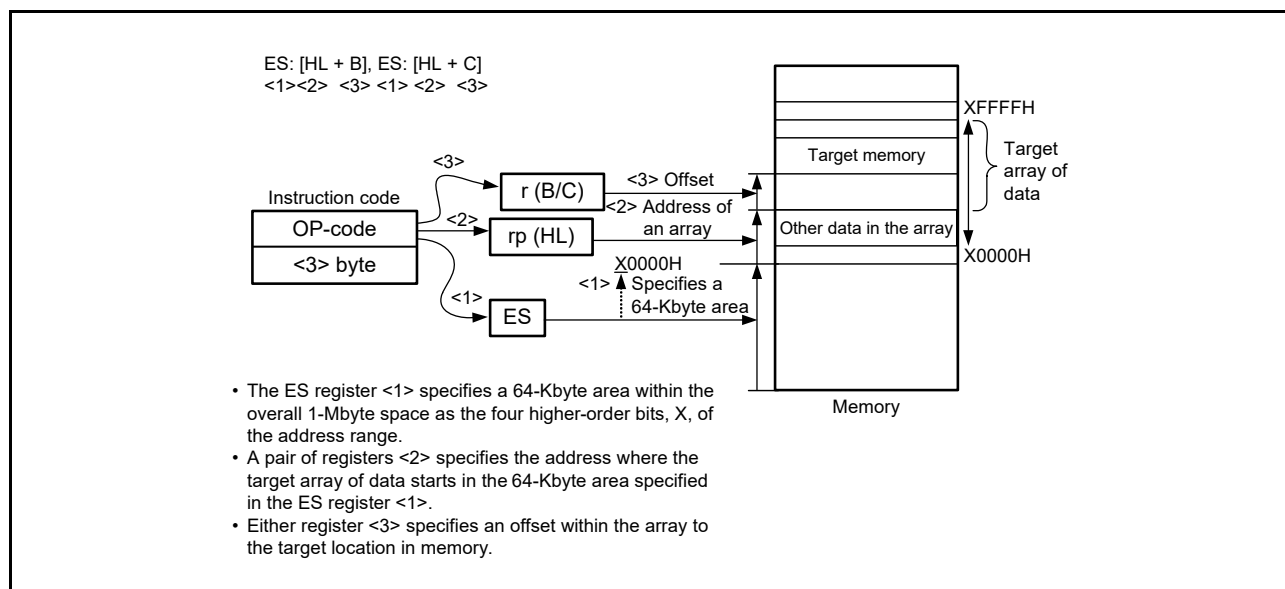


Figure 3 - 34 Example of ES:[HL + B], ES:[HL + C]



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 - 35** to **Figure 3 - 40**.

Figure 3 - 35 Example of PUSH rp

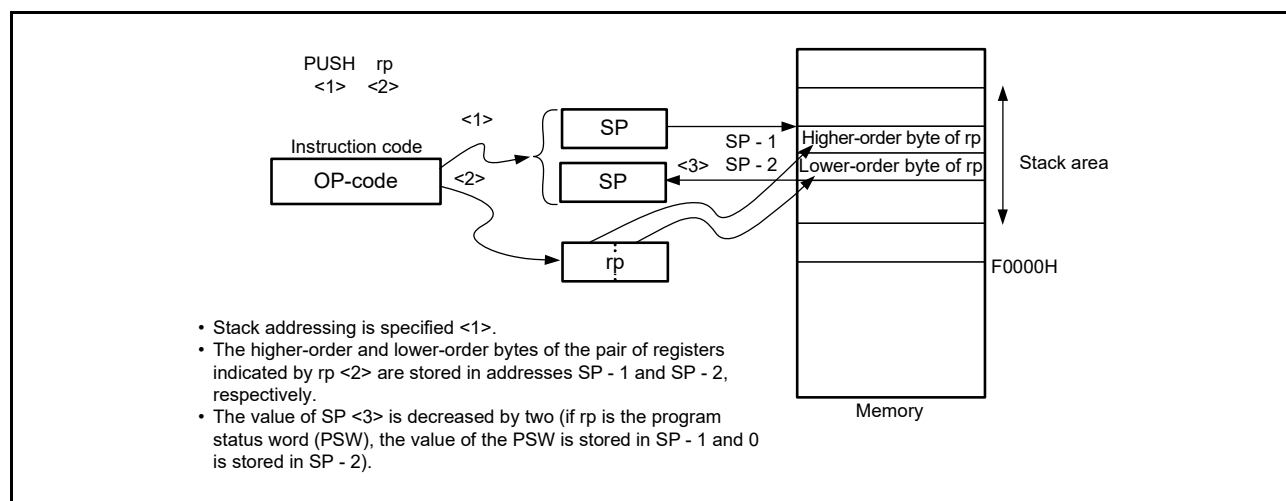


Figure 3 - 36 Example of POP

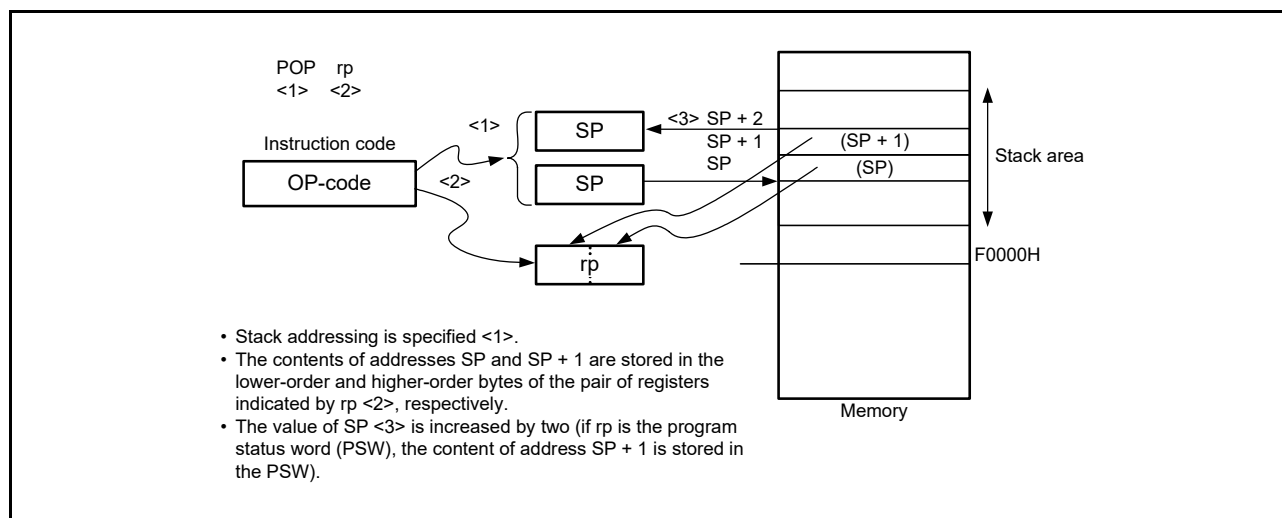


Figure 3 - 37 Example of CALL, CALLT

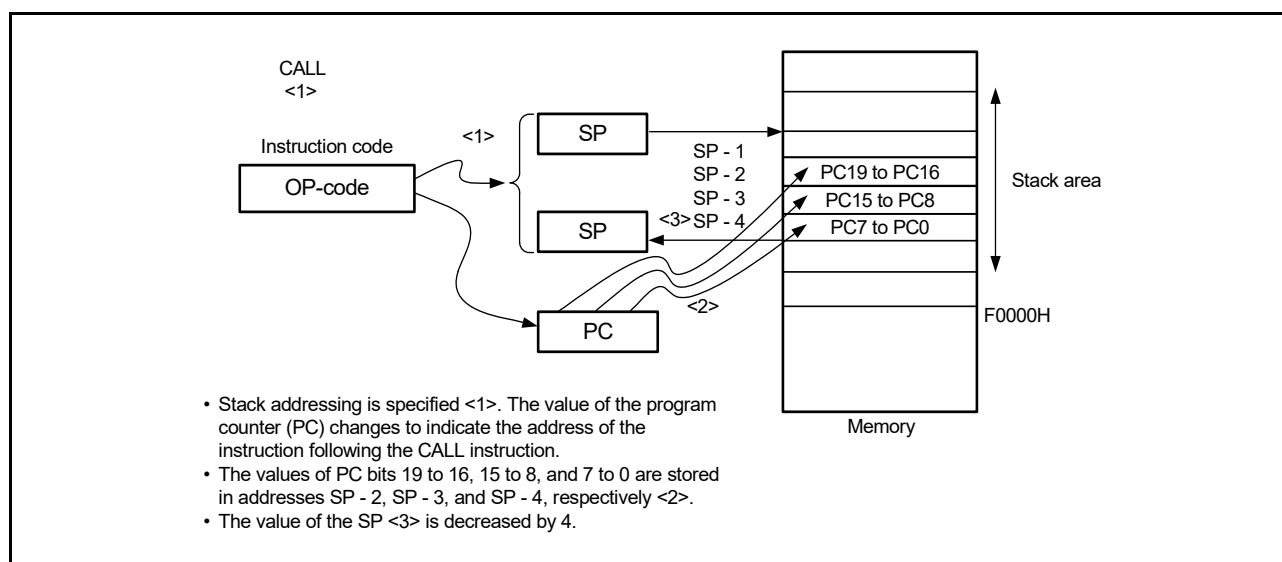


Figure 3 - 38 Example of RET

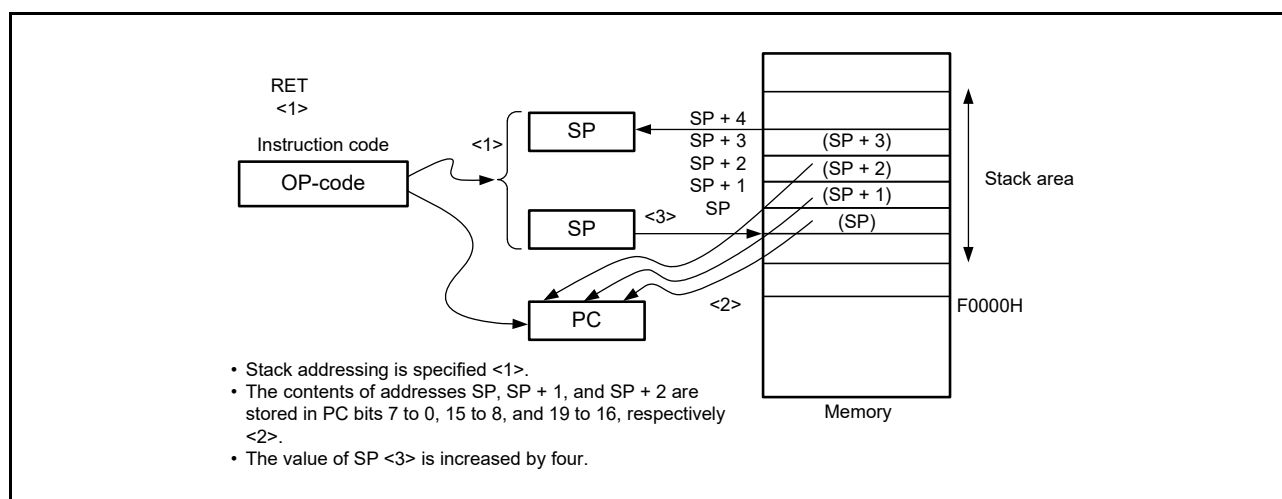


Figure 3 - 39 Example of Interrupt, BRK

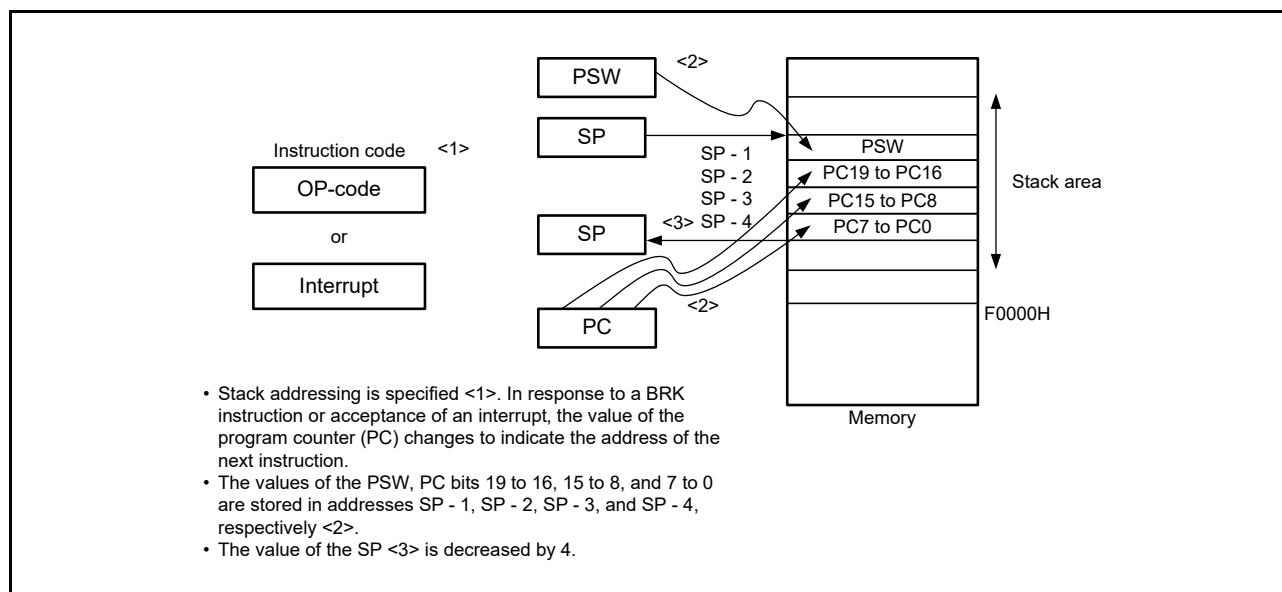
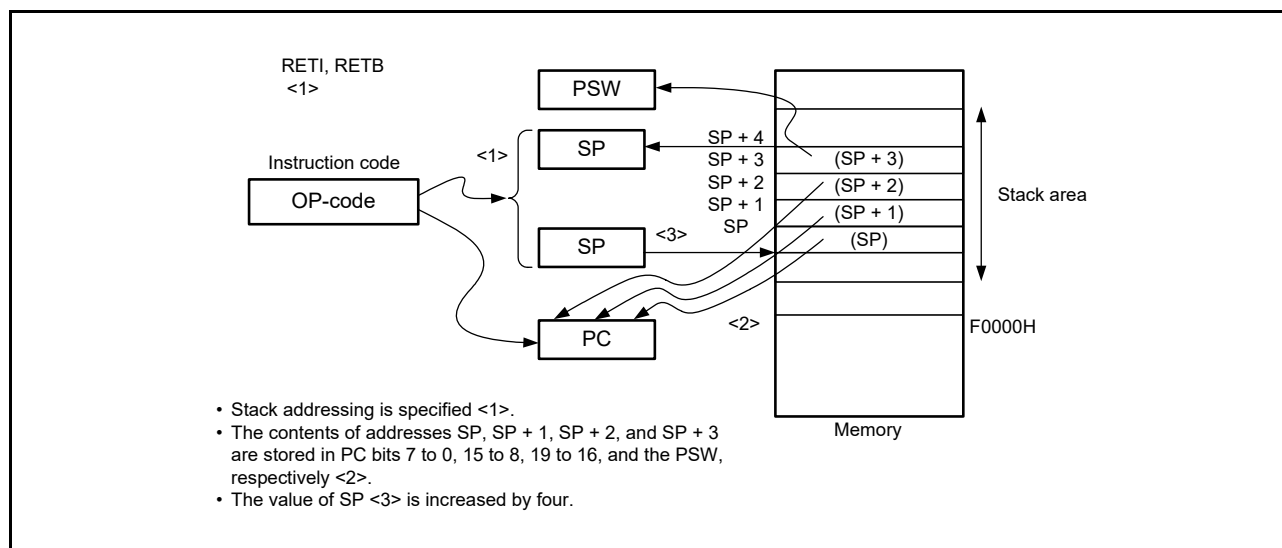


Figure 3 - 40 Example of RETI, RETB



Section 4 Port Functions

4.1 Port Functions

The RL78/L23 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

Item	Configuration
Control registers	Port mode registers (PM0 to PM9, PM12, and PM14) Port registers (P0 to P9 and P12 to P14) Pull-up resistor option registers (PU0 to PU9, PU12, and PU14) Port input mode registers (PIM0 to PIM9, PIM12, and PIM14) Port output mode registers (POM0 to POM9, and POM12 to POM14) Port digital input disable registers (PDIDIS0 to PDIDIS9, and PDIDIS12 to PDIDIS14) Port mode control A registers (PMCA1, PMCA2, and PMCA4) Port mode control T registers (PMCT0, PMCT3, PMCT5, PMCT7, PMCT9, and PMCT12) Port mode control E registers (PMCE0, PMCE1, PMCE6, and PMCE7) Peripheral I/O redirection registers (PIOR0 to PIOR13) Output current control enable register (CCDE) Output current select registers (CCS0, and CCS4 to CCS7) 40-mA port output control register (PTDC) Port function output enable registers (PFOE0 and PFOE1) Port mode select register (PMS) LCD port function registers (PFSEG0 to PFSEG7) LCD input switch control register (ISCLCD)
Port	44-pin products Total: 40 (CMOS I/O: 35 (N-ch open-drain I/O [V_{DD} withstand voltage]: 24, controlled current drive port: 4), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 2) 48-pin products Total: 44 (CMOS I/O: 39 (N-ch open-drain I/O [V_{DD} withstand voltage]: 27, controlled current drive port: 5), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 2) 52-pin products Total: 48 (CMOS I/O: 43 (N-ch open-drain I/O [V_{DD} withstand voltage]: 29, controlled current drive port: 6), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 2) 64-pin products Total: 60 (CMOS I/O: 53 (N-ch open-drain I/O [V_{DD} withstand voltage]: 35, controlled current drive port: 8), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 4) 80-pin products Total: 76 (CMOS I/O: 68 (N-ch open-drain I/O [V_{DD} withstand voltage]: 42, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) 100-pin products Total: 95 (CMOS I/O: 87 (N-ch open-drain I/O [V_{DD} withstand voltage]: 57, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4)
Pull-up resistor	44-pin products: 33 in total 48-pin products: 37 in total 52-pin products: 41 in total 64-pin products: 51 in total 80-pin products: 66 in total 100-pin products: 85 in total

4.2.1 Port 0

Port 0 is an I/O port with output latches. The pins of port 0 can be set as inputs or outputs in 1-bit units by port mode register 0 (PM0). When the P00 to P07 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 0 (PU0). Input to the P02, P03, P05, and P06 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 0 (PIM0). Output from the P00 to P07 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 0 (POM0). To use the P00 to P07 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function registers 6 and 7 (PFSEG6 and PFSEG7). This is also specifiable in 1-bit units. This port can also be used for comparator output, capacitance measurement, external interrupt request input, clock output/buzzer output, data I/O and clock I/O through the serial interface, timer I/O, logic and event link controller I/O, UART data transmission for the external device connection used during flash memory programming, output of segment signals for the LCD controller/driver, and external signal sampler phase detection input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P00 to P07 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.2 Port 1

Port 1 is an I/O port with output latches. The pins of port 1 can be set as inputs or outputs in 1-bit units by port mode register 1 (PM1). When the P10 to P17 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 1 (PU1). Input to the P11, P16, and P17 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 1 (PIM1). Output from the P10, P11, P16, and P17 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 1 (POM1). To use the P10 to P14 pins as digital I/O port pins, set “digital I/O” by using port mode control A register 1 (PMCA1) and make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function register 5 (PFSEG5). Both register settings are specifiable in 1-bit units. This port can also be used for analog input for the A/D converter, comparator output, external interrupt request input, clock output/buzzer output, realtime clock correction clock output, data I/O and clock I/O through the serial interface, timer I/O, logic and event link controller I/O, UART data reception for the external device connection used during flash memory programming, output of segment signals for the LCD controller/driver, and external signal sampler phase detection input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P10 to P14 pins are set to the analog input mode and the P15 to P17 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.3 Port 2

Port 2 is an I/O port with output latches. The pins of port 2 can be set as inputs or outputs in 1-bit units by port mode register 2 (PM2). When the P22 to P27 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 2 (PU2). Input to the P22, P26, and P27 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 2 (PIM2). Output from the P22, P26, and P27 pins can be set to the N-ch open-drain output mode (withstand voltage of V_{DD}) in 1-bit units by port output mode register 2 (POM2). To use the P22 to P27 pins as digital I/O port pins, set “digital I/O” by using port mode control A register 2 (PMCA2) and make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function registers 4 and 5 (PFSEG4 and PFSEG5). Both register settings are specifiable in 1-bit units. This port can also be used for analog input for the A/D converter, D/A converter output, external interrupt request input, clock output/buzzer output, data I/O and clock I/O through the serial interface, timer I/O, logic and event link controller input, power supply for battery backup, positive and negative reference voltage inputs of the A/D converter, output of segment signals for the LCD controller/driver, and external signal sampler phase detection input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P20 to P27 pins are set to the analog input mode following a reset.

4.2.4 Port 3

Port 3 is an I/O port with output latches. The pins of port 3 can be set as inputs or outputs in 1-bit units by port mode register 3 (PM3). When the P30 to P35 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 3 (PU3). Input to the P30 and P34 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 3 (PIM3). Output from the P33 to P35 pins can be set to the N-ch open-drain output mode (withstand voltage of V_{DD}) in 1-bit units by port output mode register 3 (POM3). To use the P30 to P35 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function registers 3 and 4 (PFSEG3 and PFSEG4). This is also specifiable in 1-bit units. This port can also be used for comparator output, capacitance measurement, external interrupt request input, remote controller output, realtime clock correction clock output, data I/O and clock I/O through the serial interface, timer I/O, and output of segment signals for the LCD controller/driver. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P30 to P35 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.5 Port 4

Port 4 is an I/O port with output latches. The pins of port 4 can be set as inputs or outputs in 1-bit units by port mode register 4 (PM4). When the P40 to P47 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 4 (PU4). Input to the P40 to P47 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 4 (PIM4). Output from the P40 to P47 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 4 (POM4). To use the P46 and P47 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function register 4 (PFSEG4). This is also specifiable in 1-bit units. This port can also be used for analog voltage input for the comparator, reference voltage input for the comparator, external interrupt request input, clock output/buzzer output, data I/O and clock I/O through the serial interface, timer I/O, data I/O for flash memory programmer or debugger, output of segment signals for the LCD controller/driver, and external signal sampler clock output. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P40 and P41 pins are set to the input mode, the P42 to P45 pins are set to the analog input mode, and the P46 and P47 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.6 Port 5

Port 5 is an I/O port with output latches. The pins of port 5 can be set as inputs or outputs in 1-bit units by port mode register 5 (PM5). When the P50 to P57 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 5 (PU5). Input to the P51, P53, and P54 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 5 (PIM5). Output from the P50 and P52 to P54 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 5 (POM5). To use the P50 to P57 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function registers 1 and 2 (PFSEG1 and PFSEG2). This is also specifiable in 1-bit units. This port can also be used for capacitance measurement, external interrupt request input, clock output/buzzer output, remote controller output, data I/O and clock I/O through the serial interface, timer I/O, and output of segment signals for the LCD controller/driver. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P50 to P57 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.7 Port 6

Port 6 is an I/O port with output latches. The pins of port 6 can be set as inputs or outputs in 1-bit units by port mode register 6 (PM6). When the P64 to P67 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 6 (PU6). Output from the P60 to P63 pins is N-ch open-drain output (withstand voltage of 6 V). Output from the P60 to P63 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE). Input to the P64 to P66 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 6 (PIM6). Output from the P64 to P66 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 6 (POM6). This port can also be used as the controlled current drive port, external interrupt request input, remote controller output, data I/O and clock I/O through the serial interface, timer I/O, and logic and event link controller output. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P60 to P67 pins are set to the input mode following a reset.

4.2.8 Port 7

Port 7 is an I/O port with output latches. The pins of port 7 can be set as inputs or outputs in 1-bit units by port mode register 7 (PM7). When the pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 7 (PU7). Input to the P75 and P76 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 7 (PIM7). Output from the P75 to P77 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 7 (POM7). Output from the P74 to P77 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE). To use the P70 to P77 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function registers 2 and 3 (PFSEG2 and PFSEG3). This is also specifiable in 1-bit units. This port can also be used for capacitance measurement, controlled current drive port, external interrupt request input, key interrupt input, remote controller output, realtime clock correction clock output, data I/O and clock I/O through the serial interface, timer I/O, logic and event link controller I/O, output of segment signals for the LCD controller/driver, and external signal sampler clock output. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P70 to P77 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.9 Port 8

Port 8 is an I/O port with output latches. The pins of port 8 can be set as inputs or outputs in 1-bit units by port mode register 8 (PM8). When the pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 8 (PU8). Input to the P80 to P86 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 8 (PIM8). Output from the P80 to P87 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 8 (POM8). This port can also be used for comparator output, external interrupt request input, clock output/buzzer output, data I/O and clock I/O through the serial interface, timer I/O, and supply of the voltage for driving the LCD. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P80 to P84 pins are set to the input mode, and the P85 to P87 pins are set to the digital input invalid mode^{Note} following a reset.

Note "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.10 Port 9

Port 9 is an I/O port with output latches. The pins of port 9 can be set as inputs or outputs in 1-bit units by port mode register 9 (PM9). When the pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 9 (PU9). Input to the P93, P94, P96, and P97 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 9 (PIM9). Output from the P93 to P97 pins can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 9 (POM9). To use the P90 to P97 pins as digital I/O port pins, make the setting for "using the pins as port pins (other than as segment output pins)" by using LCD port function registers 0 and 1 (PFSEG0 and PFSEG1). This is also specifiable in 1-bit units. This port can also be used for capacitance measurement, external interrupt request input, clock output/buzzer output, data I/O and clock I/O through the serial interface, timer I/O, and outputs of segment and common signals for the LCD controller/driver. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P90 to P97 pins are set to the digital input invalid mode^{Note} following a reset.

Note "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.11 Port 12

The P121, P122, and P125 to P127 pins operate as a 5-bit I/O port with output latches. The pins can be set as inputs or outputs in 1-bit units by port mode register 12 (PM12). When the pins are to be used as input pins, the use of on-chip pull-up resistors can be specified by pull-up resistor option register 12 (PU12). Input to the P125 pin can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 12 (PIM12). Output from the P125 pin can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 12 (POM12). The P123 and P124 pins operate as a 2-bit input-only port. This port can also be used for comparator output, capacitance measurement, external interrupt request input, clock output/buzzer output, remote controller output, data I/O through the serial interface, logic and event link controller input, supply of the voltage for driving the LCD, connection of a capacitor for the LCD controller/driver, connection of a resonator for the main system clock, external clock input for the main system clock, connection of a resonator for the subsystem 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 state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**. The P121 to P124 pins are set to the input mode, and the P125 to P127 pins are set to the digital input invalid mode^{Note} following a reset.

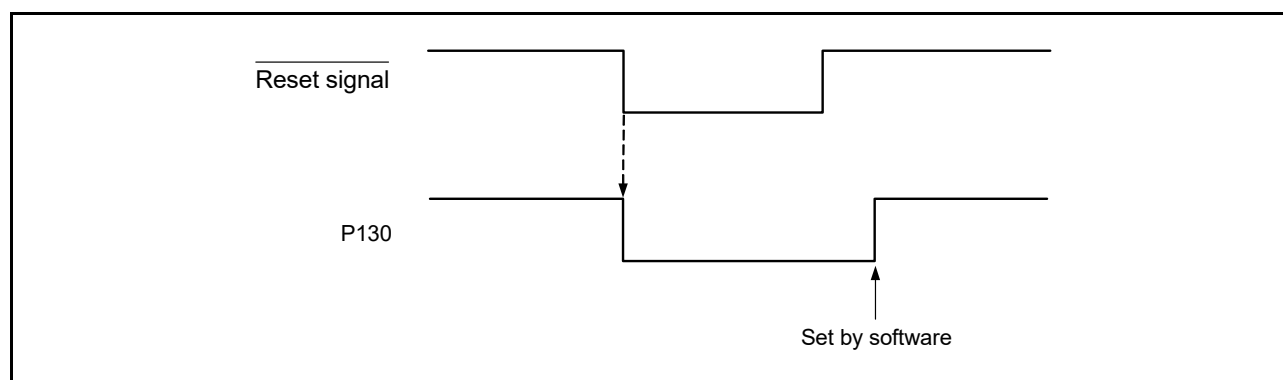
Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

4.2.12 Port 13

The P130 pin operates as a 1-bit output-only port with an output latch. The P137 pin operates as a 1-bit input-only port. To use the P130 pin as a digital output port pin, make the setting for “using the pin as a port pin (other than as a segment output pin)” by using LCD port function register 4 (PFSEG4). This is specifiable in 1-bit units. The P130 pin is set to the output mode following a reset. The P137 pin is fixed to the input mode. Output from the P130 pin can be set to the N-ch open-drain output mode (withstand voltage of VDD) in 1-bit units by port output mode register 13 (POM13). This port can also be used for external interrupt request input, data output through the serial interface, logic and event link controller input, timer output, and output of segment signals for the LCD controller/driver. Use the registers shown in **4.3**

Registers for Controlling the Port Function to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5 Correspondence between Register Settings and Port Pin State**.

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.



4.2.13 Port 14

Port 14 is an I/O port with output latches. The pins of port 14 can be set as inputs or outputs in 1-bit units by port mode register 14 (PM14). When the P140 to P147 pins are to be used as input pins, the use of on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 14 (PU14). Input to the P140 to P142 and P144 to P147 pins can be specified as being through a normal input buffer or a TTL input buffer in 1-bit units by configuring port input mode register 14 (PIM14). Output from the P140 and P143 to P147 pins can be set to the N-ch open-drain output mode (withstand voltage of V_{DD}) in 1-bit units by port output mode register 14 (POM14). To use the P141 to P145 pins as digital I/O port pins, make the setting for “using the pins as port pins (other than as segment output pins)” by using LCD port function register 7 (PFSEG7). This is also specifiable in 1-bit units. This port can also be used for comparator output, external interrupt request input, data I/O and clock I/O through the serial interface, and output of segment signals for the LCD controller/driver. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the state of each of the pins. For the correspondence between register settings and pin states, see **Table 4 - 5**

Correspondence between Register Settings and Port Pin State. The P140, P146, and P147 pins are set to the input mode, and the P141 to P145 pins are set to the digital input invalid mode^{Note} following a reset.

Note “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, analog inputs, and LCD outputs are disabled.

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)
- Port mode control E registers (PMCExx)
- Peripheral I/O redirection registers (PIORx)
- Output current control enable register (CCDE)
- Output current select registers (CCSx)
- 40-mA port output control register (PTDC)
- Port function output enable registers (PFOEx)
- Port mode select register (PMS)
- LCD port function registers (PFSEGx)
- LCD input switch control register (ISCLCD)

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, PMCTxx, PMCExx, and PFSEGx Registers and the Bits Implemented in Each Product (1/3)

Port		Bit name									100 pins	80 pins	64 pins	52 pins	48 pins	44 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register	PFSEGx register					
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	—	—	PMCE00	PFSEG43	✓	✓	✓	✓	✓
	1	PM01	P01	PU01	—	POM01	PDIDIS01	—	—	—	PFSEG44	✓	✓	✓	—	—
	2	PM02	P02	PU02	PIM02	POM02	PDIDIS02	—	—	—	PFSEG45	✓	✓	✓	—	—
	3	PM03	P03	PU03	PIM03	POM03	PDIDIS03	—	—	—	PFSEG46	✓	✓	✓	—	—
	4	PM04	P04	PU04	—	POM04	PDIDIS04	—	PMCT04	—	PFSEG47	✓	✓	✓	—	—
	5	PM05	P05	PU05	PIM05	POM05	PDIDIS05	—	PMCT05	PMCE05	PFSEG48	✓	✓	✓	✓	✓
	6	PM06	P06	PU06	PIM06	POM06	PDIDIS06	—	PMCT06	—	PFSEG49	✓	✓	✓	✓	✓
Port 1	7	PM07	P07	PU07	—	POM07	PDIDIS07	—	PMCT07	—	PFSEG50	✓	✓	✓	✓	✓
	0	PM10	P10	PU10	—	POM10	PDIDIS10	PMCA10	—	—	PFSEG35	✓	✓	✓	✓	✓
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	PMCA11	—	—	PFSEG36	✓	✓	✓	✓	✓
	2	PM12	P12	PU12	—	—	—	PMCA12	—	—	PFSEG37	✓	✓	✓	✓	✓
	3	PM13	P13	PU13	—	—	—	PMCA13	—	—	PFSEG38	✓	✓	✓	✓	—
	4	PM14	P14	PU14	—	—	—	PMCA14	—	—	PFSEG39	✓	✓	✓	—	—
	5	PM15	P15	PU15	—	—	—	—	—	—	PFSEG40	✓	✓	✓	—	—
Port 2	6	PM16	P16	PU16	PIM16	POM16	PDIDIS16	—	—	—	PFSEG41	✓	✓	✓	✓	✓
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	—	—	PMCE17	PFSEG42	✓	✓	✓	✓	✓
	0	PM20	P20	—	—	—	—	PMCA20	—	—	—	✓	✓	✓	✓	✓
	1	PM21	P21	—	—	—	—	PMCA21	—	—	—	✓	✓	✓	✓	✓
	2	PM22	P22	PU22	PIM22	POM22	PDIDIS22	PMCA22	—	—	PFSEG29	✓	✓	✓	✓	✓
	3	PM23	P23	PU23	—	—	—	PMCA23	—	—	PFSEG30	✓	✓	—	—	—
	4	PM24	P24	PU24	—	—	—	PMCA24	—	—	PFSEG31	✓	✓	—	—	—
Port 3	5	PM25	P25	PU25	—	—	—	PMCA25	—	—	PFSEG32	✓	✓	—	—	—
	6	PM26	P26	PU26	PIM26	POM26	PDIDIS26	PMCA26	—	—	PFSEG33	✓	✓	✓	✓	✓
	7	PM27	P27	PU27	PIM27	POM27	PDIDIS27	PMCA27	—	—	PFSEG34	✓	✓	✓	✓	✓
	0	PM30	P30	PU30	—	—	—	—	PMCT30	—	PFSEG20	✓	✓	✓	—	—
	1	PM31	P31	PU31	—	—	—	—	PMCT31	—	PFSEG21	✓	✓	✓	—	—
	2	PM32	P32	PU32	—	—	—	—	PMCT32	—	PFSEG22	✓	—	—	—	—
	3	PM33	P33	PU33	PIM33	POM33	PDIDIS33	—	PMCT33	—	PFSEG23	✓	—	—	—	—
Port 4	4	PM34	P34	PU34	PIM34	POM34	PDIDIS34	—	PMCT34	—	PFSEG24	✓	✓	—	—	—
	5	PM35	P35	PU35	—	POM35	PDIDIS35	—	PMCT35	—	PFSEG25	✓	✓	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	0	PM40	P40	PU40	PIM40	POM40	PDIDIS40	—	—	—	—	✓	✓	✓	✓	✓
	1	PM41	P41	PU41	PIM41	POM41	PDIDIS41	—	—	—	—	✓	✓	—	—	—
	2	PM42	P42	PU42	PIM42	POM42	PDIDIS42	PMCA42	—	—	—	✓	✓	✓	—	—
Port 5	3	PM43	P43	PU43	PIM43	POM43	PDIDIS43	PMCA43	—	—	—	✓	✓	✓	—	—
	4	PM44	P44	PU44	PIM44	POM44	PDIDIS44	PMCA44	—	—	—	✓	✓	✓	✓	✓
	5	PM45	P45	PU45	PIM45	POM45	PDIDIS45	PMCA45	—	—	—	✓	✓	✓	✓	—
	6	PM46	P46	PU46	PIM46	POM46	PDIDIS46	—	—	—	PFSEG26	✓	✓	—	—	—
	7	PM47	P47	PU47	PIM47	POM47	PDIDIS47	—	—	—	PFSEG27	✓	✓	—	—	—
	0	PM50	P50	PU50	—	POM50	PDIDIS50	—	PMCT50	—	PFSEG04	✓	✓	—	—	—
	1	PM51	P51	PU51	PIM51	—	—	—	PMCT51	—	PFSEG05	✓	✓	—	—	—
Port 5	2	PM52	P52	PU52	—	POM52	PDIDIS52	—	PMCT52	—	PFSEG06	✓	✓	✓	—	—
	3	PM53	P53	PU53	PIM53	POM53	PDIDIS53	—	PMCT53	—	PFSEG07	✓	✓	✓	✓	—
	4	PM54	P54	PU54	PIM54	POM54	PDIDIS54	—	PMCT54	—	PFSEG08	✓	✓	✓	✓	—
	5	PM55	P55	PU55	—	—	—	—	PMCT55	—	PFSEG09	✓	✓	—	—	—
	6	PM56	P56	PU56	—	—	—	—	PMCT56	—	PFSEG10	✓	✓	—	—	—
	7	PM57	P57	PU57	—	—	—	—	PMCT57	—	PFSEG11	✓	✓	✓	✓	✓

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, PMCExx, and PFSEGx Registers and the Bits Implemented in Each Product (2/3)

Port		Bit name										100 pins	80 pins	64 pins	52 pins	48 pins	44 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register	PFSEGx register						
Port 6	0	PM60	P60	—	—	—	—	—	—	PMCE60	—	✓	✓	✓	✓	✓	✓
	1	PM61	P61	—	—	—	—	—	—	PMCE61	—	✓	✓	✓	✓	✓	✓
	2	PM62	P62	—	—	—	—	—	—	—	—	✓	✓	✓	—	—	—
	3	PM63	P63	—	—	—	—	—	—	—	—	✓	✓	✓	—	—	—
	4	PM64	P64	PU64	PIM64	POM64	PDIDIS64	—	—	—	—	✓	—	—	—	—	—
	5	PM65	P65	PU65	PIM65	POM65	PDIDIS65	—	—	—	—	✓	—	—	—	—	—
	6	PM66	P66	PU66	PIM66	POM66	PDIDIS66	—	—	—	—	✓	—	—	—	—	—
Port 7	7	PM67	P67	PU67	—	—	—	—	—	—	—	✓	—	—	—	—	—
	0	PM70	P70	PU70	—	—	—	—	PMCT70	—	PFSEG12	✓	✓	✓	—	—	—
	1	PM71	P71	PU71	—	—	—	—	PMCT71	—	PFSEG13	✓	✓	—	—	—	—
	2	PM72	P72	PU72	—	—	—	—	PMCT72	—	PFSEG14	✓	✓	—	—	—	—
	3	PM73	P73	PU73	—	—	—	—	PMCT73	—	PFSEG15	✓	✓	—	—	—	—
	4	PM74	P74	PU74	—	—	—	—	PMCT74	—	PFSEG16	✓	✓	✓	✓	—	—
	5	PM75	P75	PU75	PIM75	POM75	PDIDIS75	—	PMCT75	—	PFSEG17	✓	✓	✓	✓	✓	—
Port 8	6	PM76	P76	PU76	PIM76	POM76	PDIDIS76	—	PMCT76	PMCE76	PFSEG18	✓	✓	✓	✓	✓	✓
	7	PM77	P77	PU77	—	POM77	PDIDIS77	—	PMCT77	PMCE77	PFSEG19	✓	✓	✓	✓	✓	✓
	0	PM80	P80	PU80	PIM80	POM80	PDIDIS80	—	—	—	—	✓	—	—	—	—	—
	1	PM81	P81	PU81	PIM81	POM81	PDIDIS81	—	—	—	—	✓	—	—	—	—	—
	2	PM82	P82	PU82	PIM82	POM82	PDIDIS82	—	—	—	—	✓	—	—	—	—	—
	3	PM83	P83	PU83	PIM83	POM83	PDIDIS83	—	—	—	—	✓	—	—	—	—	—
	4	PM84	P84	PU84	PIM84	POM84	PDIDIS84	—	—	—	—	✓	—	—	—	—	—
Port 9	5	PM85	P85	PU85	PIM85	POM85	PDIDIS85	—	—	—	—	✓	✓	✓	✓	✓	✓
	6	PM86	P86	PU86	PIM86	POM86	PDIDIS86	—	—	—	—	✓	✓	✓	✓	✓	✓
	7	PM87	P87	PU87	—	POM87	PDIDIS87	—	—	—	—	✓	✓	✓	✓	✓	✓
	0	PM90	P90	PU90	—	—	—	—	PMCT90	—	PFCOM0	✓	✓	✓	✓	✓	✓
	1	PM91	P91	PU91	—	—	—	—	PMCT91	—	PFCOM1	✓	✓	✓	✓	✓	✓
	2	PM92	P92	PU92	—	—	—	—	PMCT92	—	PFCOM2	✓	✓	✓	✓	✓	✓
	3	PM93	P93	PU93	PIM93	POM93	PDIDIS93	—	PMCT93	—	PFCOM3	✓	✓	✓	✓	✓	✓
Port 12	4	PM94	P94	PU94	PIM94	POM94	PDIDIS94	—	PMCT94	—	PFSEG00	✓	✓	✓	✓	✓	✓
	5	PM95	P95	PU95	—	POM95	PDIDIS95	—	PMCT95	—	PFSEG01	✓	✓	✓	✓	✓	✓
	6	PM96	P96	PU96	PIM96	POM96	PDIDIS96	—	PMCT96	—	PFSEG02	✓	✓	✓	✓	✓	✓
	7	PM97	P97	PU97	PIM97	POM97	PDIDIS97	—	PMCT97	—	PFSEG03	✓	✓	✓	✓	✓	✓
	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	1	PM121	P121	PU121	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	2	PM122	P122	PU122	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
Port 13	3	—	P123	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	4	—	P124	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	5	PM125	P125	PU125	PIM125	POM125	PDIDIS125	—	PMCT125	—	—	✓	✓	✓	✓	✓	✓
	6	PM126	P126	PU126	—	—	—	—	PMCT126	—	—	✓	✓	✓	✓	✓	✓
	7	PM127	P127	PU127	—	—	—	—	PMCT127	—	—	✓	✓	✓	✓	✓	✓
	0	—	P130	—	—	POM130	—	—	—	—	PFSEG28	✓	✓	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 13	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS137	—	—	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, PMCExx, and PFSEGx Registers and the Bits Implemented in Each Product (3/3)

Port		Bit name									100 pins	80 pins	64 pins	52 pins	48 pins	44 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register	PFSEGx register					
Port 14	0	PM140	P140	PU140	PIM140	POM140	PDIDIS14 0	—	—	—	—	✓	—	—	—	—
	1	PM141	P141	PU141	PIM141	—	—	—	—	—	PFSEG51	✓	—	—	—	—
	2	PM142	P142	PU142	PIM142	—	—	—	—	—	PFSEG52	✓	—	—	—	—
	3	PM143	P143	PU143	—	POM143	PDIDIS14 3	—	—	—	PFSEG53	✓	—	—	—	—
	4	PM144	P144	PU144	PIM144	POM144	PDIDIS14 4	—	—	—	PFSEG54	✓	—	—	—	—
	5	PM145	P145	PU145	PIM145	POM145	PDIDIS14 5	—	—	—	PFSEG55	✓	—	—	—	—
	6	PM146	P146	PU146	PIM146	POM146	PDIDIS14 6	—	—	—	—	✓	—	—	—	—
	7	PM147	P147	PU147	PIM147	POM147	PDIDIS14 7	—	—	—	—	✓	—	—	—	—

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	PM07	PM06	PM05	PM04	PM03	PM02	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	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62 Note	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29H	FFH	R/W
PM12	PM127	PM126	PM125	1	1	PM122	PM121	1	FFF2CH	FFH	R/W
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9, 12, or 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note In products with 128 or fewer Kbytes of code flash memory and 52 or fewer pins, the circuitry for the pin is implemented, but is only connected to an internal terminal. After release from the reset state, the corresponding bit of the port mode register must be set to 0 by software.

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 input mode, the pin level is read. If data is read in output mode, the output latch value of the port 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 any of the P10 to P14, P20 to P27, and P42 to P45 pins are set as analog function ports, in input mode, 0 is always read (rather than the pin level) from such ports.

Caution If any of the P04 to P07, P30 to P35, P50 to P57, P70 to P77, P90 to P97, and P125 to P127 pins are set as capacitance measurement ports, in input mode, 0 is always read (rather than the pin level) from such ports.

Figure 4 - 2 Format of Port Registers (Pxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	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	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	P67	P66	P65	P64	P63	P62 Note 1	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08H	00H (output latch)	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09H	00H (output latch)	R/W
P12	P127	P126	P125	P124	P123	P122	P121	0	FFF0CH	Undefined	R/W ^{Note 2}
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Note 3	R/W ^{Note 2}
P14	P147	P146	P145	P144	P143	P142	P141	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. In products with 128 or fewer Kbytes of code flash memory and 52 or fewer pins, the circuitry for the pin is implemented, but is only connected to an internal terminal. After release from the reset state, the corresponding bit of the port register must be set to 0 by software.

Note 2. Only P123, P124, and P137 are readable.

Note 3. P137: Undefined
P130: 0 (output latch)

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

Remark m = 0 to 9 or 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	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU2	PU27	PU26	PU25	PU24	PU23	PU22	0	0	F0032H	00H	R/W
PU3	0	0	PU35	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039H	00H	R/W
PU12	PU127	PU126	PU125	0	0	PU122	PU121	0	F003CH	00H	R/W
PU14	PU147	PU146	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0 to 9, 12, or 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	PIM06	PIM05	0	PIM03	PIM02	0	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	0	0	0	0	PIM11	0	F0041H	00H	R/W
PIM2	PIM27	PIM26	0	0	0	PIM22	0	0	F0042H	00H	R/W
PIM3	0	0	0	PIM34	PIM33	0	0	0	F0043H	00H	R/W
PIM4	PIM47	PIM46	PIM45	PIM44	PIM43	PIM42	PIM41	PIM40	F0044H	00H	R/W
PIM5	0	0	0	PIM54	PIM53	0	PIM51	0	F0045H	00H	R/W
PIM6	0	PIM66	PIM65	PIM64	0	0	0	0	F0046H	00H	R/W
PIM7	0	PIM76	PIM75	0	0	0	0	0	F0047H	00H	R/W
PIM8	0	PIM86	PIM85	PIM84	PIM83	PIM82	PIM81	PIM80	F0048H	00H	R/W
PIM9	PIM97	PIM96	0	PIM94	PIM93	0	0	0	F0049H	00H	R/W
PIM12	0	0	PIM125	0	0	0	0	0	F004CH	00H	R/W
PIM14	PIM147	PIM146	PIM145	PIM144	0	PIM142	PIM141	PIM140	F004EH	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0 to 9, 12, or 14; n = 0 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 (VDD withstand voltage) mode can be selected during serial communication with an external device operating at a different voltage, and for the SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, and SDA31 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 (VDD withstand voltage) 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	POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00	F0050H	00H	R/W
POM1	POM17	POM16	0	0	0	0	POM11	POM10	F0051H	00H	R/W
POM2	POM17	POM26	0	0	0	POM22	0	0	F0052H	00H	R/W
POM3	0	0	POM35	POM34	POM33	0	0	0	F0053H	00H	R/W
POM4	POM47	POM46	POM45	POM44	POM43	POM42	POM41	POM40	F0054H	00H	R/W
POM5	0	0	0	POM54	POM53	POM52	0	POM50	F0055H	00H	R/W
POM6	0	POM66	POM65	POM64	0	0	0	0	F0056H	00H	R/W
POM7	POM77	POM76	POM75	0	0	0	0	0	F0057H	00H	R/W
POM8	POM87	POM86	POM85	POM84	POM83	POM82	POM81	POM80	F0058H	00H	R/W
POM9	POM97	POM96	POM95	POM94	POM93	0	0	0	F0059H	00H	R/W
POM12	0	0	POM125	0	0	0	0	0	F005CH	00H	R/W
POM13	0	0	0	0	0	0	0	POM130	F005DH	00H	R/W
POM14	POM147	POM146	POM145	POM144	POM143	0	0	POM140	F005EH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0 to 9 or 12 to 14; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (VDD withstand voltage) 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 to be used, power consumption can be lowered by setting the corresponding bit of the PDIDISxx register 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	PDIDIS 07	PDIDIS 06	PDIDIS 05	PDIDIS 04	PDIDIS 03	PDIDIS 02	PDIDIS 01	PDIDIS 00	F02B0H	00H	R/W
PDIDIS1	PDIDIS 17	PDIDIS 16	0	0	0	0	PDIDIS 11	PDIDIS 10	F02B1H	00H	R/W
PDIDIS2	PDIDIS 27	PDIDIS 26	0	0	0	PDIDIS 22	0	0	F02B2H	00H	R/W
PDIDIS3	0	0	PDIDIS 35	PDIDIS 34	PDIDIS 33	0	0	0	F02B3H	00H	R/W
PDIDIS4	PDIDIS 47	PDIDIS 46	PDIDIS 45	PDIDIS 44	PDIDIS 43	PDIDIS 42	PDIDIS 41	PDIDIS 40	F02B4H	00H	R/W
PDIDIS5	0	0	0	PDIDIS 54	PDIDIS 53	PDIDIS 52	0	PDIDIS 50	F02B5H	00H	R/W
PDIDIS6	0	PDIDIS 66	PDIDIS 65	PDIDIS 64	0	0	0	0	F02B6H	00H	R/W
PDIDIS7	PDIDIS 77	PDIDIS 76	PDIDIS 75	0	0	0	0	0	F02B7H	00H	R/W
PDIDIS8	PDIDIS 87	PDIDIS 86	PDIDIS 85	PDIDIS 84	PDIDIS 83	PDIDIS 82	PDIDIS 81	PDIDIS 80	F02B8H	00H	R/W
PDIDIS9	PDIDIS 97	PDIDIS 96	PDIDIS 95	PDIDIS 94	PDIDIS 93	0	0	0	F02B9H	00H	R/W
PDIDIS12	0	0	PDIDIS 125	0	0	0	0	0	F02BCH	00H	R/W
PDIDIS13	PDIDIS 137	0	0	0	0	0	0	0	F02BDH	00H	R/W
PDIDIS14	PDIDIS 147	PDIDIS 146	PDIDIS 145	PDIDIS 144	PDIDIS 143	0	0	PDIDIS 140	F02BEH	00H	R/W

PDIDISmn	Setting of input buffers (m = 0 to 9 or 12 to 14; n = 0 to 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 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
PMCA1	1	1	1	PMCA1 4	PMCA1 3	PMCA1 2	PMCA1 1	PMCA1 0	F0061H	FFH	R/W
PMCA2	PMCA2 7	PMCA2 6	PMCA2 5	PMCA2 4	PMCA2 3	PMCA2 2	PMCA2 1	PMCA2 0	F0062H	FFH	R/W
PMCA4	1	1	PMCA4 5	PMCA4 4	PMCA4 3	PMCA4 2	1	1	F0064H	FFH	R/W

PMCAmn	Selection of whether the Pmn pins are to be used for digital I/O or the analog function (m = 1, 2, or 4; n = 0 to 7)
0	Digital I/O
1	Analog function

Caution 1. For the ports which are set to the analog function by using the PMCAxx registers, select input mode by using port mode registers 1, 2, and 4 (PM1, PM2, and PM4).

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	PMCT0 7	PMCT0 6	PMCT0 5	PMCT0 4	0	0	0	0	F0260H	00H	R/W
PMCT3	0	0	PMCT3 5	PMCT3 4	PMCT3 3	PMCT3 2	PMCT3 1	PMCT3 0	F0263H	00H	R/W
PMCT5	PMCT5 7	PMCT5 6	PMCT5 5	PMCT5 4	PMCT5 3	PMCT5 2	PMCT5 1	PMCT5 0	F0265H	00H	R/W
PMCT7	PMCT7 7	PMCT7 6	PMCT7 5	PMCT7 4	PMCT7 3	PMCT7 2	PMCT7 1	PMCT7 0	F0267H	00H	R/W
PMCT9	PMCT9 7	PMCT9 6	PMCT9 5	PMCT9 4	PMCT9 3	PMCT9 2	PMCT9 1	PMCT9 0	F0269H	00H	R/W
PMCT12	PMCT1 27	PMCT1 26	PMCT1 25	0	0	0	0	0	F026CH	00H	R/W

PMCTmn	Selection of whether the Pmn pins are to be used for digital I/O or the capacitance measurement function (m = 0, 3, 5, 7, 9, or 12; n = 0, 1, or 5 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 Port mode control E registers (PMCExx)

The PMCExx registers specify the digital I/O or ELCL output function in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMCExx register following a reset is 00H.

Figure 4 - 9 Format of Port Mode Control E Registers (PMCExx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCE0	0	0	PMCE0 5	0	0	0	0	PMCE0 0	F0280H	00H	R/W
PMCE1	PMCE1 7	0	0	0	0	0	0	0	F0281H	00H	R/W
PMCE6	0	0	0	0	0	0	PMCE6 1	PMCE6 0	F0286H	00H	R/W
PMCE7	PMCE7 7	PMCE7 6	0	0	0	0	0	0	F0287H	00H	R/W

PMCEmn	Selection of whether the Pmn pins are to be used for digital I/O or the ELCL output function (m = 0, 1, 6, or 7; n = 0, 1, or 5 to 7)
0	Digital I/O
1	ELCL output function

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

4.3.10 Peripheral I/O redirection registers (PIORx)

The PIORx registers are used to specify whether to enable or disable the peripheral I/O redirection function. This function is used to switch alternate functions that are multiplexed with port pins to replace the port functions of the pins. Use the corresponding PIORx register to assign the target alternate function for redirection to a selected port pin before enabling the function. Note that the redirection setting cannot be changed once the operation of a function has been enabled. The PIORx registers can be set by an 8-bit memory manipulation instruction. The value of each PIORx register following a reset is 00H.

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (1/6)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR0	0	PIOR0 6	PIOR0 5	PIOR0 4	PIOR0 3	PIOR0 2	PIOR0 1	PIOR0 0	F04E0H	00H	R/W
PIOR1	PIOR1 7	PIOR1 6	PIOR1 5	PIOR1 4	PIOR1 3	PIOR1 2	PIOR1 1	PIOR1 0	F04E1H	00H	R/W
PIOR2	0	0	0	0	PIOR2 3	PIOR2 2	PIOR2 1	PIOR2 0	F04E2H	00H	R/W
PIOR3	0	PIOR3 6	PIOR3 5	PIOR3 4	PIOR3 3	PIOR3 2	PIOR3 1	PIOR3 0	F04E3H	00H	R/W
PIOR4	PIOR4 7	PIOR4 6	0	0	0	0	PIOR4 1	PIOR4 0	F04E4H	00H	R/W
PIOR5	PIOR5 7	PIOR5 6	PIOR5 5	PIOR5 4	PIOR5 3	PIOR5 2	PIOR5 1	PIOR5 0	F04E5H	00H	R/W
PIOR6	0	0	PIOR6 5	PIOR6 4	PIOR6 3	PIOR6 2	PIOR6 1	PIOR6 0	F04E6H	00H	R/W
PIOR7	0	0	PIOR7 5	PIOR7 4	PIOR7 3	PIOR7 2	PIOR7 1	PIOR7 0	F04E7H	00H	R/W
PIOR8	0	0	PIOR8 5	PIOR8 4	PIOR8 3	PIOR8 2	PIOR8 1	PIOR8 0	F04E8H	00H	R/W
PIOR9	0	0	PIOR9 5	PIOR9 4	PIOR9 3	PIOR9 2	PIOR9 1	PIOR9 0	F04E9H	00H	R/W
PIOR10	PIOR1 07	PIOR1 06	PIOR1 05	PIOR1 04	PIOR1 03	PIOR1 02	PIOR1 01	PIOR1 00	F04EAH	00H	R/W
PIOR11	PIOR1 17	PIOR1 16	PIOR1 15	PIOR1 14	PIOR1 13	PIOR1 12	PIOR1 11	PIOR1 10	F04EBH	00H	R/W
PIOR12	0	0	0	0	PIOR1 23	PIOR1 22	PIOR1 21	PIOR1 20	F04ECH	00H	R/W
PIOR13	PIOR1 37	PIOR1 36	PIOR1 35	PIOR1 34	PIOR1 33	PIOR1 32	PIOR1 31	PIOR1 30	F04EDH	00H	R/W

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (2/6)

PIOR0 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR06, PIOR05	TI02/TO02	P54 ^{Note 1}	P55 ^{Note 2}	P61	P95	—	—	—	—
PIOR04 to PIOR02	TI01/TO01	P32 ^{Note 3}	P31 ^{Note 4}	P33 ^{Note 3}	P34 ^{Note 2}	P60	P92	P67 ^{Note 3}	—
PIOR01, PIOR00	TI00	P52 ^{Note 4}	P20	P40	P90	—	—	—	—
	TO00	P52 ^{Note 4}	P21	P40	P90	—	—	—	—

PIOR1 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR17 to PIOR15	TI05/TO05	P42 ^{Note 4}	P01 ^{Note 4}	P43 ^{Note 5}	P66 ^{Note 3}	P97	P65 ^{Note 3}	—	—
PIOR14, PIOR13	TI04/TO04	P14 ^{Note 5}	P44	P63 ^{Note 4}	P126	—	—	—	—
PIOR12 to PIOR10	TI03/TO03/R EMOOUT	P30 ^{Note 4}	P32 ^{Note 3}	P57	P62 ^{Note 4}	P64 ^{Note 3}	P75 ^{Note 1}	P127	—

PIOR2 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR23, PIOR22	TI07/TO07	P15 ^{Note 4}	P41 ^{Note 2}	P53 ^{Note 5}	P77	—	—	—	—
PIOR21, PIOR20	TI06/TO06	P56 ^{Note 2}	P86	P125	P94	—	—	—	—

PIOR3 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR36, PIOR35	TKBO21 ^{Note 6}	P72 ^{Note 2}	P77	P02 ^{Note 4}	P27	—	—	—	—
	TKBO20 ^{Note 6}	P70 ^{Note 4}	P70 ^{Note 4}	P26	P26	—	—	—	—
PIOR34, PIOR33	TKBO11 ^{Note 6}	P74 ^{Note 5}	P75 ^{Note 1}	P11	P01 ^{Note 4}	—	—	—	—
	TKBO10 ^{Note 6}	P73 ^{Note 2}	P73 ^{Note 2}	P10	P10	—	—	—	—
PIOR32 to PIOR30	TKBO01	P77	P75 ^{Note 1}	P74 ^{Note 5}	P17	P16	P00	—	—
	TKBO00	P76	P76	P76	P12	P12	P12	—	—

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (3/6)

PIOR4 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR47	EXSDI1	P17	P27	—	—	—	—	—	—
	EXSDI0	P00	P26	—	—	—	—	—	—
PIOR46	EXSDO1	P44	P76	—	—	—	—	—	—
	EXSDO0	P45Note 1	P77	—	—	—	—	—	—
PIOR41	TRJO1Note 6	P130 Note 2	P12	—	—	—	—	—	—
	TRJO1Note 6	P23Note 2	P11	—	—	—	—	—	—
PIOR40	TRJO0	P42Note 4	P00	—	—	—	—	—	—
	TRJO0	P03Note 4	P17	—	—	—	—	—	—

PIOR5 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR57 to PIOR55	RTC1HZ	P31Note 4	P30Note 4	P32Note 3	P33Note 3	P16	P76	—	—
PIOR54, PIOR53	VCOUT1	P04Note 4	P12	P146 Note 3	P125	—	—	—	—
	VCOUT0	P03Note 4	P16	P35Note 2	P85	—	—	—	—
PIOR52 to PIOR50	PCLBUZ1	P01Note 4	P12	P20	P42Note 4	P86	P125	P52Note 4	P06
	PCLBUZ0	P02	P04Note 4	P07	P21	P53Note 5	P54Note 1	P82Note 3	P96

PIOR6 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR65 to PIOR63	INTP1	P52Note 4	P66Note 3	P77	P83Note 3	P90	P95	P47Note 2	P11
PIOR62 to PIOR60	INTP0	P137	P10	P34Note 2	P54Note 1	P65Note 3	P84Note 3	P75Note 1	—

PIOR7 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR75 to PIOR73	INTP3	P31Note 4	P30Note 4	P32Note 3	P60	P86	P145 Note 3	P93	—
PIOR72 to PIOR70	INTP2	P53Note 5	P67Note 3	P76	P91	P94	P147 Note 3	P80Note 3	—

PIOR8 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR85 to PIOR83	INTP5	P55Note 2	P01Note 4	P26	P62Note 4	P64Note 3	P96	P143 Note 3	—
PIOR82 to PIOR80	INTP4	P33Note 3	P27	P35Note 2	P61	P85	P144 Note 3	P126	P43Note 5

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (4/6)

PIOR9 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR95 to PIOR93	INTP7	P02Note 4	P20	P22	P42Note 4	P44	P125	P142 Note 3	P146 Note 3
PIOR92 to PIOR90	INTP6	P57	P04Note 4	P21	P41Note 2	P63Note 4	P141 Note 3	P51Note 2	—

PIOR10 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR107, PIOR106	SCK11	P10	P44Note 4	—	—	—	—	—	—
	SCL11	P10	P44Note 5	—	—	—	—	—	—
	SI11	P11	P43Note 4	—	—	—	—	—	—
	SDA11	P11	P43Note 5	—	—	—	—	—	—
	SO11	P12	P42Note 4	—	—	—	—	—	—
PIOR105, PIOR104	SCK10	P05	P66Note 3	P82Note 3	P85	—	—	—	—
	SCL10	P05	P66Note 3	P82Note 3	P85	—	—	—	—
	SI10	P06	P41Note 3	P81Note 3	P86	—	—	—	—
	RxD1	P06	P41Note 2	P81Note 3	P86	—	—	—	—
	SDA10	P06	P41Note 3	P81Note 3	P86	—	—	—	—
	SO10	P07	P42Note 3	P07Note 3	P87	—	—	—	—
	TxD1	P07	P42Note 2	P07Note 3	P87	—	—	—	—
PIOR103, PIOR102	SCK01	P54Note 4	P46Note 2	P93	P97	—	—	—	—
	SCL01	P54Note 5	P46Note 2	P93	P97	—	—	—	—
	SI01	P53Note 4	P47Note 2	P94	P96	—	—	—	—
	SDA01	P53Note 5	P47Note 2	P94	P96	—	—	—	—
	SO01	P52Note 4	P130 Note 2	P95	P95	—	—	—	—
PIOR101, PIOR100	SCK00	P16	P22	P40Note 2	—	—	—	—	—
	SCL00	P16	P22	P40Note 4	—	—	—	—	—
	SI00	P17	P26	P42Note 2	—	—	—	—	—
	RxD0	P17	P26	P42Note 2	—	—	—	—	—
	SDA00	P17	P26	P42Note 4	—	—	—	—	—
	SO00	P00	P27	P41Note 2	—	—	—	—	—
	TxD0	P00	P27	P41Note 2	—	—	—	—	—

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (5/6)

PIOR11 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR117, PIOR116	SCK31	P140 Note 3	P02Note 3	P02Note 4	—	—	—	—	—
	SCL31	P140 Note 3	P02Note 4	P02Note 4	—	—	—	—	—
	SI31	P46Note 3	P03Note 3	P03Note 4	—	—	—	—	—
	SDA31	P46Note 3	P03Note 4	P03Note 4	—	—	—	—	—
	SO31	P47Note 3	P146 Note 3	P01Note 4	—	—	—	—	—
PIOR115, PIOR114	SCK30	P45Note 5	P85Note 2	P33Note 3	—	—	—	—	—
	SCL30	P45Note 1	P85	P33Note 3	—	—	—	—	—
	SI30	P44Note 5	P125 Note 2	P34Note 3	—	—	—	—	—
	RxD3	P44Note 5	P125 Note 2	P34Note 2	—	—	—	—	—
	SDA30	P44Note 1	P125	P34Note 3	—	—	—	—	—
	SO30	P43Note 5	P35Note 2	P35Note 3	—	—	—	—	—
	TxD3	P43Note 5	P35Note 2	P35Note 2	—	—	—	—	—
PIOR113, PIOR112	SCK21	P22	P46Note 2	—	—	—	—	—	—
	SCL21	P22	P46Note 2	—	—	—	—	—	—
	SI21	P26	P47Note 2	—	—	—	—	—	—
	SDA21	P26	P47Note 2	—	—	—	—	—	—
	SO21	P27	P130 Note 2	—	—	—	—	—	—
PIOR111, PIOR110	SCK20	P75Note 1	P80Note 3	P02Note 4	—	—	—	—	—
	SCL20	P75Note 1	P80Note 3	P02Note 4	—	—	—	—	—
	SI20	P76Note 1	P147 Note 3	P03Note 4	—	—	—	—	—
	RxD2	P76	P147 Note 3	P03Note 4	—	—	—	—	—
	SDA20	P76Note 1	P147 Note 3	P03Note 4	—	—	—	—	—
	SO20	P77Note 1	P146 Note 3	P04Note 4	—	—	—	—	—
	TxD2	P77	P146 Note 3	P04Note 4	—	—	—	—	—

Figure 4 - 10 Format of Peripheral I/O Redirection Registers (PIORx) (6/6)

PIOR12 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR123, PIOR122	SCLA1	P62 ^{Note 4}	P65 ^{Note 3}	P97	P84 ^{Note 3}	—	—	—	—
	SDAA1	P63 ^{Note 4}	P64 ^{Note 3}	P96	P83 ^{Note 3}	—	—	—	—
PIOR121, PIOR120	SCLA0	P60	P42 ^{Note 2}	P144 ^{Note 3}	—	—	—	—	—
	SDAA0	P61	P41 ^{Note 2}	P145 ^{Note 3}	—	—	—	—	—

PIOR13 bit	Function	Set value							
		0	1	2	3	4	5	6	7
PIOR137, PIOR136	RxDA3 ^{Note 6}	P51 ^{Note 2}	P41 ^{Note 3}	P141 ^{Note 3}	—	—	—	—	—
	TxDA3 ^{Note 6}	P50 ^{Note 2}	P66 ^{Note 3}	P147 ^{Note 3}	—	—	—	—	—
PIOR135, PIOR134	RxDA2 ^{Note 6}	P44 ^{Note 1}	P146 ^{Note 3}	P142 ^{Note 3}	—	—	—	—	—
	TxDA2 ^{Note 6}	P45 ^{Note 1}	P03 ^{Note 3}	P143 ^{Note 3}	—	—	—	—	—
PIOR133, PIOR132	RxDA1	P27	P43 ^{Note 4}	P144 ^{Note 3}	—	—	—	—	—
	TxDA1	P26	P42 ^{Note 4}	P145 ^{Note 3}	—	—	—	—	—
PIOR131, PIOR130	RxDA0	P11	P83 ^{Note 3}	P47 ^{Note 2}	—	—	—	—	—
	TxDA0	P10	P84 ^{Note 3}	P130 ^{Note 2}	—	—	—	—	—

Note 1. Not applicable in 44-pin products

Note 2. Not applicable in products with 64 or fewer pins

Note 3. Not applicable in products with 80 or fewer pins

Note 4. Not applicable in products with 52 or fewer pins

Note 5. Not applicable in products with 48 or fewer pins

Note 6. Not applicable in 44- to 64-pin products with 64 to 128 Kbytes of flash memory

Remark —: These functions are not available for use.

4.3.11 Output current control enable register (CCDE)

The CCDE register is used to specify the use of P60 to P63 and P74 to P77 pins as controlled current drive port pins in 1-bit units. The setting of the corresponding output current select register (CCSx) controls a pin for which this function is selected to be at the low level and produce the selected output current (low-level output current) or to be in the high-impedance state. The CCDE 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 4 - 11 Format of Output Current Control Enable Register (CCDE) (1/2)

Address: F02A8H
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CCDE	CCDE07	CCDE06	CCDE05	CCDE04	CCDE03	CCDE02	CCDE01	CCDE00
CCDE07	Selection of whether the CCD07 (P63) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							
CCDE06	Selection of whether the CCD06 (P62) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							
CCDE05	Selection of whether the CCD05 (P61) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							
CCDE04	Selection of whether the CCD04 (P60) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							
CCDE03	Selection of whether the CCD03 (P74) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							
CCDE02	Selection of whether the CCD02 (P75) pin is to be used for digital I/O or output current control function							
0	Digital I/O (alternate function other than current control function)							
1	Current control function							

Figure 4 - 11 Format of Output Current Control Enable Register (CCDE) (2/2)

CCDE01	Selection of whether the CCD01 (P76) pin is to be used for digital I/O or output current control function
0	Digital I/O (alternate function other than current control function)
1	Current control function

CCDE00	Selection of whether the CCD00 (P77) pin is to be used for digital I/O or output current control function
0	Digital I/O (alternate function other than current control function)
1	Current control function

Caution 1. When a port pin is to be used with output current control, make the setting for the output current control function and then set the corresponding bit in the PMxx register for output mode.

Caution 2. The state of a pin takes 10 μ s to become stable after 1 having been written to the corresponding bit of the CCDE register.

4.3.12 Output current select registers (CCSx)

The CCSx registers are used to set the current control for the port pins selected for output current control in the output current control enable register (CCDE) or to place them in the high-impedance state. For current control, the pins are at the low level and produce output currents controlled to be 2 mA, 5 mA, 10 mA, or 15 mA. The CCSx registers can be set by an 8-bit memory manipulation instruction. The value of each CCSx register following a reset is 00H.

Figure 4 - 12 Format of Output Current Select Registers (CCSx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CCS0	0	0	0	0	0	CCS02	CCS01	CCS00	F02A0H	00H	R/W
CCS4	0	0	0	0	0	CCS42	CCS41	CCS40	F02A4H	00H	R/W
CCS5	0	0	0	0	0	CCS52	CCS51	CCS50	F02A5H	00H	R/W
CCS6	0	0	0	0	0	CCS62	CCS61	CCS60	F02A6H	00H	R/W
CCS7	0	0	0	0	0	CCS72	CCS71	CCS70	F02A7H	00H	R/W

CCSx2	CCSx1	CCSx0	Setting for the Low-level Output Current				
			x = 0 CCD00 to CCD03	x = 4 CCD04	x = 5 CCD05	x = 6 CCD06	x = 7 CCD07
0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
0	0	1	2 mA	2 mA	2 mA	2 mA	2 mA
0	1	0	5 mA	5 mA	5 mA	5 mA	5 mA
0	1	1	10 mA	10 mA	10 mA	10 mA	10 mA
1	0	0	Setting prohibited	15 mA	15 mA	15 mA	15 mA
Others			Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

Remark x = 0, 4 to 7

4.3.13 40-mA port output control register (PTDC)

The PTDC register is used to set the allowable output current to 40 mA when the P07, P10, P27, P16, P05, or P00 pin is an output at the low level. The pins among those stated above which can be set for 40-mA output depend on the product. **Table 4 - 3** is a list by product of port pins which can be set for 40-mA output.

Table 4 - 3 Products Having 40-mA Output Port Pins and the Pins Supporting 40-mA Output

	Product Names	Port Pins Supporting 40-mA Output
100 pins	R7F100LPL, R7F100LPJ, R7F100LPG	P07, P10, P27
80 pins	R7F100LML, R7F100LMJ, R7F100LMG	P07, P10, P27
64 pins	R7F100LLL, R7F100LLJ (products with 512 or 256 Kbytes of flash memory)	P07, P10, P27
	R7F100LLG, R7F100LLE (products with 128 or 64 Kbytes of flash memory)	P16, P05, P00
52 pins	R7F100LJL, R7F100LJJ, R7F100LJG, R7F100LJE	P16, P05, P00
48 pins	R7F100LGL, R7F100LGJ, R7F100LGG, R7F100LGE	P16, P05, P00
44 pins	R7F100LFL, R7F100LFJ, R7F100LFG, R7F100LFE	P16, P05, P00

The PTDC 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 4 - 13 Format of 40-mA Port Output Control Register (PTDC)

Address: F02A9H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PTDC	0	0	PTDC5	PTDC4	PTDC3	PTDC2	PTDC1	PTDC0
PTDC5	Allowable output current when a low level is output from P07 pin ^{Note 1}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							
PTDC4	Allowable output current when a low level is output from P10 pin ^{Note 2}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							
PTDC3	Allowable output current when a low level is output from P27 pin ^{Note 3}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							
PTDC2	Allowable output current when a low level is output from P16 pin ^{Note 4}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							
PTDC1	Allowable output current when a low level is output from P05 pin ^{Note 5}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							
PTDC0	Allowable output current when a low level is output from P00 pin ^{Note 6}							
0	IOL1 = 20.0 mA							
1	IOL1 = 40.0 mA							

Note 1. This is usable in the 64-pin products with 256 to 512 Kbytes of code flash memory and in the 80- and 100-pin products. The P81 pin is not available on 100-pin products. Be sure to short-circuit the P07 and P81 pins outside the device.

Note 2. This is usable in the 64-pin products with 256 to 512 Kbytes of code flash memory and in the 80- and 100-pin products. The P145 pin is not available on 100-pin products. Be sure to short-circuit the P10 and P145 pins outside the device.

Note 3. This is usable in the 64-pin products with 256 to 512 Kbytes of code flash memory and in the 80- and 100-pin products. The P144 pin is not available on 100-pin products. Be sure to short-circuit the P27 and P144 pins outside the device.

Note 4. This is usable in the 44- to 52-pin products and in the 64-pin products with 64 to 128 Kbytes of code flash memory. The P15 pin is not available on 64-pin products. Be sure to short-circuit the P16 and P15 pins outside the device.

Note 5. This is usable in the 44- to 52-pin products and in the 64-pin products with 64 to 128 Kbytes of code flash memory. The P04 pin is not available on 64-pin products. Be sure to short-circuit the P05 and P04 pins outside the device.

Note 6. This is usable in the 44- to 52-pin products and in the 64-pin products with 64 to 128 Kbytes of code flash memory. The P01 pin is not available on 64-pin products. Be sure to short-circuit the P00 and P01 pins outside the device.

4.3.14 Port function output enable registers (PFOEx)

The PFOEx registers are used to enable timer output, serial clock output, serial data output, and clock output for pins on which these functions are multiplexed. Disable the output when timer output, serial data output, and serial clock output are only to be input to the ELCL. The PFOEx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PFOEx register following a reset is FFH.

Figure 4 - 14 Format of Port Function Output Enable Register 0 (PFOE0)

Address: F02AAH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFOE0	1	PFOE06	PFOE05	PFOE04	PFOE03	PFOE02	PFOE01	PFOE00

PFOE0n	TO0n pin output enable (n = 0 to 6)
0	Timer output to the TO0n pin is disabled. The timer output is only to be used for input to the ELCL and not for the TO0n pin. The TO0n pin is available for use for a multiplexed function or the port pin.
1	Timer output to the TO0n pin is enabled. If channel n of TAU0 is not in use, the TO0n pin is available for use for the port pin or a multiplexed function.

Figure 4 - 15 Format of Port Function Output Enable Register 1 (PFOE1) (1/2)

Address: F02ABH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFOE1	1	1	PFOE15	PFOE14	PFOE13	PFOE12	PFOE11	PFOE10

PFOE15	CLKA0 pin output enable
0	Serial clock output to the CLKA0 pin is disabled. The serial clock output is only to be used for input to the ELCL and not for the CLKA0 pin. The CLKA0 pin is available for use for the port pin or a multiplexed function.
1	Serial clock output to the CLKA0 pin is enabled. If the serial clock output is not in use, the CLKA0 pin is available for use for the port pin or a multiplexed function.

PFOE14	TxDA0 pin output enable
0	Serial data output to the TxDA0 pin is disabled. The serial data output is only to be used for input to the ELCL and not for the TxDA0 pin. The TxDA0 pin is available for use for the port pin or a multiplexed function.
1	Serial data output to the TxDA0 pin is enabled. If UARTA0 is not in use, the TxDA0 pin is available for use for the port pin or a multiplexed function.

Figure 4 - 15 Format of Port Function Output Enable Register 1 (PFOE1) (2/2)

PFOE13	SCK01 and SCL01 pins output enable
0	Serial clock output to the SCK01 and SCL01 pins is disabled. The serial clock output is only to be used for input to the ELCL and not for the SCK01 or SCL01 pin. 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 serial clock output is only to be used for input to the ELCL and not for the SCK00 or SCL00 pin. 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 serial data output is only to be used for input to the ELCL and not for the SO01 pin. 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 serial data output is only to be used for input to the ELCL and not for the SO00 or TxD0 pin. 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.

4.3.15 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 **32.3.11.1 Port mode select register (PMS)**.

4.3.16 LCD port function registers (PFSEGx)

The PFSEGx registers set whether to use the P00 to P07, P10 to P17, P22 to P27, P30 to P35, P46, P47, P50 to P57, P70 to P77, P90 to P97, P130, and P141 to P145 pins as port pins (other than as segment or common output pins) or as segment or common output pins. The PFSEGx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The values of the PFSEGx registers following a reset are as listed below.

Figure 4 - 16 Format of LCD Port Function Registers (PFSEGx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PFSEG0	0	0	0	0	PFCO M3	PFCO M2	PFCO M1	PFCO M0	F0290H	0FH	R/W
PFSEG1	PFSEG 07	PFSEG 06	PFSEG 05	PFSEG 04	PFSEG 03	PFSEG 02	PFSEG 01	PFSEG 00	F0291H	FFH	R/W
PFSEG2	PFSEG 15	PFSEG 14	PFSEG 13	PFSEG 12	PFSEG 11	PFSEG 10	PFSEG 09	PFSEG 08	F0292H	FFH	R/W
PFSEG3	PFSEG 23	PFSEG 22	PFSEG 21	PFSEG 20	PFSEG 19	PFSEG 18	PFSEG 17	PFSEG 16	F0293H	FFH	R/W
PFSEG4	PFSEG 31	PFSEG 30	PFSEG 29	PFSEG 28	PFSEG 27	PFSEG 26	PFSEG 25	PFSEG 24	F0294H	EFH	R/W
PFSEG5	PFSEG 39	PFSEG 38	PFSEG 37	PFSEG 36	PFSEG 35	PFSEG 34	PFSEG 33	PFSEG 32	F0295H	FFH	R/W
PFSEG6	PFSEG 47	PFSEG 46	PFSEG 45	PFSEG 44	PFSEG 43	PFSEG 42	PFSEG 41	PFSEG 40	F0296H	FFH	R/W
PFSEG7	PFSEG 55	PFSEG 54	PFSEG 53	PFSEG 52	PFSEG 51	PFSEG 50	PFSEG 49	PFSEG 48	F0297H	FFH	R/W

PFCOMy (y = 0 to 3)	Selection of whether to use the Pmn pins as port pins (other than as common output pins) or as common output pins (mn = 90 to 93)
0	Uses the pins as port pins (other than as common output pins).
1	Uses the pins as common output pins.

PFSEGz (z = 00 to 55)	Selection of whether to use the Pmn pins as port pins (other than as segment or common output pins) or as segment or common output pins (mn = 00 to 07, 10 to 17, 22 to 27, 30 to 35, 46, 47, 50 to 57, 70 to 77, 94 to 97, 130, or 141 to 145)
0	Uses the pins as port pins (other than as segment or common output pins).
1	Uses the pins as segment or common output pins.

Remark For the common/segment output pins provided by each product and their corresponding PFSEG registers, see **Table 22 - 10 Common and Segment Output Pins in Each Product and the Corresponding PFSEGx Registers (PFCOMy and PFSEGz Bits)** in **22.3.8 LCD port function registers (PFSEGx)**.

4.3.17 LCD input switch control register (ISCLCD)

The ISCLCD register sets whether to use the P125 to P127 and P85 to P87 pins as port pins (other than as LCD function pins) or as LCD function pins (VL1, VL2, VL3, VL4, CAPL, or CAPH). The ISCLCD register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

<R> Figure 4 - 17 Format of LCD Input Switch Control Register (ISCLCD)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
ISCLCD	0	0	0	ISCVL4	ISCVL3	ISCVL2	ISCVL1	ISCCAP	F0298H	00H	R/W

ISCVL4	Control of the Schmitt trigger buffer of the VL4/P85 pin
0	The LCD function enabled (i.e. the pin is used as the VL4 pin).
1	The LCD function disabled (i.e. digital input enabled).

ISCVL3	Control of the Schmitt trigger buffer of the VL3/P125 pin
0	The LCD function enabled (i.e. the pin is used as the VL3 pin).
1	The LCD function disabled (i.e. digital input enabled).

ISCVL2	Control of the Schmitt trigger buffer of the VL2/P86 pin
0	The LCD function enabled (i.e. the pin is used as the VL2 pin).
1	The LCD function disabled (i.e. digital input enabled).

ISCVL1	Control of the Schmitt trigger buffer of the VL1/P87 pin
0	The LCD function enabled (i.e. the pin is used as the VL1 pin).
1	The LCD function disabled (i.e. digital input enabled).

ISCCAP	Control of the Schmitt trigger buffer of the CAPL/P126 and CAPH/P127 pins
0	The LCD function enabled (i.e. the pins are used as the CAPL and CAPH pins).
1	The LCD function disabled (i.e. digital input enabled).

Caution If ISCVL1 bit = 0, ISCVL2 bit = 0, ISCVL3 bit = 0, ISCVL4 bit = 0, and ISCCAP bit = 0, set the corresponding port control registers as follows:

- PU127 bit of PU12 register = 0, P127 bit of P12 register = 0
- PU126 bit of PU12 register = 0, P126 bit of P12 register = 0
- PU125 bit of PU12 register = 0, P125 bit of P12 register = 0
- PU85 bit of PU8 register = 0, P85 bit of P8 register = 0
- PU86 bit of PU8 register = 0, P86 bit of P8 register = 0
- PU87 bit of PU8 register = 0, P87 bit of P8 register = 0

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 to 9, 12, and 14 (PIM0 to PIM9, PIM12, and PIM14) 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 to 9 and 12 to 14 (POM0 to POM9 and POM12 to POM14) in 1-bit units to switch from normal output (CMOS) to N-ch open-drain output (VDD withstand voltage) mode. Setting port digital input disable registers 0 to 9 and 12 to 14 (PDIDIS0 to PDIDIS9 and PDIDIS12 to PDIDIS14) bit-by-bit can prevent the flow of through currents to the corresponding input buffers. 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 UART3, UARTA0, UARTA1, CSI00, CSI01, CSI10, CSI20, CSI30, and CSI31 for use with the TTL input buffers

Use the following port pin or pins for each interface.

- P17 (P26 and P42) for UART0
- P06 (P41, P81, and P86) for UART1
- P76 (P147 and P03) for UART2
- P44 (P125 and P34) for UART3
- P11 (P83 and P47) for UARTA0
- P27 (P43 and P144) for UARTA1
- P44 (P146 and P142) for UARTA2
- P51 (P41 and P141) for UARTA3
- P17 and P16 (P26, P42, P22, and P40) for CSI00
- P53 and P54 (P47, P94, P96, P46, P93, and P97) for CSI01
- P06 and P05 (P41, P81, P86, P66, P82, and P85) for CSI10
- P76 and P75 (P147, P03, P80, and P02) for CSI20
- P44 and P45 (P125, P34, P85, and P33) for CSI30
- P46 and P140 (P03 and P02) for CSI31

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

- <1> Using an external resistor, pull up externally the input pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
- <2> Set the corresponding bit of the PIM0 to PIM9, PIM12, and PIM14 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 UART3, UARTA0, UARTA1, CSI00, CSI01, CSI10, CSI20, CSI30, and CSI31 for use with the N-ch open-drain output mode

- P00 (P27 and P41) for UART0
- P07 (P42, P07, and P87) for UART1
- P77 (P146 and P04) for UART2
- P43 (P35) for UART3
- P10 (P84 and P130) for UARTA0
- P26 (P42 and P145) for UARTA1
- P45 (P03 and P143) for UARTA2
- P50 (P66 and P147) for UARTA3
- P00 and P16 (P27, P41, P22, and P40) for CSI00
- P52 and P54 (P130, P95, P46, P93, and P97) for CSI01
- P07 and P05 (P42, P07, P87, P66, P82, and P85) for CSI10
- P77 and P75 (P146, P04, P80, and P02) for CSI20
- P43 and P45 (P35, P85, and P33) for CSI30
- P47 and P140 (P146, P01, and P02) for CSI31

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

- <1> Using an external resistor, pull up externally the output pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
- <2> After reset release, the ports are in the input mode (Hi-Z).
- <3> Set the corresponding bit of the PDIDIS0 to PDIDIS9 and PDIDIS14 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 to POM9, POM13, and POM14 registers to 1 to set the N-ch open-drain output (V_{DD} withstand voltage) 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, PM3 to PM5, PM8, and PM14 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, IIC10, IIC20, IIC30, and IIC31 for use in connection with a device operating at a different voltage (1.8 V, 2.5 V, or 3 V)
- P17 and P16 (P26, P42, P22, and P40) for IIC00
 - P53 and P54 (P47, P94, P96, P46, P93, and P97) for IIC01
 - P06 and P05 (P41, P81, P86, P66, P82, and P85) for IIC10
 - P76 and P75 (P147, P03, P80, and P02) for IIC20
 - P44 and P45 (P125, P34, P85, and P33) for IIC30
 - P46 and P140 (P03 and P02) for IIC31

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

- <1> Using an external resistor, pull up externally the input pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
- <2> After reset release, the ports are in the input mode (Hi-Z).
- <3> Set the output latch of the corresponding port to 1.
- <4> Set the corresponding bit of the POM0 to POM9, POM12, and POM14 registers to 1 to set the N-ch open-drain output (VDD withstand voltage) mode.
- <5> Set the corresponding bit of the PIM0 to PIM9, PIM12, and PIM14 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 to PM9, PM12, and PM14 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.

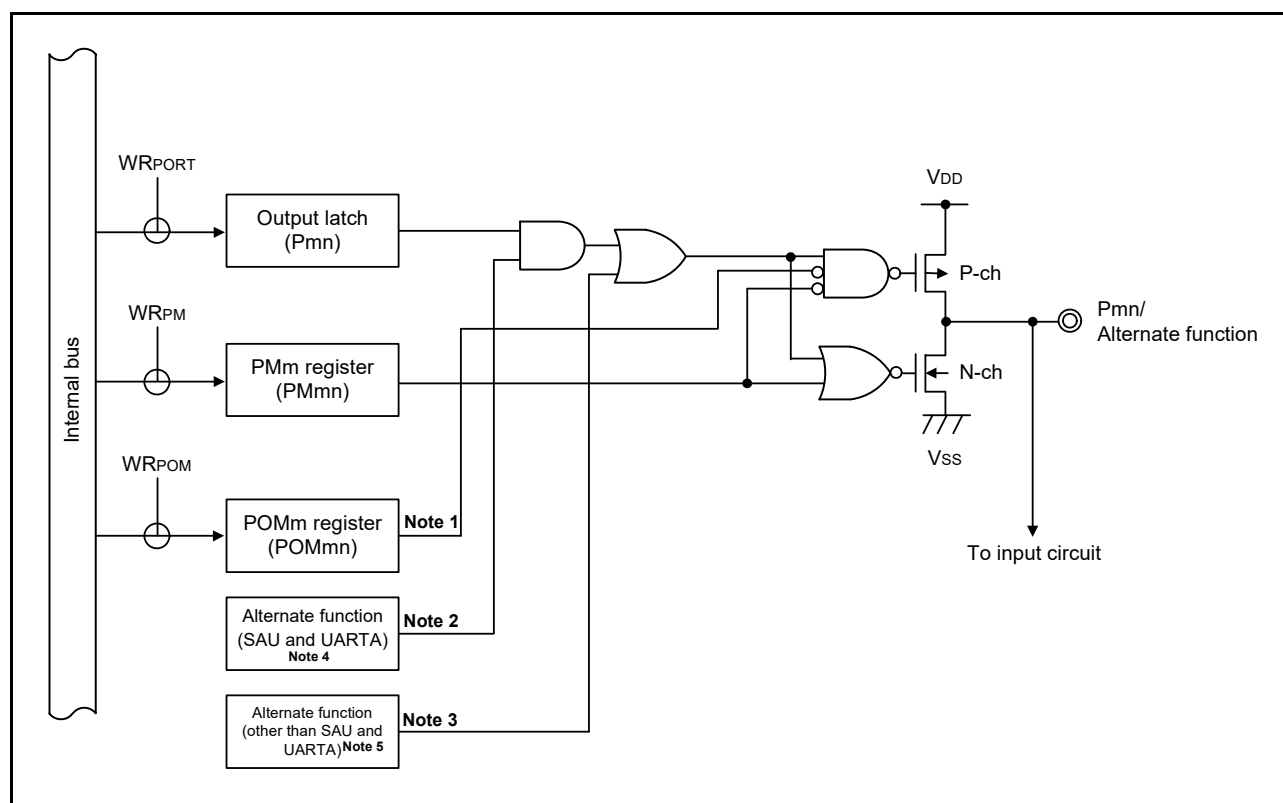
4.5 Register Settings When Using Alternate Function

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 can be used for both analog input and segment output, use the corresponding LCD port function register (PFSEGx) and port mode control A register (PMCAxx) to specify which the pin is to be used for, analog input, segment output, 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. For a pin that is also assigned to the logic and event link controller output, use the corresponding port mode control E register (PMCExx) to specify whether to use the pin for the logic and event link controller function.

Figure 4 - 18 shows the basic configuration of an output circuit for pins used for digital input/output. The outputs from multiplexed SAU and UARTA functions (excluding the clock output from the UARTA) 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. An idea of basic settings for this kind of case is shown in **Table 4 - 4**.

Figure 4 - 18 Basic Configuration of Output Circuit for a Pin



Note 1. When there is no POMm 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).

Note 4. This excludes the clock outputs from the UARTA.

Note 5. This includes the clock outputs from the UARTA.

Remark m: Port number (0 to 9 or 12 to 14); n: Bit number (0 to 7)

Table 4 - 4 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 ^{Note 1}	Output Functions Other than Those of the SAU and UARTA ^{Note 2}
Port-pins output function	—	Output is high level (1)	Output is low level (0)
Output functions of the SAU and UARTA ^{Note 1}	High level (1)	Output is high level (1)	Output is low level (0)
Output functions other than those of the SAU and UARTA ^{Note 2}	Low level (0)	Don't care	Output is low level (0) ^{Note 3}

Note 1. This excludes the clock outputs from the UARTA.

Note 2. This includes the clock outputs from the UARTA.

Note 3. 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 a pin is not to be 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 corresponding peripheral I/O redirection register (PIORx). This allows usage of the port function or another alternate function that is 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-level output). 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 CKOm bits in serial output register m (SOM) to 1 (high-level output). 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-level output). These are the same settings as the initial state. Regardless of the selection for timer output operation, a TOMn output can also be stopped by setting the PFOE0x bit corresponding to the TOMn pin to 0.
4. SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) 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. TxDAn = 1, CLKAAn = 0 (settings when the UARTA is not used)
When the UARTA is not used, set the UARTAENn, TXEAn, and RXEAn bits in operation mode setting register n0 (ASIMAn0) to 0 (operation disabled). In addition, set the PFOE1x bit corresponding to the TxDAn pin to 1. These are the same settings as the initial state. Output of the CLKA0 signal from the CLKA0 pin can also be stopped by setting the PFOE1x bit corresponding to the CLKA0 pin to 0 regardless of the selection for UARTA operation.
7. REMOOUT = 0 (settings when remote-control output is not used)
When the REMOOUT output of TAU is not used, set the bit in timer output enable register 0 (TOE0) of channels 2 to 5 to 0 (output disabled), and set the bit in timer output register 0 (TO0) to 0 (low-level output). These are the same settings as the initial state.

Remark p: CSI number (00, 01, 10, 11, 20, 21, 30, or 31); q: UART number (0 to 3);
 r: IIC number (00, 01, 10, 11, 20, 21, 30, or 31); n: UARTA number (0 or 1)

8. VCOUTn = 0 (setting when VCOUTn is not used)

When VCOUTn of comparators is not used, set bits 5 and 1 in the comparator output control register (COMPOCR) to 0 (VCOUTn pin of comparator n output disabled). This is the same setting as the initial state.

9. TKBOxx = 0 (settings when 16-bit timer KB4 is not used)

When 16-bit timer KB4 is not used, set bit 7 in 16-bit timer KB operation control register 01 (TKBCTL01) to 0 (timer operation stopped), set bits 1 and 0 in 16-bit timer KB output control register 01 (TKBIOC01) to 0 (timer output disabled), set bits 1 and 0 in 16-bit timer KB output control register 00 (TKBIOC00) to 0 (default level is low level), and set bits 1 and 0 in forced output stop function control register 02 (TKBPACTL02) to 0 (forced output stop function operation disabled). These are the same settings as the initial state.

10. TRJION = 0/TRJOn = 0 (setting when timer RJn output is not used)

When the pulse output function of timer RJ is not used for the TRJOn pin, set bit 2 (TOENAn) of timer RJ I/O control registers n (TRJIOCn) to 0 (TRJOn output disabled). This is the same setting as the initial value. When the TRJION pin of timer RJ is not used as the output function, make sure that bits 2 to 0 (TMODn2 to TMODn0) of timer RJ mode registers n (TRJMRn) are set to a value other than 001B (pulse output mode). The initial value is 000B, which sets timer mode.

11. EXSDO0 = 0/EXSDO1 = 0 (setting when the external signal sampler clock output is not used)

When the external signal sampler clock output (EXSDOn) is not used, set bit n (EXSDOE0n) of external signal sampler control register 0 (EXSDM0) to 0 (EXSDOn output disabled). This is the same setting as the initial value.

4.5.3 Register settings and port pin state

The correspondence between register settings and port pin states is shown in **Table 4 - 5**.

Table 4 - 5 Correspondence between Register Settings and Port Pin State

PMCAxx	PFSEGx	PMCTxx	PMCExx	PMxx	Pxx	PUxx	CCDE0x	CCSx	Pin State
1	1	x	x	x	x	x	x	x	Analog input/output
0	1	0	x	0	x	x	x	x	LCD common/segment output
0	1	0	x	1	x	x	x	x	Digital input disabled
0	0	1	x	x	x	x	x	x	Capacitance measurement
0	0	0	1	0	x	x	x	x	ELCL output
0	0	0	x	1	x	1	x	x	Pulled up
0	0	0	x	1	x	0	x	x	Hi-Z
0	0	0	0	0	1	x	x	x	High-level port output
0	0	0	0	0	0	x	1	000	Hi-Z
0	0	0	0	0	0	x	1	001 010 011 100	Controlled current drive port
0	0	0	0	0	0	x	0	x	Low-level port output

Remark x: Don't care

4.5.4 Examples of register settings for port and alternate functions

Register setting examples for used port and alternate functions are shown in **Table 4 - 6**. The registers used to control the port functions should be set as shown in **Table 4 - 6**. See the following remark for legends.

Remark —: Not supported

x: Don't care

PIORx: Peripheral I/O redirection registers

POMxx: Port output mode registers

PMCAxx: Port mode control A registers

PMCTxx: Port mode control T registers

PMCExx: Port mode control E registers

CCDE: Output current control enable register

CCSx: Output current select registers

PMxx: Port mode registers

Pxx: Port output latch

PFSEGx: LCD port function registers

ISCLCD: LCD input switch control register

Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx).

For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

Caution In examples of register settings, the port digital input disable registers (PDIDISxx) are set for input to the input buffer being enabled.

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (1/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P00	P00	Input	—	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	0	0/1	0	SO00/TxD0 = 1	(TKBO01) = 0 (TRJO0) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	—	0	0	0/1	0	SO00/TxD0 = 1	(TKBO01) = 0 (TRJO0) = 0	✓	✓	✓	✓	✓	✓
	SO00	Output	PIOR101 = 0 PIOR100 = 0	0/1	—	0	0	1	0	—	(TKBO01) = 0 (TRJO0) = 0	✓	✓	✓	✓	✓	✓
	TxD0	Output	PIOR101 = 0 PIOR100 = 0	0/1	—	0	0	1	0	—	(TKBO01) = 0 (TRJO0) = 0	✓	✓	✓	✓	✓	✓
	EI00	Input	—	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EO00	Output	—	0/1	—	1	0	x	0	—	—	✓	✓	✓	✓	✓	✓
	EXSDI0	Input	PIOR47 = 0	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TKBO01)	Output	PIOR32 = 1 PIOR31 = 0 PIOR30 = 1	0	—	0	0	0	0	SO00/TxD0 = 1	(TRJO0) = 0	✓	✓	✓	✓	✓	✓
	(TRJO0)	Output	PIOR40 = 1	0	—	0	0	0	0	SO00/TxD0 = 1	(TKBO01) = 0	✓	✓	✓	✓	✓	✓
	SEG43	Output	—	0	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P01	P01	Input	—	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	0	—	—	0	0/1	0	(SO31) = 1	PCLBUZ1 = 0 (TO05) = 0 (TKBO11) = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	—	—	0	0/1	0	(SO31) = 1	PCLBUZ1 = 0 (TO05) = 0 (TKBO11) = 0	x	x	x	✓	✓	✓
	PCLBUZ1	Output	PIOR52 = 0 PIOR51 = 0 PIOR50 = 0	0	—	—	0	0	0	(SO31) = 1	(TO05) = 0 (TKBO11) = 0	x	x	x	✓	✓	✓
	(TI05)	Input	PIOR17 = 0 PIOR16 = 0 PIOR15 = 1	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(TO05)	Output	PIOR17 = 0 PIOR16 = 0 PIOR15 = 1	0	—	—	0	0	0	(SO31) = 1	PCLBUZ1 = 0 (TKBO11) = 0	x	x	x	✓	✓	✓
	(TKBO11) ^{Note 2}	Output	PIOR34 = 1 PIOR33 = 1	0	—	—	0	0	0	(SO31) = 1	PCLBUZ1 = 0 (TO05) = 0	x	x	x	✓	✓	✓
	(INTP5)	Input	PIOR85 = 0 PIOR84 = 0 PIOR83 = 1	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SO31)	Output	PIOR117 = 1 PIOR116 = 0	0/1	—	—	0	1	0	—	PCLBUZ1 = 0 (TO05) = 0 (TKBO11) = 0	x	x	x	✓	✓	✓
	SEG44	Output	—	0	—	—	0	0	1	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (2/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEG2) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P02	P02	Input	—	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	0	—	—	0	0/1	0	(SCK20/SCL20) = 1 (SCK31/SCL31) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	—	—	0	0/1	0	(SCK20/SCL20) = 1 (SCK31/SCL31) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
	INTP7	Input	PIOR95 = 0 PIOR94 = 0 PIOR93 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	PCLBUZ0	Output	PIOR52 = 0 PIOR51 = 0 PIOR50 = 0	0	—	—	0	0	0	(SCK20/SCL20) = 1 (SCK31/SCL31) = 1	(TKBO21) = 0	x	x	x	✓	✓	✓
	(TKBO21) ^{Note 2}	Output	PIOR36 = 1 PIOR35 = 0	0	—	—	0	0	0	(SCK20/SCL20) = 1 (SCK31/SCL31) = 1	PCLBUZ0 = 0	x	x	x	✓	✓	✓
	(SCK20)	Input	PIOR111 = 1 PIOR110 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SCK20)	Output	PIOR111 = 1 PIOR110 = 0	0/1	—	—	0	1	0	(SCK31/SCL31) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
	(SCL20)	Output	PIOR111 = 1 PIOR110 = 0	0/1	—	—	0	1	0	(SCK31/SCL31) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
	(SCK31)	Input	PIOR117 = 0 ^{Note 3} PIOR116 = 1 or PIOR117 = 1 PIOR116 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SCK31)	Output	PIOR117 = 0 ^{Note 3} PIOR116 = 1 or PIOR117 = 1 PIOR116 = 0	0/1	—	—	0	1	0	(SCK20/SCL20) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
	(SCL31)	Output	PIOR117 = 0 PIOR116 = 1 or PIOR117 = 1 PIOR116 = 0	0/1	—	—	0	1	0	(SCK20/SCL20) = 1	PCLBUZ0 = 0 (TKBO21) = 0	x	x	x	✓	✓	✓
	SEG45	Output	—	0	—	—	0	0	1	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (3/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP)Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P03	P03	Input	—	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	0	—	—	0	0/1	0	(TxDA2) = 1 (SDA20) = 1 (SDA31) = 1	VCOUT0 = 0 TRJIO0 = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	—	—	0	0/1	0	(TxDA2) = 1 (SDA20) = 1 (SDA31) = 1	VCOUT0 = 0 TRJIO0 = 0	x	x	x	✓	✓	✓
	VCOUT0	Output	PIOR54 = 0 PIOR53 = 0	0	—	—	0	0	0	(TxDA2) = 1 (SDA20) = 1 (SDA31) = 1	TRJIO0 = 0	x	x	x	✓	✓	✓
	TRJIO0	Input	PIOR40 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	TRJIO0	Output	PIOR40 = 0	0	—	—	0	0	0	(TxDA2) = 1 (SDA20) = 1 (SDA31) = 1	VCOUT0 = 0	x	x	x	✓	✓	✓
	(TxDA2) Note 2	Output	PIOR135 = 0 PIOR134 = 1	0/1	—	—	0	1	0	(SDA20) = 1 (SDA31) = 1	VCOUT0 = 0 TRJIO0 = 0	x	x	x	x	x	✓
	(RxD2)	Input	PIOR111 = 1 PIOR110 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SI20)	Input	PIOR111 = 1 PIOR110 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SDA20)	Input/output	PIOR111 = 1 PIOR110 = 0	1	—	—	0	1	0	(TxDA2) = 1 (SDA31) = 1	VCOUT0 = 0 TRJIO0 = 0	x	x	x	✓	✓	✓
	(SI31)	Input	PIOR117 = 0Note 3 PIOR116 = 1 or PIOR117 = 1 PIOR116 = 0	x	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SDA31)	Input/output	PIOR117 = 0 PIOR116 = 1 or PIOR117 = 1 PIOR116 = 0	1	—	—	0	1	0	(TxDA2) = 1 (SDA20) = 1	VCOUT0 = 0 TRJIO0 = 0	x	x	x	✓	✓	✓
	SEG46	Output	—	0	—	—	0	0	1	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (4/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P04	P04	Input	—	x	0	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	0	0	—	0	0/1	0	(SO20/TxD2) = 1	VCOUT1 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	0	—	0	0/1	0	(SO20/TxD2) = 1	VCOUT1 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	VCOUT1	Output	PIOR54 = 0 PIOR53 = 0	0	0	—	0	0	0	(SO20/TxD2) = 1	(PCLBUZ0) = 0	x	x	x	✓	✓	✓
	TS35	Input/output	—	x	1	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(INTP6)	Input	PIOR92 = 0 PIOR91 = 0 PIOR90 = 1	x	0	—	1	x	0	—	—	x	x	x	✓	✓	✓
	(SO20)	Output	PIOR111 = 1 PIOR110 = 0	0/1	0	—	0	1	0	—	VCOUT1 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	(TxD2)	Output	PIOR111 = 1 PIOR110 = 0	0/1	0	—	0	1	0	—	VCOUT1 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 0 PIOR51 = 0 PIOR50 = 1	0	0	—	0	0	0	(SO20/TxD2) = 1	VCOUT1 = 0	x	x	x	✓	✓	✓
P05	SEG47	Output	—	0	0	—	0	0	1	—	—	x	x	x	✓	✓	✓
	P05	Input	—	x	0	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0	0/1	0	SCK10/SCL10 = 1	—	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0	0/1	0	SCK10/SCL10 = 1	—	✓	✓	✓	✓	✓	✓
	SCK10	Input	PIOR105 = 0 PIOR104 = 0	x	0	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SCK10	Output	PIOR105 = 0 PIOR104 = 0	0/1	0	0	0	1	0	—	—	✓	✓	✓	✓	✓	✓
	SCL10	Output	PIOR105 = 0 PIOR104 = 0	0/1	0	0	0	1	0	—	—	✓	✓	✓	✓	✓	✓
	TS34	Input/output	—	x	1	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
P06	EO05	Output	—	0/1	0	1	0	x	0	—	—	✓	✓	✓	✓	✓	✓
	SEG48	Output	—	0	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	P06	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	0	0/1	0	SDA10 = 1	(PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	—	0	0/1	0	SDA10 = 1	(PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	SI10	Input	PIOR105 = 0 PIOR104 = 0	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	RxD1	Input	PIOR105 = 0 PIOR104 = 0	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SDA10	Input/output	PIOR105 = 0 PIOR104 = 0	1	0	—	0	1	0	—	(PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	TS33	Input/output	—	x	1	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EI06	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 1 PIOR51 = 1 PIOR50 = 1	0	0	—	0	0	0	SDA10 = 1	—	✓	✓	✓	✓	✓	✓
SEG49	Output	—	0	0	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (5/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P07	P07	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	0	0/1	0	SO10/TxD1 = 1	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	—	0	0/1	0	SO10/TxD1 = 1	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	SO10	Output	PIOR105 = 0 PIOR104 = 0 or PIOR105 = 1 ^{Note 3} PIOR104 = 0	0/1	0	—	0	1	0	—	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	TxD1	Output	PIOR105 = 0 PIOR104 = 0 or PIOR105 = 1 ^{Note 3} PIOR104 = 0	0/1	0	—	0	1	0	—	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	TS32	Input/output	—	x	1	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EI07	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 0 PIOR51 = 1 PIOR50 = 0	0	0	—	0	0	0	SO10/TxD1 = 1	—	✓	✓	✓	✓	✓	✓
	SEG50	Output	—	0	0	—	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (6/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P10	P10	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	0	0/1	0	TxDA0 = 1 SCK11/SCL11 = 1	(TKBO10) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	—	0	0/1	0	TxDA0 = 1 SCK11/SCL11 = 1	(TKBO10) = 0	✓	✓	✓	✓	✓	✓
	ANI22	Analog input	—	x	1	—	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	TxDA0	Output	PIOR131 = 0 PIOR130 = 0	0/1	0	—	0	1	0	SCK11/SCL11 = 1	(TKBO10) = 0	✓	✓	✓	✓	✓	✓
	SCK11	Input	PIOR107 = 0 PIOR106 = 0	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SCK11	Output	PIOR107 = 0 PIOR106 = 0	0/1	0	—	0	1	0	TxDA0 = 1	(TKBO10) = 0	✓	✓	✓	✓	✓	✓
	SCL11	Output	PIOR107 = 0 PIOR106 = 0	0/1	0	—	0	1	0	TxDA0 = 1	(TKBO10) = 0	✓	✓	✓	✓	✓	✓
	(TKBO10) ^{Note 2}	Output	PIOR34 = 1 PIOR33 = 0 or PIOR34 = 1 PIOR33 = 1	0	0	—	0	0	0	TxDA0 = 1 SCK11/SCL11 = 1	—	✓	✓	✓	✓	✓	✓
P11	P11	Input	—	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	—	0	0/1	0	SDA11 = 1	(TKBO11) = 0 (TRJIO1) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	—	0	0/1	0	SDA11 = 1	(TKBO11) = 0 (TRJIO1) = 0	✓	✓	✓	✓	✓	✓
	ANI23	Analog input	—	x	1	—	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	RxDA0	Input	PIOR131 = 0 PIOR130 = 0	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SI11	Input	PIOR107 = 0 PIOR106 = 0	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SDA11	Input/output	PIOR107 = 0 PIOR106 = 0	1	0	—	0	1	0	—	(TKBO11) = 0 (TRJIO1) = 0	✓	✓	✓	✓	✓	✓
	(TKBO11) ^{Note 2}	Output	PIOR34 = 1 PIOR33 = 0	0	0	—	0	0	0	SDA11 = 1	(TRJIO1) = 0	✓	✓	✓	✓	✓	✓
	(TRJIO1) ^{Note 2}	Input	PIOR41 = 1	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TRJIO1) ^{Note 2}	Output	PIOR41 = 1	0	0	—	0	0	0	SDA11 = 1	(TKBO11) = 0	✓	✓	✓	✓	✓	✓
	(INTP1)	Input	PIOR65 = 1 PIOR64 = 1 PIOR63 = 1	x	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SEG36	Output	—	0	0	—	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (7/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P12	P12	Input	—	—	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	—	0	0/1	0	SO11 = 1	CLKA0 = 0 (VCOUT1) = 0 (TKBO00) = 0 (TRJO1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	ANI24	Analog input	—	—	1	—	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	EI12	Input	—	—	0	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	CLKA0	Output	—	—	0	—	0	0	0	SO11 = 1	(VCOUT1) = 0 (TKBO00) = 0 (TRJO1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	SO11	Output	PIOR107 = 0 PIOR106 = 0	—	0	—	0	1	0	—	CLKA0 = 0 (VCOUT1) = 0 (TKBO00) = 0 (TRJO1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(VCOUT1)	Output	PIOR54 = 0 PIOR53 = 1	—	0	—	0	0	0	SO11 = 1	CLKA0 = 0 (TKBO00) = 0 (TRJO1) = 0 (PCLBUZ1) = 0	x	x	✓	✓	✓	✓
	(TKBO00)	Output	PIOR32 = 0 PIOR31 = 1 PIOR30 = 1 or PIOR32 = 1 PIOR31 = 0 PIOR30 = 0 or PIOR32 = 1 PIOR31 = 0 PIOR30 = 1	—	0	—	0	0	0	SO11 = 1	CLKA0 = 0 (VCOUT1) = 0 (TRJO1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(TRJO1) ^{Note 2}	Output	PIOR41 = 1	—	0	—	0	0	0	SO11 = 1	CLKA0 = 0 (VCOUT1) = 0 (TKBO00) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 0 PIOR51 = 0 PIOR50 = 1	—	0	—	0	0	0	SO11 = 1	CLKA0 = 0 (VCOUT1) = 0 (TKBO00) = 0 (TRJO1) = 0	✓	✓	✓	✓	✓	✓
	SEG37	Output	—	—	0	—	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P13	P13	Input	—	—	0	—	1	x	0	—	—	x	✓	✓	✓	✓	✓
		Output	—	—	0	—	0	0/1	0	—	—	x	✓	✓	✓	✓	✓
	ANI25	Analog input	—	—	1	—	1	x	1	—	—	x	✓	✓	✓	✓	✓
	SEG38	Output	—	—	0	—	0	0	1	—	—	x	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (8/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P14	P14	Input	—	—	0	—	1	x	0	—	—	x	x	✓	✓	✓	✓
		Output	—	—	0	—	0	0/1	0	—	TO04 = 0	x	x	✓	✓	✓	✓
	TI04	Input	PIOR14 = 0 PIOR13 = 0	—	0	—	1	x	0	—	—	x	x	✓	✓	✓	✓
	TO04	Output	PIOR14 = 0 PIOR13 = 0	—	0	—	0	0	0	—	—	x	x	✓	✓	✓	✓
	ANI26	Analog input	—	—	1	—	1	x	1	—	—	x	x	✓	✓	✓	✓
	SEG39	Output	—	—	0	—	0	0	1	—	—	x	x	✓	✓	✓	✓
P15	P15	Input	—	—	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	—	—	—	0	0/1	0	—	TO07 = 0	x	x	x	✓	✓	✓
	TI07	Input	PIOR23 = 0 PIOR22 = 0	—	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	TO07	Output	PIOR23 = 0 PIOR22 = 0	—	—	—	0	0	0	—	—	x	x	x	✓	✓	✓
	SEG40	Output	—	—	—	—	0	0	1	—	—	x	x	x	✓	✓	✓
P16	P16	Input	—	x	—	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	—	—	0	0/1	0	SCK00/SCL00 = 1	(VCOUT0) = 0 (TKBO01) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	—	—	0	0/1	0	SCK00/SCL00 = 1	(VCOUT0) = 0 (TKBO01) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	SCK00	Input	PIOR101 = 0 PIOR100 = 0	x	—	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SCK00	Output	PIOR101 = 0 PIOR100 = 0	0/1	—	—	0	1	0	—	(VCOUT0) = 0 (TKBO01) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	SCL00	Output	PIOR101 = 0 PIOR100 = 0	0/1	—	—	0	1	0	—	(VCOUT0) = 0 (TKBO01) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	EI16	Input	—	x	—	—	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(VCOUT0)	Output	PIOR54 = 0 PIOR53 = 1	0	—	—	0	0	0	SCK00/SCL00 = 1	(TKBO01) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	(TKBO01)	Output	PIOR32 = 1 PIOR31 = 0 PIOR30 = 0	0	—	—	0	0	0	SCK00/SCL00 = 1	(VCOUT0) = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	(RTC1HZ)	Output	PIOR57 = 1 PIOR56 = 0 PIOR55 = 0	0	—	—	0	0	0	SCK00/SCL00 = 1	(VCOUT0) = 0 (TKBO01) = 0	✓	✓	✓	✓	✓	✓
	SEG41	Output	—	0	—	—	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (9/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMCExx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P17	P17	Input	—	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	0	0/1	0	SDA00 = 1	(TKBO01) = 0 (TRJIO0) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	—	0	0	0/1	0	SDA00 = 1	(TKBO01) = 0 (TRJIO0) = 0	✓	✓	✓	✓	✓	✓
	SI00	Input	PIOR101 = 0 PIOR100 = 0	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	RxD0	Input	PIOR101 = 0 PIOR100 = 0	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SDA00	Input/output	PIOR101 = 0 PIOR100 = 0	1	—	0	0	1	0	—	(TKBO01) = 0 (TRJIO0) = 0	✓	✓	✓	✓	✓	✓
	EI17	Input	—	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EO17	Output	—	0/1	—	1	0	x	0	—	—	✓	✓	✓	✓	✓	✓
	EXSDI1	Input	PIOR47 = 0	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TKBO01)	Output	PIOR32 = 0 PIOR31 = 1 PIOR30 = 1	0	—	0	0	0	0	SDA00 = 1	(TRJIO0) = 0	✓	✓	✓	✓	✓	✓
	(TRJIO0)	Input	PIOR40 = 1	x	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TRJIO0)	Output	PIOR40 = 1	0	—	0	0	0	0	SDA00 = 1	(TKBO01) = 0	✓	✓	✓	✓	✓	✓
	SEG42	Output	—	0	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (10/46)

Pin Name	Used Function		PIORx	POMxx	PMCAXx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P20	P20	Input	—	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—	(PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	ANI1	Analog input	—	—	1	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	AVREFM	Reference voltage	—	—	1	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	EI20	Input	—	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	(TI00)	Input	PIOR01 = 0 PIOR00 = 1	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	(INTP7)	Input	PIOR95 = 0 PIOR94 = 0 PIOR93 = 1	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 0 PIOR51 = 1 PIOR50 = 0	—	0	0	0	—	—	—	✓	✓	✓	✓	✓	✓
P21	P21	Input	—	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—	(TO00) = 0 (PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	ANI0	Analog input	—	—	1	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	AVREFP	Reference voltage	—	—	1	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	VBAT1	Input	—	—	0	0	1	—	—	—	✓	✓	✓	✓	✓	✓
	EI21	Input	—	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	(TO00)	Output	PIOR01 = 0 PIOR00 = 1	—	0	0	0	—	—	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	(INTP6)	Input	PIOR92 = 0 PIOR91 = 1 PIOR90 = 0	—	0	1	x	—	—	—	✓	✓	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 0 PIOR51 = 1 PIOR50 = 1	—	0	0	0	—	—	(TO00) = 0	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (11/46)

Pin Name	Used Function		PIORx	POMxx	PMCaxx	PMxx	Pxx	PFSEGx (PFCONMy/PFSEG2) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P22	P22	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	SCK21/SCL21 = 1 (SCK00/SCL00) = 1	—	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	SCK21/SCL21 = 1 (SCK00/SCL00) = 1	—	✓	✓	✓	✓	✓	✓
	ANI16	Analog input	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	SCK21	Input	PIOR113 = 0 PIOR112 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SCK21	Output	PIOR113 = 0 PIOR112 = 0	0/1	0	0	1	0	SCK21/SCL21 = 1	—	✓	✓	✓	✓	✓	✓
	SCL21	Output	PIOR113 = 0 PIOR112 = 0	0/1	0	0	1	0	SCK21/SCL21 = 1	—	✓	✓	✓	✓	✓	✓
	(SCK00)	Input	PIOR101 = 0 PIOR100 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SCK00)	Output	PIOR101 = 0 PIOR100 = 1	0/1	0	0	1	0	(SCK00/SCL00) = 1	—	✓	✓	✓	✓	✓	✓
	(SCL00)	Output	PIOR101 = 0 PIOR100 = 1	0/1	0	0	1	0	(SCK00/SCL00) = 1	—	✓	✓	✓	✓	✓	✓
	(INTP7)	Input	PIOR95 = 0 PIOR94 = 1 PIOR93 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SEG29	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P23	P23	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	TRJIO1 = 0	x	x	x	x	✓	✓
	ANI17	Analog input	—	—	1	1	x	1	—	—	x	x	x	x	✓	✓
	TRJIO1 Note 2	Input	PIOR41 = 0	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	TRJIO1 Note 2	Output	PIOR41 = 0	—	0	0	0	0	—	—	x	x	x	x	✓	✓
	SEG30	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓
P24	P24	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	—	x	x	x	x	✓	✓
	ANI18	Analog input	—	—	1	1	x	1	—	—	x	x	x	x	✓	✓
	SEG31	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓
P25	P25	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	CLKA1 = 0	x	x	x	x	✓	✓
	ANI19	Analog input	—	—	1	1	x	1	—	—	x	x	x	x	✓	✓
	CLKA1	Output	—	—	0	0	0	0	—	—	x	x	x	x	✓	✓
	SEG32	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (12/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P26	P26	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	TxDA1 = 1 SDA21 = 1 (SDA00) = 1	(TKBO20) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	TxDA1 = 1 SDA21 = 1 (SDA00) = 1	(TKBO20) = 0	✓	✓	✓	✓	✓	✓
	ANI20	Analog input	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	ANO0	Analog output	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	EI26	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	TxDA1	Output	PIOR133 = 0 PIOR132 = 0	0/1	0	0	1	0	SDA21 = 1 (SDA00) = 1	(TKBO20) = 0	✓	✓	✓	✓	✓	✓
	SI21	Input	PIOR113 = 0 PIOR112 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SDA21	Input/output	PIOR113 = 0 PIOR112 = 0	1	0	0	1	0	TxDA1 = 1 (SDA00) = 1	(TKBO20) = 0	✓	✓	✓	✓	✓	✓
	(TKBO20) ^{Note 2}	Output	PIOR36 = 1 PIOR35 = 0 or PIOR36 = 1 PIOR35 = 1	0	0	0	0	0	TxDA1 = 1 SDA21 = 1 (SDA00) = 1	—	✓	✓	✓	✓	✓	✓
	(INTP5)	Input	PIOR85 = 0 PIOR84 = 1 PIOR83 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(RxD0)	Input	PIOR101 = 0 PIOR100 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SI00)	Input	PIOR101 = 0 PIOR100 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SDA00)	Input/output	PIOR101 = 0 PIOR100 = 1	1	0	0	1	0	TxDA1 = 1 SDA21 = 1	(TKBO20) = 0	✓	✓	✓	✓	✓	✓
	(EXSDI0)	Input	PIOR47 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SEG33	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (13/46)

Pin Name	Used Function		PIORx	POMxx	PMCAXx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P27	P27	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	SO21 = 1 (TxD0/SO00) = 1	(TKBO21) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	SO21 = 1 (TxD0/SO00) = 1	(TKBO21) = 0	✓	✓	✓	✓	✓	✓
	ANI21	Analog input	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	ANO1	Analog output	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	EI27	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	RxDA1	Input	PIOR133 = 0 PIOR132 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SO21	Output	PIOR113 = 0 PIOR112 = 0	0/1	0	0	1	0	(TxD0/SO00) = 1	(TKBO21) = 0	✓	✓	✓	✓	✓	✓
	(TKBO21) ^{Note 2}	Output	PIOR36 = 1 PIOR35 = 1	0	0	0	0	0	SO21 = 1 (TxD0/SO00) = 1	—	✓	✓	✓	✓	✓	✓
	(INTP4)	Input	PIOR82 = 0 PIOR81 = 0 PIOR80 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TxD0)	Output	PIOR101 = 0 PIOR100 = 1	0/1	0	0	1	0	SO21 = 1	(TKBO21) = 0	✓	✓	✓	✓	✓	✓
	(SO00)	Output	PIOR101 = 0 PIOR100 = 1	0/1	0	0	1	0	SO21 = 1	(TKBO21) = 0	✓	✓	✓	✓	✓	✓
	(EXSDI1)	Input	PIOR47 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	SEG34	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (14/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P30	P30	Input	—	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	TO03/REMOOUT = 0 (RTC1HZ) = 0	x	x	x	✓	✓	✓
	TI03	Input	PIOR12 = 0 PIOR11 = 0 PIOR10 = 0	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
	TO03	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 0	—	0	0	0	0	—	(RTC1HZ) = 0	x	x	x	✓	✓	✓
	REMOOUT	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 0	—	0	0	0	0	—	(RTC1HZ) = 0	x	x	x	✓	✓	✓
	TS08	Input/output	—	—	1	1	x	0	—	—	x	x	x	✓	✓	✓
	(INTP3)	Input	PIOR75 = 0 PIOR74 = 0 PIOR73 = 1	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
	(RTC1HZ)	Output	PIOR57 = 0 PIOR56 = 0 PIOR55 = 1	—	0	0	0	0	—	TO03/REMOOUT = 0	x	x	x	✓	✓	✓
	SEG20	Output	—	—	0	0	0	1	—	—	x	x	x	✓	✓	✓
P31	P31	Input	—	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	RTC1HZ = 0 (TO01) = 0	x	x	x	✓	✓	✓
	INTP3	Input	PIOR75 = 0 PIOR74 = 0 PIOR73 = 0	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
	RTC1HZ	Output	PIOR57 = 0 PIOR56 = 0 PIOR55 = 0	—	0	0	0	0	—	RTC1HZ = 0	x	x	x	✓	✓	✓
	TS07	Input/output	—	—	1	1	x	0	—	—	x	x	x	✓	✓	✓
	(TI01)	Input	PIOR04 = 0 PIOR03 = 0 PIOR02 = 1	—	0	1	x	0	—	—	x	x	x	✓	✓	✓
	(TO01)	Output	PIOR04 = 0 PIOR03 = 0 PIOR02 = 1	—	0	0	0	0	—	(TO01) = 0	x	x	x	✓	✓	✓
	SEG21	Output	—	—	0	0	0	1	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (15/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P32	P32	Input	—	—	0	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	—	0	0	0/1	0	—	TO01 = 0 (TO03/REMOOUT) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	TI01	Input	PIOR04 = 0 PIOR03 = 0 PIOR02 = 0	—	0	1	x	0	—	—	x	x	x	x	x	✓
	TO01	Output	PIOR04 = 0 PIOR03 = 0 PIOR02 = 0	—	0	0	0	0	—	(TO03/REMOOUT) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	TS06	Input/output	—	—	1	1	x	0	—	—	x	x	x	x	x	✓
	(TI03)	Input	PIOR12 = 0 PIOR11 = 0 PIOR10 = 1	—	0	1	x	0	—	—	x	x	x	x	x	✓
	(TO03)	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 1	—	0	0	0	0	—	TO01 = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	(REMOOUT)	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 1	—	0	0	0	0	—	TO01 = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	(INTP3)	Input	PIOR75 = 0 PIOR74 = 1 PIOR73 = 0	—	0	1	x	0	—	—	x	x	x	x	x	✓
	(RTC1HZ)	Output	PIOR57 = 0 PIOR56 = 1 PIOR55 = 0	—	0	0	0	0	—	TO01 = 0 (TO03/REMOOUT) = 0	x	x	x	x	x	✓
	SEG22	Output	—	—	0	0	0	1	—	—	x	x	x	x	x	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (16/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P33	P33	Input	—	x	0	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	0	0	0	0/1	0	(SCK30/SCL30) = 1	(TO01) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SCK30/SCL30) = 1	(TO01) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	INTP4	Input	PIOR82 = 0 PIOR81 = 0 PIOR80 = 0	x	0	1	x	0	—	—	x	x	x	x	x	✓
	TS05	Input/output	—	x	1	1	x	0	—	—	x	x	x	x	x	✓
	(TI01)	Input	PIOR04 = 0 PIOR03 = 1 PIOR02 = 0	x	0	1	x	0	—	—	x	x	x	x	x	✓
	(TO01)	Output	PIOR04 = 0 PIOR03 = 1 PIOR02 = 0	0	0	0	0	0	(SCK30/SCL30) = 1	(RTC1HZ) = 0	x	x	x	x	x	✓
	(RTC1HZ)	Output	PIOR57 = 0 PIOR56 = 1 PIOR55 = 1	0	0	0	0	0	(SCK30/SCL30) = 1	(TO01) = 0	x	x	x	x	x	✓
	(SCK30)	Input	PIOR115 = 1 PIOR114 = 0	x	0	1	x	0	—	—	x	x	x	x	x	✓
	(SCK30)	Output	PIOR115 = 1 PIOR114 = 0	0/1	0	0	1	0	—	(TO01) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
	(SCL30)	Output	PIOR115 = 1 PIOR114 = 0	0/1	0	0	1	0	—	(TO01) = 0 (RTC1HZ) = 0	x	x	x	x	x	✓
P34	P34	Input	—	x	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	0	0	0	0/1	0	(SDA30) = 1	(TO01) = 0	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SDA30) = 1	(TO01) = 0	x	x	x	x	✓	✓
	TS04	Input/output	—	x	1	1	x	0	—	—	x	x	x	x	✓	✓
	(TI01)	Input	PIOR04 = 0 PIOR03 = 1 PIOR02 = 1	x	0	1	x	0	—	—	x	x	x	x	✓	✓
	(TO01)	Output	PIOR04 = 0 PIOR03 = 1 PIOR02 = 1	0	0	0	0	0	(SDA30) = 1	—	x	x	x	x	✓	✓
	(INTP0)	Input	PIOR62 = 0 PIOR61 = 1 PIOR60 = 0	x	0	1	x	0	—	—	x	x	x	x	✓	✓
	(SI30)	Input	PIOR115 = 1 PIOR114 = 0	x	0	1	x	0	—	—	x	x	x	x	x	✓
	(RxD3)	Input	PIOR115 = 1 PIOR114 = 0	x	0	1	x	0	—	—	x	x	x	x	✓	✓
	(SDA30)	Input/output	PIOR115 = 1 PIOR114 = 0	1	0	0	1	0	—	(TO01) = 0	x	x	x	x	x	✓
	SEG24	Output	—	0	0	0	0	1	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (17/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P35	P35	Input	—	x	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	0	0	0	0/1	0	(SO30/TxD3) = 1	(VCOUT0) = 0	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SO30/TxD3) = 1	(VCOUT0) = 0	x	x	x	x	✓	✓
	TS03	Input/output	—	x	1	1	x	0	—	—	x	x	x	x	✓	✓
	(VCOUT0)	Output	PIOR54 = 1 PIOR53 = 0	0	0	0	0	0	(SO30/TxD3) = 1	—	x	x	x	x	✓	✓
	(INTP4)	Input	PIOR82 = 0 PIOR81 = 1 PIOR80 = 0	x	0	1	x	0	—	—	x	x	x	x	✓	✓
	(SO30)	Output	PIOR115 = 0 PIOR114 = 1 or PIOR115 = 1 ^{Note 3} PIOR114 = 0	0/1	0	0	1	0	—	(VCOUT0) = 0	x	x	x	x	✓	✓
	(TxD3)	Output	PIOR115 = 0 PIOR114 = 1 or PIOR115 = 1 PIOR114 = 0	0/1	0	0	1	0	—	(VCOUT0) = 0	x	x	x	x	✓	✓
	SEG25	Output	—	0	0	0	0	1	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (18/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P40	P40	Input	—	x	—	1	x	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	0/1	(SCK00/SCL00) = 1	(TO00) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	—	0	0/1	(SCK00/SCL00) = 1	(TO00) = 0	✓	✓	✓	✓	✓	✓
	(TI00)	Input	PIOR01 = 1 PIOR00 = 0	x	—	1	x	—	—	✓	✓	✓	✓	✓	✓
	(TO00)	Output	PIOR01 = 1 PIOR00 = 0	0	—	0	0	(SCK00/SCL00) = 1	—	✓	✓	✓	✓	✓	✓
	(SCK00)	Input	PIOR101 = 1 PIOR100 = 0	x	—	1	x	—	—	x	x	x	x	✓	✓
	(SCK00)	Output	PIOR101 = 1 PIOR100 = 0	0/1	—	0	1	—	(TO00) = 0	x	x	x	x	✓	✓
	(SCL00)	Output	PIOR101 = 1 PIOR100 = 0	0/1	—	0	1	—	(TO00) = 0	x	x	x	✓	✓	✓
P41	P41	Input	—	x	—	1	x	—	—	x	x	x	x	✓	✓
		Output	—	0	—	0	0/1	(TxD0/SO00) = 1 (SDAA0) = 1 (SDA10) = 1	(TO07) = 0	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	—	0	0/1	(TxD0/SO00) = 1 (SDAA0) = 1 (SDA10) = 1	(TO07) = 0	x	x	x	x	✓	✓
	(TI07)	Input	PIOR23 = 0 PIOR22 = 1	x	—	1	x	—	—	x	x	x	x	✓	✓
	(TO07)	Output	PIOR23 = 0 PIOR22 = 1	0	—	0	0	(TxD0/SO00) = 1 (SDAA0) = 1 (SDA10) = 1	—	x	x	x	x	✓	✓
	(INTP6)	Input	PIOR92 = 0 PIOR91 = 1 PIOR90 = 1	x	—	1	x	—	—	x	x	x	x	✓	✓
	(TxD0)	Output	PIOR101 = 1 PIOR100 = 0	0/1	—	0	1	(SDAA0) = 1 (SDA10) = 1	(TO07) = 0	x	x	x	x	✓	✓
	(RxD A3) Note 2	Input	PIOR137 = 0 PIOR136 = 1	x	—	1	x	—	—	x	x	x	x	x	✓
	(SDAA0)	Input/output	PIOR121 = 0 PIOR120 = 1	1	—	0	0	(TxD0/SO00) = 1 (SDA10) = 1	(TO07) = 0	x	x	x	x	✓	✓
	(SO00)	Output	PIOR101 = 1 PIOR100 = 0	0/1	—	0	1	(SDAA0) = 1 (SDA10) = 1	(TO07) = 0	x	x	x	x	✓	✓
	(SI10)	Input	PIOR105 = 0 PIOR104 = 1	x	—	1	x	—	—	x	x	x	x	x	✓
	(SDA10)	Input/output	PIOR105 = 0 PIOR104 = 1	1	—	0	1	(TxD0/SO00) = 1 (SDAA0) = 1	(TO07) = 0	x	x	x	x	x	✓
	(RxD1)	Input	PIOR105 = 0 PIOR104 = 1	x	—	1	x	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (19/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P42	P42	Input	—	x	0	1	x	—	—	x	x	x	✓	✓	✓
		Output	—	0	0	0	0/1	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	TI05	Input	PIOR17 = 0 PIOR16 = 0 PIOR15 = 0	x	0	1	x	—	—	x	x	x	✓	✓	✓
	TO05	Output	PIOR17 = 0 PIOR16 = 0 PIOR15 = 0	0	0	0	0	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	IVREF1	Analog input	—	x	1	1	x	—	—	x	x	x	✓	✓	✓
	TRJ00	Output	PIOR40 = 0	0	0	0	0	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	(INTP7)	Input	PIOR95 = 0 PIOR94 = 1 PIOR93 = 1	x	0	1	x	—	—	x	x	x	✓	✓	✓
	(TxDA1)	Output	PIOR133 = 0 PIOR132 = 1	0/1	0	0	1	(TxDA1) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	(SCLA0)	Input/output	PIOR121 = 0 PIOR120 = 1	1	0	0	0	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (PCLBUZ1) = 0	x	x	x	x	✓	✓
	(RxD0)	Input	PIOR101 = 1 PIOR100 = 0	x	0	1	x	—	—	x	x	x	x	✓	✓
	(TxD1)	Output	PIOR105 = 0 PIOR104 = 1	0/1	0	0	1	(TxDA1) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	x	✓	✓
	(SI00)	Input	PIOR101 = 1 PIOR100 = 0	x	0	1	x	—	—	x	x	x	x	✓	✓
	(SDA00)	Input/output	PIOR101 = 1 PIOR100 = 0	1	0	0	1	(TxDA1) = 1 (TxD1/SO10) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	(SO10)	Output	PIOR105 = 0 PIOR104 = 1	0/1	0	0	1	(TxDA1) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	x	x	✓
	(SO11)	Output	PIOR107 = 0 PIOR106 = 1	0/1	0	0	1	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0 (PCLBUZ1) = 0	x	x	x	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 0 PIOR51 = 1 PIOR50 = 1	0	0	0	0	(TxDA1) = 1 (TxD1/SO10) = 1 (SDA00) = 1 (SO11) = 1	TO05 = 0 TRJ00 = 0 (SCLA0) = 0	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (20/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P43	P43	Input	—	x	0	1	x	—	—	x	x	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SO30/TxD3 = 1 (SDA11) = 1	(TO05) = 0	x	x	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	SO30/TxD3 = 1 (SDA11) = 1	(TO05) = 0	x	x	✓	✓	✓	✓
	IVCMP1	Analog input	—	x	1	1	x	—	—	x	x	✓	✓	✓	✓
	SO30	Output	PIOR115 = 0 PIOR114 = 0	0/1	0	0	1	(SDA11) = 1	(TO05) = 0	x	x	✓	✓	✓	✓
	TxD3	Output	PIOR115 = 0 PIOR114 = 0	0/1	0	0	1	(SDA11) = 1	(TO05) = 0	x	x	✓	✓	✓	✓
	(TI05)	Input	PIOR17 = 0 PIOR16 = 1 PIOR15 = 0	x	0	1	x	—	—	x	x	✓	✓	✓	✓
	(TO05)	Output	PIOR17 = 0 PIOR16 = 1 PIOR15 = 0	0	0	0	0	SO30/TxD3 = 1 (SDA11) = 1	—	x	x	✓	✓	✓	✓
	(INTP4)	Input	PIOR82 = 1 PIOR81 = 1 PIOR80 = 1	x	0	1	x	—	—	x	x	✓	✓	✓	✓
	(RxD A1)	Input	PIOR133 = 0 PIOR132 = 1	x	0	1	x	—	—	x	x	x	✓	✓	✓
	(SI11)	Input	PIOR107 = 0 PIOR106 = 1	x	0	1	x	—	—	x	x	x	✓	✓	✓
	(SDA11)	Input/output	PIOR107 = 0 PIOR106 = 1	1	0	0	1	SO30/TxD3 = 1	(TO05) = 0	x	x	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (21/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P44	P44	Input	—	x	0	1	x	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SDA30 = 1 (SCK11/SCL11) = 1	EXSDO1 = 0 (TO04) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	SDA30 = 1 (SCK11/SCL11) = 1	EXSDO1 = 0 (TO04) = 0	✓	✓	✓	✓	✓	✓
	IVCMP0	Analog input	—	x	1	1	x	—	—	✓	✓	✓	✓	✓	✓
	RxDA2 ^{Note 2}	Input	PIOR135 = 0 PIOR134 = 0	x	0	1	x	—	—	x	✓	✓	✓	✓	✓
	SI30	Input	PIOR115 = 0 PIOR114 = 0	x	0	1	x	—	—	x	x	✓	✓	✓	✓
	SDA30	Input/output	PIOR115 = 0 PIOR114 = 0	1	0	0	1	(SCK11/SCL11) = 1	EXSDO1 = 0 (TO04) = 0	x	✓	✓	✓	✓	✓
	RxD3	Input	PIOR115 = 0 PIOR114 = 0	x	0	1	x	—	—	x	x	✓	✓	✓	✓
	EXSDO1	Output	PIOR46 = 0	0	0	0	0	SDA30 = 1 (SCK11/SCL11) = 1	(TO04) = 0	✓	✓	✓	✓	✓	✓
	(TI04)	Input	PIOR14 = 0 PIOR13 = 1	x	0	1	x	—	—	✓	✓	✓	✓	✓	✓
	(TO04)	Output	PIOR14 = 0 PIOR13 = 1	0	0	0	0	SDA30 = 1 (SCK11/SCL11) = 1	EXSDO1 = 0	✓	✓	✓	✓	✓	✓
	(INTP7)	Input	PIOR95 = 1 PIOR94 = 0 PIOR93 = 0	x	0	1	x	—	—	✓	✓	✓	✓	✓	✓
	(SCK11)	Input	PIOR107 = 0 PIOR106 = 1	x	0	1	x	—	—	x	x	x	✓	✓	✓
	(SCK11)	Output	PIOR107 = 0 PIOR106 = 1	0/1	0	0	1	SDA30 = 1	EXSDO1 = 0 (TO04) = 0	x	x	x	✓	✓	✓
	(SCL11)	Output	PIOR107 = 0 PIOR106 = 1	0/1	0	0	1	SDA30 = 1	EXSDO1 = 0 (TO04) = 0	x	x	✓	✓	✓	✓
P45	P45	Input	—	x	0	1	x	—	—	x	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	TxDA2 = 1 SCK30/SCL30 = 1	EXSDO0 = 0	x	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	TxDA2 = 1 SCK30/SCL30 = 1	EXSDO0 = 0	x	✓	✓	✓	✓	✓
	IVREF0	Analog input	—	x	1	1	x	—	—	x	✓	✓	✓	✓	✓
	TxDA2 ^{Note 2}	Output	PIOR135 = 0 PIOR134 = 0	0/1	0	0	1	SCK30/SCL30 = 1	EXSDO0 = 0	x	✓	✓	✓	✓	✓
	SCK30	Input	PIOR115 = 0 PIOR114 = 0	x	0	1	x	—	—	x	x	✓	✓	✓	✓
	SCK30	Output	PIOR115 = 0 PIOR114 = 0	0/1	0	0	1	TxDA2 = 1	EXSDO0 = 0	x	x	✓	✓	✓	✓
	SCL30	Output	PIOR115 = 0 PIOR114 = 0	0/1	0	0	1	TxDA2 = 1	EXSDO0 = 0	x	✓	✓	✓	✓	✓
	EXSDO0	Output	PIOR46 = 0	0	0	0	0	TxDA2 = 1 SCK30/SCL30 = 1	—	x	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (22/46)

Pin Name	Used Function		PIORx	POMxx	PMCAxx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P46	P46	Input	—	x	—	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	0	—	0	0/1	0	SDA31 = 1 (SCK21/SCL21) = 1 (SCL01/SCK01) = 1	—	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	—	0	0/1	0	SDA31 = 1 (SCK21/SCL21) = 1 (SCL01/SCK01) = 1	—	x	x	x	x	✓	✓
	SI31	Input	PIOR117 = 0 PIOR116 = 0	x	—	1	x	0	—	—	x	x	x	x	x	✓
	SDA31	Input/output	PIOR117 = 0 PIOR116 = 0	1	—	0	1	0	(SCK21/SCL21) = 1 (SCL01/SCK01) = 1	—	x	x	x	x	x	✓
	(SCK21)	Input	PIOR113 = 0 PIOR112 = 1	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	(SCK21)	Output	PIOR113 = 0 PIOR112 = 1	0/1	—	0	1	0	SDA31 = 1 (SCL01/SCK01) = 1	—	x	x	x	x	✓	✓
	(SCL21)	Output	PIOR113 = 0 PIOR112 = 1	0/1	—	0	1	0	SDA31 = 1 (SCL01/SCK01) = 1	—	x	x	x	x	✓	✓
	(SCL01)	Output	PIOR103 = 0 PIOR102 = 1	0/1	—	0	1	0	SDA31 = 1 (SCK21/SCL21) = 1	—	x	x	x	x	✓	✓
	(SCK01)	Input	PIOR103 = 0 PIOR102 = 1	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	(SCK01)	Output	PIOR103 = 0 PIOR102 = 1	0/1	—	0	1	0	SDA31 = 1 (SCK21/SCL21) = 1	—	x	x	x	x	✓	✓
	SEG26	Output	—	0	—	0	0	1	—	—	x	x	x	x	✓	✓
P47	P47	Input	—	x	—	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	0	—	0	0/1	0	SO31 = 1 (SDA21) = 1 (SDA01) = 1	—	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	—	0	0/1	0	SO31 = 1 (SDA21) = 1 (SDA01) = 1	—	x	x	x	x	✓	✓
	SO31	Output	PIOR117 = 0 PIOR116 = 0	0/1	—	0	1	0	(SDA21) = 1 (SDA01) = 1	—	x	x	x	x	x	✓
	(RxD A0)	Input	PIOR131 = 1 PIOR130 = 0	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	(SI21)	Input	PIOR113 = 0 PIOR112 = 1	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	(SDA21)	Input/output	PIOR113 = 0 PIOR112 = 1	1	—	0	1	0	SO31 = 1 (SDA01) = 1	—	x	x	x	x	✓	✓
	(SDA01)	Input/output	PIOR103 = 0 PIOR102 = 1	1	—	0	1	0	SO31 = 1 (SDA21) = 1	—	x	x	x	x	✓	✓
	(SI01)	Input	PIOR103 = 0 PIOR102 = 1	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	(INTP1)	Input	PIOR65 = 1 PIOR64 = 1 PIOR63 = 0	x	—	1	x	0	—	—	x	x	x	x	✓	✓
	SEG27	Output	—	0	—	0	0	1	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (23/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCLVx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P50	P50	Input	—	x	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	0	0	0	0/1	0	TxDA3 = 1	—	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	TxDA3 = 1	—	x	x	x	x	✓	✓
	TS23	Input/output	—	x	1	1	x	0	—	—	x	x	x	x	✓	✓
	TxDA3 Note 2	Output	PIOR137 = 0 PIOR136 = 0	0/1	0	0	1	0	—	—	x	x	x	x	✓	✓
	SEG4	Output	—	0	0	0	0	1	—	—	x	x	x	x	✓	✓
P51	P51	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	—	x	x	x	x	✓	✓
	TS22	Input/output	—	—	1	1	x	0	—	—	x	x	x	x	✓	✓
	RxDA3 Note 2	Input	PIOR137 = 0 PIOR136 = 0	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	(INTP6)	Input	PIOR92 = 1 PIOR91 = 1 PIOR90 = 0	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	SEG5	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓
P52	P52	Input	—	x	0	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	0	0	0	0/1	0	SO01 = 1	TO00 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	SO01 = 1	TO00 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	TI00	Input	PIOR01 = 0 PIOR00 = 0	x	0	1	x	0	—	—	x	x	x	✓	✓	✓
	TO00	Output	PIOR01 = 0 PIOR00 = 0	0	0	0	0	0	SO01 = 1	(PCLBUZ0) = 0	x	x	x	✓	✓	✓
	INTP1	Input	PIOR65 = 0 PIOR64 = 0 PIOR63 = 0	x	0	1	x	0	—	—	x	x	x	✓	✓	✓
	TS21	Input/output	—	x	1	1	x	0	—	—	x	x	x	✓	✓	✓
	SO01	Output	PIOR103 = 0 PIOR102 = 0	0/1	0	0	1	0	—	TO00 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 1 PIOR51 = 1 PIOR50 = 0	0	0	0	0	0	SO01 = 1	TO00 = 0	x	x	x	✓	✓	✓
	SEG6	Output	—	0	0	0	0	1	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (24/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCONMy/PFSEGz) or ISCLCD (ISCLVx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P53	P53	Input	—	x	0	1	x	0	—	—	x	x	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	SDA01 = 1	(TO07) = 0 (PCLBUZ0) = 0	x	x	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	SDA01 = 1	(TO07) = 0 (PCLBUZ0) = 0	x	x	✓	✓	✓	✓
	INTP2	Input	PIOR72 = 0 PIOR71 = 0 PIOR70 = 0	x	0	1	x	0	—	—	x	x	✓	✓	✓	✓
	TS20	Input/output	—	x	1	1	x	0	—	—	x	x	✓	✓	✓	✓
	SI01	Input	PIOR103 = 0 PIOR102 = 0	x	0	1	x	0	—	—	x	x	x	✓	✓	✓
	SDA01	Input/output	PIOR103 = 0 PIOR102 = 0	1	0	0	1	0	—	(TO07) = 0 (PCLBUZ0) = 0	x	x	✓	✓	✓	✓
	(TI07)	Input	PIOR23 = 1 PIOR22 = 0	x	0	1	x	0	—	—	x	x	✓	✓	✓	✓
	(TO07)	Output	PIOR23 = 1 PIOR22 = 0	0	0	0	0	0	SDA01 = 1	(PCLBUZ0) = 0	x	x	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 1 PIOR51 = 0 PIOR50 = 0	0	0	0	0	0	SDA01 = 1	(TO07) = 0	x	x	✓	✓	✓	✓
P54	SEG7	Output	—	0	0	0	0	1	—	—	x	x	✓	✓	✓	✓
	P54	Input	—	x	0	1	x	0	—	—	x	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	SCK01/SCL01 = 1	TO02 = 0 (PCLBUZ0) = 0	x	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	SCK01/SCL01 = 1	TO02 = 0 (PCLBUZ0) = 0	x	✓	✓	✓	✓	✓
	TI02	Input	PIOR06 = 0 PIOR05 = 0	x	0	1	x	0	—	—	x	✓	✓	✓	✓	✓
	TO02	Output	PIOR06 = 0 PIOR05 = 0	0	0	0	0	0	SCK01/SCL01 = 1	(PCLBUZ0) = 0	x	✓	✓	✓	✓	✓
	TS19	Input/output	—	x	1	1	x	0	—	—	x	✓	✓	✓	✓	✓
	SCK01	Input	PIOR103 = 0 PIOR102 = 0	x	0	1	x	0	—	—	x	x	x	✓	✓	✓
	SCK01	Output	PIOR103 = 0 PIOR102 = 0	0/1	0	0	1	0	—	TO02 = 0 (PCLBUZ0) = 0	x	x	x	✓	✓	✓
	SCL01	Output	PIOR103 = 0 PIOR102 = 0	0/1	0	0	1	0	—	TO02 = 0 (PCLBUZ0) = 0	x	x	✓	✓	✓	✓
	(INTP0)	Input	PIOR62 = 0 PIOR61 = 1 PIOR60 = 1	x	0	1	x	0	—	—	x	✓	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 1 PIOR51 = 0 PIOR50 = 1	0	0	0	0	0	SCK01/SCL01 = 1	TO02 = 0	x	✓	✓	✓	✓	✓
	SEG8	Output	—	0	0	0	0	1	—	—	x	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (25/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P55	P55	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	(TO02) = 0	x	x	x	x	✓	✓
	INTP5	Input	PIOR85 = 0 PIOR84 = 0 PIOR83 = 0	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	TS18	Input/output	—	—	1	1	x	0	—	—	x	x	x	x	✓	✓
	(TI02)	Input	PIOR06 = 0 PIOR05 = 1	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	(TO02)	Output	PIOR06 = 0 PIOR05 = 1	—	0	0	0	0	—	—	x	x	x	x	✓	✓
	SEG9	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓
P56	P56	Input	—	—	0	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	0	0/1	0	—	TO06 = 0	x	x	x	x	✓	✓
	TI06	Input	PIOR21 = 0 PIOR20 = 0	—	0	1	x	0	—	—	x	x	x	x	✓	✓
	TO06	Output	PIOR21 = 0 PIOR20 = 0	—	0	0	0	0	—	—	x	x	x	x	✓	✓
	TS17	Input/output	—	—	1	1	x	0	—	—	x	x	x	x	✓	✓
	SEG10	Output	—	—	0	0	0	1	—	—	x	x	x	x	✓	✓
P57	P57	Input	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	(TO03/REMOOUT) = 0	✓	✓	✓	✓	✓	✓
	INTP6	Input	PIOR92 = 0 PIOR91 = 0 PIOR90 = 0	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	TSCAP	—	—	—	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR12 = 0 PIOR11 = 1 PIOR10 = 0	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO03)	Output	PIOR12 = 0 PIOR11 = 1 PIOR10 = 0	—	0	0	0	0	—	—	✓	✓	✓	✓	✓	✓
	(REMOOUT)	Output	PIOR12 = 0 PIOR11 = 1 PIOR10 = 0	—	0	0	0	0	—	—	✓	✓	✓	✓	✓	✓
	SEG11	Output	—	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (26/46)

Pin Name	Used Function		PIORx	POMxx	PMCExx	CCDE	CCSx	PMxx	Px	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P60	P60	Input	—	—	0	CCDE04 = 0	CCS4x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	CCDE04 = 0	CCS4x = 000	0	0/1	—	SCLA0 = 0 (TO01) = 0	✓	✓	✓	✓	✓	✓
	SCLA0	Input/output	PIOR121 = 0 PIOR120 = 0	—	0	CCDE04 = 0	CCS4x = 000	0	0	—	(TO01) = 0	✓	✓	✓	✓	✓	✓
	CCD04	Output	—	—	0	CCDE04 = 1	CCS4x = 001 to 100	0	0	—	SCLA0 = 0 (TO01) = 0	✓	✓	✓	✓	✓	✓
	EO60	Output	—	—	1	CCDE04 = 0	CCS4x = 000	0	x	—	—	✓	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR04 = 1 PIOR03 = 0 PIOR02 = 0	—	0	CCDE04 = 0	CCS4x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓
	(TO01)	Output	PIOR04 = 1 PIOR03 = 0 PIOR02 = 0	—	0	CCDE04 = 0	CCS4x = 000	0	0	—	SCLA0 = 0	✓	✓	✓	✓	✓	✓
	(INTP3)	Input	PIOR75 = 0 PIOR74 = 1 PIOR73 = 1	—	0	CCDE04 = 0	CCS4x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓
P61	P61	Input	—	—	0	CCDE05 = 0	CCS5x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	CCDE05 = 0	CCS5x = 000	0	0/1	—	SDAA0 = 0 (TO02) = 0	✓	✓	✓	✓	✓	✓
	SDAA0	Input/output	PIOR121 = 0 PIOR120 = 0	—	0	CCDE05 = 0	CCS5x = 000	0	0	—	(TO02) = 0	✓	✓	✓	✓	✓	✓
	CCD05	Output	—	—	0	CCDE05 = 1	CCS5x = 001 to 100	0	0	—	SDAA0 = 0 (TO02) = 0	✓	✓	✓	✓	✓	✓
	EO61	Output	—	—	1	CCDE05 = 0	CCS5x = 000	0	x	—	—	✓	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR06 = 1 PIOR05 = 0	—	0	CCDE05 = 0	CCS5x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓
	(TO02)	Output	PIOR06 = 1 PIOR05 = 0	—	0	CCDE05 = 0	CCS5x = 000	0	0	—	SDAA0 = 0	✓	✓	✓	✓	✓	✓
	(INTP4)	Input	PIOR82 = 0 PIOR81 = 1 PIOR80 = 1	—	0	CCDE05 = 0	CCS5x = 000	1	x	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (27/46)

Pin Name	Used Function		PIORx	POMxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P62	P62	Input	—	—	—	CCDE06 = 0	CCS6x = 000	1	x	—	—	x	x	x	✓	✓	✓
		Output	—	—	—	CCDE06 = 0	CCS6x = 000	0	0/1	—	SCLA1 = 0 (TO03/REMOOUT) = 0	x	x	x	✓	✓	✓
	SCLA1	Input/output	PIOR123 = 0 PIOR122 = 0	—	—	CCDE06 = 0	CCS6x = 000	0	0	—	(TO03/REMOOUT) = 0	x	x	x	✓	✓	✓
	CCD06	Output	—	—	—	CCDE06 = 1	CCS6x = 001 to 100	0	0	—	SCLA1 = 0 (TO03/REMOOUT) = 0	x	x	x	✓	✓	✓
	(TI03)	Input	PIOR12 = 0 PIOR11 = 1 PIOR10 = 1	—	—	CCDE06 = 0	CCS6x = 000	1	x	—	—	x	x	x	✓	✓	✓
	(TO03)	Output	PIOR12 = 0 PIOR11 = 1 PIOR10 = 1	—	—	CCDE06 = 0	CCS6x = 000	0	0	—	SCLA1 = 0	x	x	x	✓	✓	✓
	(REMOOUT)	Output	PIOR12 = 0 PIOR11 = 1 PIOR10 = 1	—	—	CCDE06 = 0	CCS6x = 000	0	0	—	SCLA1 = 0	x	x	x	✓	✓	✓
	(INTP5)	Input	PIOR85 = 0 PIOR84 = 1 PIOR83 = 1	—	—	CCDE06 = 0	CCS6x = 000	1	x	—	—	x	x	x	✓	✓	✓
P63	P63	Input	—	—	—	CCDE07 = 0	CCS7x = 000	1	x	—	—	x	x	x	✓	✓	✓
		Output	—	—	—	CCDE07 = 0	CCS7x = 000	0	0/1	—	SDAA1 = 0 (TO04) = 0	x	x	x	✓	✓	✓
	SDAA1	Input/output	PIOR123 = 0 PIOR122 = 0	—	—	CCDE07 = 0	CCS7x = 000	0	0	—	(TO04) = 0	x	x	x	✓	✓	✓
	CCD07	Output	—	—	—	CCDE07 = 1	CCS7x = 001 to 100	0	0	—	SDAA1 = 0 (TO04) = 0	x	x	x	✓	✓	✓
	(TI04)	Input	PIOR14 = 1 PIOR13 = 0	—	—	CCDE07 = 0	CCS7x = 000	1	x	—	—	x	x	x	✓	✓	✓
	(TO04)	Output	PIOR14 = 1 PIOR13 = 0	—	—	CCDE07 = 0	CCS7x = 000	0	0	—	SDAA1 = 0	x	x	x	✓	✓	✓
	(INTP6)	Input	PIOR92 = 1 PIOR91 = 0 PIOR90 = 0	—	—	CCDE07 = 0	CCS7x = 000	1	x	—	—	x	x	x	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (28/46)

Pin Name	Used Function		PIORx	POMxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P64	P64	Input	—	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	—	—	—	0	0/1	—	(SDAA1) = 0 (TO03/REMOOUT) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	—	—	—	0	0/1	—	(SDAA1) = 0 (TO03/REMOOUT) = 0	x	x	x	x	x	✓
	(INTP5)	Input	PIOR85 = 1 PIOR84 = 0 PIOR83 = 0	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(SDAA1)	Input/output	PIOR123 = 0 PIOR122 = 1	1	—	—	—	0	0	—	(TO03/REMOOUT) = 0	x	x	x	x	x	✓
	(TI03)	Input	PIOR12 = 1 PIOR11 = 0 PIOR10 = 0	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(TO03)	Output	PIOR12 = 1 PIOR11 = 0 PIOR10 = 0	0	—	—	—	0	0	—	(SDAA1) = 0	x	x	x	x	x	✓
	(REMOOUT)	Output	PIOR12 = 1 PIOR11 = 0 PIOR10 = 0	0	—	—	—	0	0	—	(SDAA1) = 0	x	x	x	x	x	✓
P65	P65	Input	—	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	—	—	—	0	0/1	—	(SCLA1) = 0 (TO05) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	—	—	—	0	0/1	—	(SCLA1) = 0 (TO05) = 0	x	x	x	x	x	✓
	(INTP0)	Input	PIOR62 = 1 PIOR61 = 0 PIOR60 = 0	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(SCLA1)	Input/output	PIOR123 = 0 PIOR122 = 1	1	—	—	—	0	0	—	(TO05) = 0	x	x	x	x	x	✓
	(TI05)	Input	PIOR17 = 1 PIOR16 = 0 PIOR15 = 1	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(TO05)	Output	PIOR17 = 1 PIOR16 = 0 PIOR15 = 1	0	—	—	—	0	0	—	(SCLA1) = 0	x	x	x	x	x	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (29/46)

Pin Name	Used Function		PIORx	POMxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output								SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P66	P66	Input	—	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	—	—	—	0	0/1	{SCK10/ SCL10} = 1 (TxDA3) = 1	(TO05) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	—	—	—	0	0/1	{SCK10/ SCL10} = 1 (TxDA3) = 1	(TO05) = 0	x	x	x	x	x	✓
	(TI05)	Input	PIOR17 = 0 PIOR16 = 1 PIOR15 = 1	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(TO05)	Output	PIOR17 = 0 PIOR16 = 1 PIOR15 = 1	0	—	—	—	0	0	{SCK10/ SCL10} = 1 (TxDA3) = 1	—	x	x	x	x	x	✓
	(INTP1)	Input	PIOR65 = 0 PIOR64 = 0 PIOR63 = 1	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(SCK10)	Input	PIOR105 = 0 PIOR104 = 1	x	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(SCK10)	Output	PIOR105 = 0 PIOR104 = 1	0/1	—	—	—	0	1	(TxDA3) = 1	(TO05) = 0	x	x	x	x	x	✓
	(SCL10)	Output	PIOR105 = 0 PIOR104 = 1	0/1	—	—	—	0	1	(TxDA3) = 1	(TO05) = 0	x	x	x	x	x	✓
	(TxDA3) Note 2	Output	PIOR137 = 0 PIOR136 = 1	0/1	—	—	—	0	1	{SCK10/ SCL10} = 1	(TO05) = 0	x	x	x	x	x	✓
P67	P67	Input	—	—	—	—	—	1	x	—	—	x	x	x	x	x	✓
		Output	—	—	—	—	—	0	0/1	—	(TO01) = 0	x	x	x	x	x	✓
	(INTP2)	Input	PIOR72 = 0 PIOR71 = 0 PIOR70 = 1	—	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(TI01)	Input	PIOR04 = 1 PIOR03 = 1 PIOR02 = 0	—	—	—	—	1	x	—	—	x	x	x	x	x	✓
	(TO01)	Output	PIOR04 = 1 PIOR03 = 1 PIOR02 = 0	—	—	—	—	0	0	—	—	x	x	x	x	x	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (30/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	CODE	CCSx	PMxx	Px	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output										SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P70	P70	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
		Output	—	—	0	—	—	—	0	0/1	0	—	TKBO20 = 0	x	x	x	✓	✓	✓
	KR0	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	TS16	Input/output	—	—	1	—	—	—	1	x	0	—	—	x	x	x	✓	✓	✓
	TKBO20 ^{Note 2}	Output	PIOR36 = 0 PIOR35 = 0 or PIOR36 = 0 PIOR35 = 1	—	0	—	—	—	0	0	0	—	—	x	x	x	✓	✓	✓
	SEG12	Output	—	—	0	—	—	—	0	0	1	—	—	x	x	x	✓	✓	✓
P71	P71	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	—	—	—	0	0/1	0	—	—	x	x	x	x	✓	✓
	KR1	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	TS15	Input/output	—	—	1	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	SEG13	Output	—	—	0	—	—	—	0	0	1	—	—	x	x	x	x	✓	✓
P72	P72	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	—	—	—	0	0/1	0	—	TKBO21 = 0	x	x	x	x	✓	✓
	KR2	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	TS14	Input/output	—	—	1	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	TKBO21 ^{Note 2}	Output	PIOR36 = 0 PIOR35 = 0	—	0	—	—	—	0	0	0	—	—	x	x	x	x	✓	✓
	SEG14	Output	—	—	0	—	—	—	0	0	1	—	—	x	x	x	x	✓	✓
P73	P73	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
		Output	—	—	0	—	—	—	0	0/1	0	—	TKBO10 = 0	x	x	x	x	✓	✓
	KR3	Input	—	—	0	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	TS13	Input/output	—	—	1	—	—	—	1	x	0	—	—	x	x	x	x	✓	✓
	TKBO10 ^{Note 2}	Output	PIOR34 = 0 PIOR33 = 0 or PIOR34 = 0 PIOR33 = 1	—	0	—	—	—	0	0	0	—	—	x	x	x	x	✓	✓
	SEG15	Output	—	—	0	—	—	—	0	0	1	—	—	x	x	x	x	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (31/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	CODE	CCSx	PMxx	Px	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output										SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P74	P74	Input	—	—	0	—	CCDE03 = 0	CCS0x = xxx	1	x	0	—	—	x	x	✓	✓	✓	✓
		Output	—	—	0	—	CCDE03 = 0	CCS0x = xxx	0	0/1	0	—	TKBO11 = 0 (TKBO01) = 0	x	x	✓	✓	✓	✓
	KR4	Input	—	—	0	—	CCDE03 = 0	CCS0x = xxx	1	x	0	—	—	x	x	✓	✓	✓	✓
	TS12	Input/output	—	—	1	—	CCDE03 = 0	CCS0x = xxx	1	x	0	—	—	x	x	✓	✓	✓	✓
	CCD03	Output	—	—	0	—	CCDE03 = 1	CCS0x = 001 to 011	0	0	0	—	TKBO11 = 0 (TKBO01) = 0	x	x	✓	✓	✓	✓
	TKBO11 Note 2	Output	PIOR34 = 0 PIOR33 = 0	—	0	—	CCDE03 = 0	CCS0x = xxx	0	0	0	—	(TKBO01) = 0	x	x	✓	✓	✓	✓
	(TKBO01)	Output	PIOR32 = 0 PIOR31 = 1 PIOR30 = 0	—	0	—	CCDE03 = 0	CCS0x = xxx	0	0	0	—	TKBO11 = 0	x	x	✓	✓	✓	✓
	SEG16	Output	—	—	0	—	CCDE03 = 0	CCS0x = xxx	0	0	1	—	—	x	x	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (32/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	CODE	CCSx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCLVx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output										SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P75	P75	Input	—	x	0	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
		Output	—	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0/1	0	SCK20/ SCL20 = 1	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	—	CCDE02 = 0	CCS0x = xxx	0	0/1	0	SCK20/ SCL20 = 1	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	KR5	Input	—	x	0	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	TS11	Input/output	—	x	1	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	CCD02	Output	—	0/1	0	—	CCDE02 = 1	CCS0x = 001 to 011	0	0	0	—	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	SCK20	Input	PIOR111 = 0 PIOR110 = 0	x	0	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	SCK20	Output	PIOR111 = 0 PIOR110 = 0	0/1	0	—	CCDE02 = 0	CCS0x = xxx	0	1	0	—	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	SCL20	Output	PIOR111 = 0 PIOR110 = 0	0/1	0	—	CCDE02 = 0	CCS0x = xxx	0	1	0	—	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	(TKBO11) ^{Note 2}	Output	PIOR34 = 0 PIOR33 = 1	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0	0	SCK20/ SCL20 = 1	(TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	(TKBO01)	Output	PIOR32 = 0 PIOR31 = 0 PIOR30 = 1	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0	0	SCK20/ SCL20 = 1	(TKBO11) = 0 (TKBO01) = 0 (TO03/ REMOOUT) = 0	x	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR12 = 1 PIOR11 = 0 PIOR10 = 1	x	0	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	(TO03)	Output	PIOR12 = 1 PIOR11 = 0 PIOR10 = 1	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0	0	SCK20/ SCL20 = 1	(TKBO11) = 0 (TKBO01) = 0	x	✓	✓	✓	✓	✓
	(REMOOUT)	Output	PIOR12 = 1 PIOR11 = 0 PIOR10 = 1	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0	0	SCK20/ SCL20 = 1	(TKBO11) = 0 (TKBO01) = 0	x	✓	✓	✓	✓	✓
	(INTP0)	Input	PIOR62 = 1 PIOR61 = 1 PIOR60 = 0	x	0	—	CCDE02 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	SEG17	Output	—	0	0	—	CCDE02 = 0	CCS0x = xxx	0	0	1	—	—	x	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (33/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	CODE	CCSx	PMxx	Px	PFSEGx (PFCONy/PFSEG2) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output										SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P76	P76	Input	—	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	CCDE01 = 0	CCS0x = xxx	0	0/1	0	SDA20 = 1	TKBO00 = 0 (RTC1HZ) = 0 (EXSD01) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	CCDE01 = 0	CCS0x = xxx	0	0/1	0	SDA20 = 1	TKBO00 = 0 (RTC1HZ) = 0 (EXSD01) = 0	✓	✓	✓	✓	✓	✓
	KR6	Input	—	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	TS10	Input/output	—	x	1	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	CCD01	Output	—	0/1	0	0	CCDE01 = 1	CCS0x = 001 to 011	0	0	0	—	TKBO00 = 0 (RTC1HZ) = 0 (EXSD01) = 0	✓	✓	✓	✓	✓	✓
	TKBO00	Output	PIOR32 = 0 PIOR31 = 0 PIOR30 = 0 or PIOR32 = 0 PIOR31 = 0 PIOR30 = 1 or PIOR32 = 0 PIOR31 = 1 PIOR30 = 0	0	0	0	CCDE01 = 0	CCS0x = xxx	0	0	0	SDA20 = 1	(RTC1HZ) = 0 (EXSD01) = 0	✓	✓	✓	✓	✓	✓
	SI20	Input	PIOR111 = 0 PIOR110 = 0	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	x	✓	✓	✓	✓	✓
	SDA20	Input/output	PIOR111 = 0 PIOR110 = 0	1	0	0	CCDE01 = 0	CCS0x = xxx	0	1	0	—	TKBO00 = 0 (RTC1HZ) = 0 (EXSD01) = 0	x	✓	✓	✓	✓	✓
	RxD2	Input	PIOR111 = 0 PIOR110 = 0	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EI76	Input	—	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EO76	Output	—	0/1	0	1	CCDE01 = 0	CCS0x = xxx	0	x	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP2)	Input	PIOR72 = 0 PIOR71 = 1 PIOR70 = 0	x	0	0	CCDE01 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(RTC1HZ)	Output	PIOR57 = 1 PIOR56 = 0 PIOR55 = 1	0	0	0	CCDE01 = 0	CCS0x = xxx	0	0	0	SDA20 = 1	TKBO00 = 0 (EXSD01) = 0	✓	✓	✓	✓	✓	✓
	(EXSD01)	Output	PIOR46 = 1	0	0	0	CCDE01 = 0	CCS0x = xxx	0	0	0	SDA20 = 1	TKBO00 = 0 (RTC1HZ) = 0	✓	✓	✓	✓	✓	✓
	SEG18	Output	—	0	0	0	CCDE01 = 0	CCS0x = xxx	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (34/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMCExx	CODE	CCSx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output										SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P77	P77	Input	—	x	0	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0/1	0	SO20/TxD2 = 1	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	CCDE00 = 0	CCS0x = xxx	0	0/1	0	SO20/TxD2 = 1	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	KR7	Input	—	x	0	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	TS09	Input/output	—	x	1	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	CCD00	Output	—	0/1	0	0	CCDE00 = 1	CCS0x = 001 to 011	0	0	0	—	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	TKBO01	Output	PIOR32 = 0 PIOR31 = 0 PIOR30 = 0	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0	0	SO20/TxD2 = 1	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	SO20	Output	PIOR111 = 0 PIOR110 = 0	0/1	0	0	CCDE00 = 0	CCS0x = xxx	0	1	0	—	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	x	✓	✓	✓	✓	✓
	TxD2	Output	PIOR111 = 0 PIOR110 = 0	0/1	0	0	CCDE00 = 0	CCS0x = xxx	0	1	0	—	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	EI77	Input	—	x	0	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	EO77	Output	—	0/1	0	1	CCDE00 = 0	CCS0x = xxx	0	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TKBO21) Note 2	Output	PIOR36 = 0 PIOR35 = 1	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0	0	SO20/TxD2 = 1	TKBO01 = 0 (TO07) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	(TI07)	Input	PIOR23 = 1 PIOR22 = 1	x	0	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO07)	Output	PIOR23 = 1 PIOR22 = 1	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0	0	SO20/TxD2 = 1	TKBO01 = 0 (TKBO21) = 0 (EXSD00) = 0	✓	✓	✓	✓	✓	✓
	(INTP1)	Input	PIOR65 = 0 PIOR64 = 1 PIOR63 = 0	x	0	0	CCDE00 = 0	CCS0x = xxx	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(EXSD00)	Output	PIOR46 = 1	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0	0	SO20/TxD2 = 1	TKBO01 = 0 (TKBO21) = 0 (TO07) = 0	✓	✓	✓	✓	✓	✓
	SEG19	Output	—	0	0	0	CCDE00 = 0	CCS0x = xxx	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (35/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Pxx	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output					SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P80	P80	Input	—	x	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	(SCK20/SCL20) = 1	—	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	(SCK20/SCL20) = 1	—	x	x	x	x	x	✓
	(INTP2)	Input	PIOR72 = 1 PIOR71 = 1 PIOR70 = 0	x	1	x	—	—	x	x	x	x	x	✓
	(SCK20)	Input	PIOR111 = 0 PIOR110 = 1	x	1	x	—	—	x	x	x	x	x	✓
	(SCK20)	Output	PIOR111 = 0 PIOR110 = 1	0/1	0	1	—	—	x	x	x	x	x	✓
	(SCL20)	Output	PIOR111 = 0 PIOR110 = 1	0/1	0	1	—	—	x	x	x	x	x	✓
P81	P81	Input	—	x	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	(SDA10) = 1	—	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	(SDA10) = 1	—	x	x	x	x	x	✓
	(RxD1)	Input	PIOR105 = 1 PIOR104 = 0	x	1	x	—	—	x	x	x	x	x	✓
	(SI10)	Input	PIOR105 = 1 PIOR104 = 0	x	1	x	—	—	x	x	x	x	x	✓
	(SDA10)	Input/output	PIOR105 = 1 PIOR104 = 0	1	0	1	—	—	x	x	x	x	x	✓
P82	P82	Input	—	x	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	(SCK10/SCL10) = 1	(PCLBUZ0) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	(SCK10/SCL10) = 1	(PCLBUZ0) = 0	x	x	x	x	x	✓
	(SCK10)	Input	PIOR105 = 1 PIOR104 = 0	x	1	x	—	—	x	x	x	x	x	✓
	(SCK10)	Output	PIOR105 = 1 PIOR104 = 0	0/1	0	1	—	(PCLBUZ0) = 0	x	x	x	x	x	✓
	(SCL10)	Output	PIOR105 = 1 PIOR104 = 0	0/1	0	1	—	(PCLBUZ0) = 0	x	x	x	x	x	✓
	(PCLBUZ0)	Output	PIOR52 = 1 PIOR51 = 1 PIOR50 = 0	0	0	0	(SCK10/SCL10) = 1	—	x	x	x	x	x	✓
P83	P83	Input	—	x	1	x	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	—	(SDAA1) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	—	(SDAA1) = 0	x	x	x	x	x	✓
	(INTP1)	Input	PIOR65 = 0 PIOR64 = 1 PIOR63 = 1	x	1	x	—	—	x	x	x	x	x	✓
	(RxDA0)	Input	PIOR131 = 0 PIOR130 = 1	x	1	x	—	—	x	x	x	x	x	✓
	(SDAA1)	Input/output	PIOR123 = 1 PIOR122 = 1	1	0	0	—	—	x	x	x	x	x	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (36/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Px	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P84	P84	Input	—	x	1	x	—	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	—	(TxDA0) = 1	(SCLA1) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	—	(TxDA0) = 1	(SCLA1) = 0	x	x	x	x	x	✓
	(INTP0)	Input	PIOR62 = 1 PIOR61 = 0 PIOR60 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
	(TxDA0)	Output	PIOR131 = 0 PIOR130 = 1	0/1	0	1	—	—	(SCLA1) = 0	x	x	x	x	x	✓
	(SCLA1)	Input/output	PIOR123 = 1 PIOR122 = 1	1	0	0	—	(TxDA0) = 1	—	x	x	x	x	x	✓
P85	P85	Input	—	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	1	(SCK10/SCL10) = 1 (SCK30/SCL30) = 1	(VCOUT0) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0/1	1	(SCK10/SCL10) = 1 (SCK30/SCL30) = 1	(VCOUT0) = 0	✓	✓	✓	✓	✓	✓
	(VCOUT0)	Output	PIOR54 = 1 PIOR53 = 1	0	0	0	1	(SCK10/SCL10) = 1 (SCK30/SCL30) = 1	—	✓	✓	✓	✓	✓	✓
	(INTP4)	Input	PIOR82 = 1 PIOR81 = 0 PIOR80 = 0	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(SCK10)	Input	PIOR105 = 1 PIOR104 = 1	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(SCK10)	Output	PIOR105 = 1 PIOR104 = 1	0/1	0	1	1	(SCK30/SCL30) = 1	(VCOUT0) = 0	✓	✓	✓	✓	✓	✓
	(SCL10)	Output	PIOR105 = 1 PIOR104 = 1	0/1	0	1	1	(SCK30/SCL30) = 1	(VCOUT0) = 0	✓	✓	✓	✓	✓	✓
	(SCK30)	Input	PIOR115 = 0 PIOR114 = 1	x	1	x	1	—	—	x	x	x	x	✓	✓
	(SCK30)	Output	PIOR115 = 0 PIOR114 = 1	0/1	0	1	1	(SCK10/SCL10) = 1	(VCOUT0) = 0	x	x	x	x	✓	✓
	(SCL30)	Output	PIOR115 = 0 PIOR114 = 1	0/1	0	1	1	(SCK10/SCL10) = 1	(VCOUT0) = 0	✓	✓	✓	✓	✓	✓
	VL4	—	—	x	1	x	0	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (37/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P86	P86	Input	—	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	1	(SDA10) = 1	(TO06) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0/1	1	(SDA10) = 1	(TO06) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(TI06)	Input	PIOR21 = 0 PIOR20 = 1	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TO06)	Output	PIOR21 = 0 PIOR20 = 1	0	0	0	1	(SDA10) = 1	(PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(INTP3)	Input	PIOR75 = 1 PIOR74 = 0 PIOR73 = 0	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(RxD1)	Input	PIOR105 = 1 PIOR104 = 1	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(SI10)	Input	PIOR105 = 1 PIOR104 = 1	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(SDA10)	Input/output	PIOR105 = 1 PIOR104 = 1	1	0	1	1	—	(TO06) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 1 PIOR51 = 0 PIOR50 = 0	0	0	0	1	(SDA10) = 1	(TO06) = 0	✓	✓	✓	✓	✓	✓
	VL2	—	—	x	1	x	0	—	—	✓	✓	✓	✓	✓	✓
P87	P87	Input	—	x	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	1	(TxD1/SO10) = 1	—	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0/1	1	(TxD1/SO10) = 1	—	✓	✓	✓	✓	✓	✓
	(TxD1)	Output	PIOR105 = 1 PIOR104 = 1	0/1	0	1	1	—	—	✓	✓	✓	✓	✓	✓
	(SO10)	Output	PIOR105 = 1 PIOR104 = 1	0/1	0	1	1	—	—	✓	✓	✓	✓	✓	✓
	VL1	—	—	x	1	x	0	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (38/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOMy/PFSEGz) or ISCLD (ISCLVx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P90	P90	Input	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	(TO00) = 0	✓	✓	✓	✓	✓	✓
	TS31	Input/output	—	—	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TI00)	Input	PIOR01 = 1 PIOR00 = 1	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO00)	Output	PIOR01 = 1 PIOR00 = 1	—	0	0	0	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP1)	Input	PIOR65 = 1 PIOR64 = 0 PIOR63 = 0	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	COM0	Output	—	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P91	P91	Input	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	—	✓	✓	✓	✓	✓	✓
	TS30	Input/output	—	—	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP2)	Input	PIOR72 = 0 PIOR71 = 1 PIOR70 = 1	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	COM1	Output	—	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P92	P92	Input	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	0	—	(TO01) = 0	✓	✓	✓	✓	✓	✓
	TS29	Input/output	—	—	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR04 = 1 PIOR03 = 0 PIOR02 = 1	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO01)	Output	PIOR04 = 1 PIOR03 = 0 PIOR02 = 1	—	0	0	0	0	—	—	✓	✓	✓	✓	✓	✓
	COM2	Output	—	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P93	P93	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	(SCK01/SCL01) = 1	—	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SCK01/SCL01) = 1	—	✓	✓	✓	✓	✓	✓
	TS28	Input/output	—	x	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP3)	Input	PIOR75 = 1 PIOR74 = 1 PIOR73 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SCK01)	Input	PIOR103 = 1 PIOR102 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SCK01)	Output	PIOR103 = 1 PIOR102 = 0	0/1	0	0	1	0	—	—	✓	✓	✓	✓	✓	✓
	(SCL01)	Output	PIOR103 = 1 PIOR102 = 0	0/1	0	0	1	0	—	—	✓	✓	✓	✓	✓	✓
	COM3	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (39/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P94	P94	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	(SDA01) = 1	(TO06) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SDA01) = 1	(TO06) = 0	✓	✓	✓	✓	✓	✓
	TS27	Input/output	—	x	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP2)	Input	PIOR72 = 1 PIOR71 = 0 PIOR70 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SI01)	Input	PIOR103 = 1 PIOR102 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SDA01)	Input/output	PIOR103 = 1 PIOR102 = 0	1	0	0	1	0	—	(TO06) = 0	✓	✓	✓	✓	✓	✓
	(TI06)	Input	PIOR21 = 1 PIOR20 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO06)	Output	PIOR21 = 1 PIOR20 = 1	0	0	0	0	0	(SDA01) = 1	—	✓	✓	✓	✓	✓	✓
	COM4	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	SEG0	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P95	P95	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	(SO01) = 1	(TO02) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SO01) = 1	(TO02) = 0	✓	✓	✓	✓	✓	✓
	TS26	Input/output	—	x	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR06 = 1 PIOR05 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO02)	Output	PIOR06 = 1 PIOR05 = 1	0	0	0	0	0	(SO01) = 1	—	✓	✓	✓	✓	✓	✓
	(INTP1)	Input	PIOR65 = 1 PIOR64 = 0 PIOR63 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SO01)	Output	PIOR103 = 1 PIOR102 = 0 or PIOR103 = 1 PIOR102 = 1	0/1	0	0	1	0	—	(TO02) = 0	✓	✓	✓	✓	✓	✓
	COM5	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	SEG1	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (40/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCLVx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P96	P96	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	(SDA01) = 1	(SDAA1) = 0 (PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SDA01) = 1	(SDAA1) = 0 (PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	TS25	Input/output	—	x	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(INTP5)	Input	PIOR85 = 1 PIOR84 = 0 PIOR83 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SI01)	Input	PIOR103 = 1 PIOR102 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SDA01)	Input/output	PIOR103 = 1 PIOR102 = 1	1	0	0	1	0	—	(SDAA1) = 0 (PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	(SDAA1)	Input/output	PIOR123 = 1 PIOR122 = 0	1	0	0	0	0	(SDA01) = 1	(PCLBUZ0) = 0	✓	✓	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR52 = 1 PIOR51 = 1 PIOR50 = 1	0	0	0	0	0	(SDA01) = 1	(SDAA1) = 0	✓	✓	✓	✓	✓	✓
	COM6	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	SEG2	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
P97	P97	Input	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	(SCK01/SCL01) = 1	(TO05) = 0 (SCLA1) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	0	(SCK01/SCL01) = 1	(TO05) = 0 (SCLA1) = 0	✓	✓	✓	✓	✓	✓
	TS24	Input/output	—	x	1	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TI05)	Input	PIOR17 = 1 PIOR16 = 0 PIOR15 = 0	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(TO05)	Output	PIOR17 = 1 PIOR16 = 0 PIOR15 = 0	0	0	0	0	0	(SCK01/SCL01) = 1	(SCLA1) = 0	✓	✓	✓	✓	✓	✓
	(SCK01)	Input	PIOR103 = 1 PIOR102 = 1	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
	(SCK01)	Output	PIOR103 = 1 PIOR102 = 1	0/1	0	0	1	0	—	(TO05) = 0 (SCLA1) = 0	✓	✓	✓	✓	✓	✓
	(SCL01)	Output	PIOR103 = 1 PIOR102 = 1	0/1	0	0	1	0	—	(TO05) = 0 (SCLA1) = 0	✓	✓	✓	✓	✓	✓
	(SCLA1)	Input/output	PIOR123 = 1 PIOR122 = 0	1	0	0	0	0	(SCK01/SCL01) = 1	(TO05) = 0	✓	✓	✓	✓	✓	✓
	COM7	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	SEG3	Output	—	0	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (41/46)

Pin Name	Used Function		CMC (EXCLK, OSCSEL, EXCLKS, OSCSELS)	PMxx	Pxx	44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output									
P121	P121	Input	00 xx / 10 xx / 11 xx	1	x	✓	✓	✓	✓	✓	✓
		Output	00 xx / 10 xx / 11 xx	0	0/1	✓	✓	✓	✓	✓	✓
	X1	—	01 xx	1	x	✓	✓	✓	✓	✓	✓
	VBAT0	—	01 xx	1	x	✓	✓	✓	✓	✓	✓
	EI121	Input	00 xx / 10 xx / 11 xx	1	x	✓	✓	✓	✓	✓	✓
P122	P122	Input	00 xx / 10 xx	1	x	✓	✓	✓	✓	✓	✓
		Output	00 xx / 10 xx	0	0/1	✓	✓	✓	✓	✓	✓
	X2	—	01 xx	1	x	✓	✓	✓	✓	✓	✓
	EXCLK	Input	11 xx	1	x	✓	✓	✓	✓	✓	✓
	EI122	Input	00 xx / 10 xx	1	x	✓	✓	✓	✓	✓	✓
P123	P123	Input	xx 00 / xx 10 / xx 11	—	x	✓	✓	✓	✓	✓	✓
	XT1	—	xx 01	—	x	✓	✓	✓	✓	✓	✓
P124	P124	Input	xx 00 / xx 10	—	x	✓	✓	✓	✓	✓	✓
	XT2	—	xx 01	—	x	✓	✓	✓	✓	✓	✓
	EXCLKS	Input	xx 11	—	x	✓	✓	✓	✓	✓	✓

<R>

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (42/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P125	P125	Input	—	x	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	1	(SDA30) = 1	(TO06) = 0 (VCOUT1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	0	0	0/1	1	(SDA30) = 1	(TO06) = 0 (VCOUT1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	TS02	Input/output	—	x	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TI06)	Input	PIOR21 = 1 PIOR20 = 0	x	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TO06)	Output	PIOR21 = 1 PIOR20 = 0	0	0	0	0	1	(SDA30) = 1	(VCOUT1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(VCOUT1)	Output	PIOR54 = 1 PIOR53 = 1	0	0	0	0	1	(SDA30) = 1	(TO06) = 0 (PCLBUZ1) = 0	x	x	✓	✓	✓	✓
	(INTP7)	Input	PIOR95 = 1 PIOR94 = 0 PIOR93 = 1	x	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(RxD3)	Input	PIOR115 = 0 PIOR114 = 1	x	0	1	x	1	—	—	x	x	x	x	✓	✓
	(SI30)	Input	PIOR115 = 0 PIOR114 = 1	x	0	1	x	1	—	—	x	x	x	x	✓	✓
	(SDA30)	Input/output	PIOR115 = 0 PIOR114 = 1	1	0	0	1	1	—	(TO06) = 0 (VCOUT1) = 0 (PCLBUZ1) = 0	✓	✓	✓	✓	✓	✓
	(PCLBUZ1)	Output	PIOR52 = 1 PIOR51 = 0 PIOR50 = 1	0	0	0	0	1	(SDA30) = 1	(TO06) = 0 (VCOUT1) = 0	✓	✓	✓	✓	✓	✓
	VL3	—	—	x	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓
P126	P126	Input	—	—	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	1	—	(TO04) = 0	✓	✓	✓	✓	✓	✓
	TS01	Input/output	—	—	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TI04)	Input	PIOR14 = 1 PIOR13 = 1	—	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TO04)	Output	PIOR14 = 1 PIOR13 = 1	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	(INTP4)	Input	PIOR82 = 1 PIOR81 = 1 PIOR80 = 0	—	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	CAPL	—	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (43/46)

Pin Name	Used Function		PIORx	POMxx	PMCTxx	PMxx	Pxx	PFSEGx (PFCOM/PSSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output							SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P127	P127	Input	—	—	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	1	—	(TO03/REMOOUT) = 0	✓	✓	✓	✓	✓	✓
	TS00	Input/output	—	—	1	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR12 = 1 PIOR11 = 1 PIOR10 = 0	—	0	1	x	1	—	—	✓	✓	✓	✓	✓	✓
	(TO03)	Output	PIOR12 = 1 PIOR11 = 1 PIOR10 = 0	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	(REMOOUT)	Output	PIOR12 = 1 PIOR11 = 1 PIOR10 = 0	—	0	0	0	1	—	—	✓	✓	✓	✓	✓	✓
	CAPH	—	—	—	0	1	x	0	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (44/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Pxx	PFSEGx (PFCOM/PSSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P130	P130	Output	—	0	—	0/1	0	(TxDA0) = 1 (SO21) = 1 (SO01) = 1	TRJO1 = 0	x	x	x	x	✓	✓
		N-ch open-drain output	—	1	—	0/1	0	(TxDA0) = 1 (SO21) = 1 (SO01) = 1	TRJO1 = 0	x	x	x	x	✓	✓
	TRJO1 ^{Note 2}	Output	PIOR41 = 0	0	—	0	0	(TxDA0) = 1 (SO21) = 1 (SO01) = 1	—	x	x	x	x	✓	✓
	(TxDA0)	Output	PIOR131 = 1 PIOR130 = 0	0/1	—	1	0	(SO21) = 1 (SO01) = 1	TRJO1 = 0	x	x	x	x	✓	✓
	(SO21)	Output	PIOR113 = 0 PIOR112 = 1	0/1	—	1	0	(TxDA0) = 1 (SO01) = 1	TRJO1 = 0	x	x	x	x	✓	✓
	(SO01)	Output	PIOR103 = 0 PIOR102 = 1	0/1	—	1	0	(TxDA0) = 1 (SO21) = 1	TRJO1 = 0	x	x	x	x	✓	✓
	SEG28	Output	—	0	—	0	1	—	—	x	x	x	x	✓	✓
P137	P137	Input	—	—	—	x	—	—	—	✓	✓	✓	✓	✓	✓
	INTP0	Input	PIOR62 = 0 PIOR61 = 0 PIOR60 = 0	—	—	x	—	—	—	✓	✓	✓	✓	✓	✓
	EI137	Input	—	—	—	x	—	—	—	✓	✓	✓	✓	✓	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (45/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Px	PFSEGx (PFCONy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) Note 1	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P140	P140	Input	—	x	1	x	—	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	—	SCK31/SCL31 = 1	—	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	—	SCK31/SCL31 = 1	—	x	x	x	x	x	✓
	SCK31	Input	PIOR117 = 0 PIOR116 = 0	x	1	x	—	—	—	x	x	x	x	x	✓
	SCK31	Output	PIOR117 = 0 PIOR116 = 0	0/1	0	1	—	—	—	x	x	x	x	x	✓
	SCL31	Output	PIOR117 = 0 PIOR116 = 0	0/1	0	1	—	—	—	x	x	x	x	x	✓
P141	P141	Input	—	—	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	—	0	0/1	0	—	—	x	x	x	x	x	✓
	(INTP6)	Input	PIOR92 = 1 PIOR91 = 0 PIOR90 = 1	—	1	x	0	—	—	x	x	x	x	x	✓
	(RxD A3) Note 2	Input	PIOR137 = 1 PIOR136 = 0	—	1	x	0	—	—	x	x	x	x	x	✓
	SEG51	Output	—	—	0	0	1	—	—	x	x	x	x	x	✓
P142	P142	Input	—	—	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	—	0	0/1	0	—	—	x	x	x	x	x	✓
	(INTP7)	Input	PIOR95 = 1 PIOR94 = 1 PIOR93 = 0	—	1	x	0	—	—	x	x	x	x	x	✓
	(RxD A2) Note 2	Input	PIOR135 = 1 PIOR134 = 0	—	1	x	0	—	—	x	x	x	x	x	✓
	SEG52	Output	—	—	0	0	1	—	—	x	x	x	x	x	✓
P143	P143	Input	—	x	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	0	(TxDA2) = 1	—	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	0	(TxDA2) = 1	—	x	x	x	x	x	✓
	(INTP5)	Input	PIOR85 = 1 PIOR84 = 1 PIOR83 = 0	x	1	x	0	—	—	x	x	x	x	x	✓
	(TxDA2) Note 2	Output	PIOR135 = 1 PIOR134 = 0	0/1	0	1	0	—	—	x	x	x	x	x	✓
	SEG53	Output	—	0	0	0	1	—	—	x	x	x	x	x	✓
P144	P144	Input	—	x	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	0	—	(SCLA0) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	0	—	(SCLA0) = 0	x	x	x	x	x	✓
	(INTP4)	Input	PIOR82 = 1 PIOR81 = 0 PIOR80 = 1	x	1	x	0	—	—	x	x	x	x	x	✓
	(RxD A1)	Input	PIOR133 = 1 PIOR132 = 0	x	1	x	0	—	—	x	x	x	x	x	✓
	(SCLA0)	Input/output	PIOR121 = 1 PIOR120 = 0	1	0	0	0	—	—	x	x	x	x	x	✓
	SEG54	Output	—	0	0	0	1	—	—	x	x	x	x	x	✓

Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions (46/46)

Pin Name	Used Function		PIORx	POMxx	PMxx	Px	PFSEGx (PFCOMy/PFSEGz) or ISCLCD (ISCVLx/ISCCAP) ^{Note 1}	Alternate Function Output		44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
	Function Name	Input/Output						SAU and UARTA (Excluding Clock Output of UARTA)	Other than SAU and UARTA (Including Clock Output of UARTA)						
P145	P145	Input	—	x	1	x	0	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	0	(TxDA1) = 1	(SDAA0) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	0	(TxDA1) = 1	(SDAA0) = 0	x	x	x	x	x	✓
	(INTP3)	Input	PIOR75 = 1 PIOR74 = 0 PIOR73 = 1	x	1	x	0	—	—	x	x	x	x	x	✓
	(TxDA1)	Output	PIOR133 = 1 PIOR132 = 0	0/1	0	1	0	—	(SDAA0) = 0	x	x	x	x	x	✓
	(SDAA0)	Input/output	PIOR121 = 1 PIOR120 = 0	1	0	0	0	(TxDA1) = 1	—	x	x	x	x	x	✓
	SEG55	Output	—	0	0	0	1	—	—	x	x	x	x	x	✓
P146	P146	Input	—	x	1	x	—	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	—	(TxD2/SO20) = 1 (SO31) = 1	(VCOUT1) = 0	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	—	(TxD2/SO20) = 1 (SO31) = 1	(VCOUT1) = 0	x	x	x	x	x	✓
	(VCOUT1)	Output	PIOR54 = 1 PIOR53 = 0	0	0	0	—	(TxD2/SO20) = 1 (SO31) = 1	—	x	x	x	x	x	✓
	(TxD2)	Output	PIOR111 = 0 PIOR110 = 1	0/1	0	1	—	(SO31) = 1	(VCOUT1) = 0	x	x	x	x	x	✓
	(SO20)	Output	PIOR111 = 0 PIOR110 = 1	0/1	0	1	—	(SO31) = 1	(VCOUT1) = 0	x	x	x	x	x	✓
	(SO31)	Output	PIOR117 = 0 PIOR116 = 1	0/1	0	1	—	(TxD2/SO20) = 1	(VCOUT1) = 0	x	x	x	x	x	✓
	(RxD2) ^{Note 2}	Input	PIOR135 = 0 PIOR134 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
	(INTP7)	Input	PIOR95 = 1 PIOR94 = 1 PIOR93 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
P147	P147	Input	—	x	1	x	—	—	—	x	x	x	x	x	✓
		Output	—	0	0	0/1	—	(TxDA3) = 1 (SDA20) = 1	—	x	x	x	x	x	✓
		N-ch open-drain output	—	1	0	0/1	—	(TxDA3) = 1 (SDA20) = 1	—	x	x	x	x	x	✓
	(INTP2)	Input	PIOR72 = 1 PIOR71 = 0 PIOR70 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
	(TxDA3) ^{Note 2}	Output	PIOR137 = 1 PIOR136 = 0	0/1	0	1	—	(SDA20) = 1	—	x	x	x	x	x	✓
	(RxD2)	Input	PIOR111 = 0 PIOR110 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
	(SI20)	Input	PIOR111 = 0 PIOR110 = 1	x	1	x	—	—	—	x	x	x	x	x	✓
	(SDA20)	Input/output	PIOR111 = 0 PIOR110 = 1	1	0	1	—	(TxDA3) = 1	—	x	x	x	x	x	✓

Note 1. The PESEGz bits correspond to the settings of the following pins: P00 to P07, P10 to P17, P22 to P27, P30 to P35, P46, P47, P50 to P57, P70 to P77, P94 to P97, P130, and P141 to 145. The PFCOMy bits correspond to the settings of the P90 to P93 pins. The ISCVLx bits correspond to the settings of the P85 to P87 and P125 pins. The ISCCAP bit corresponds to the settings of the P126 and P127 pins.

Note 2. Not applicable in 44- to 64-pin products with 64 to 128 Kbytes of flash memory

Note 3. Not applicable in products with 80 or fewer pins

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit manipulation instruction for port register m (Pm)

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

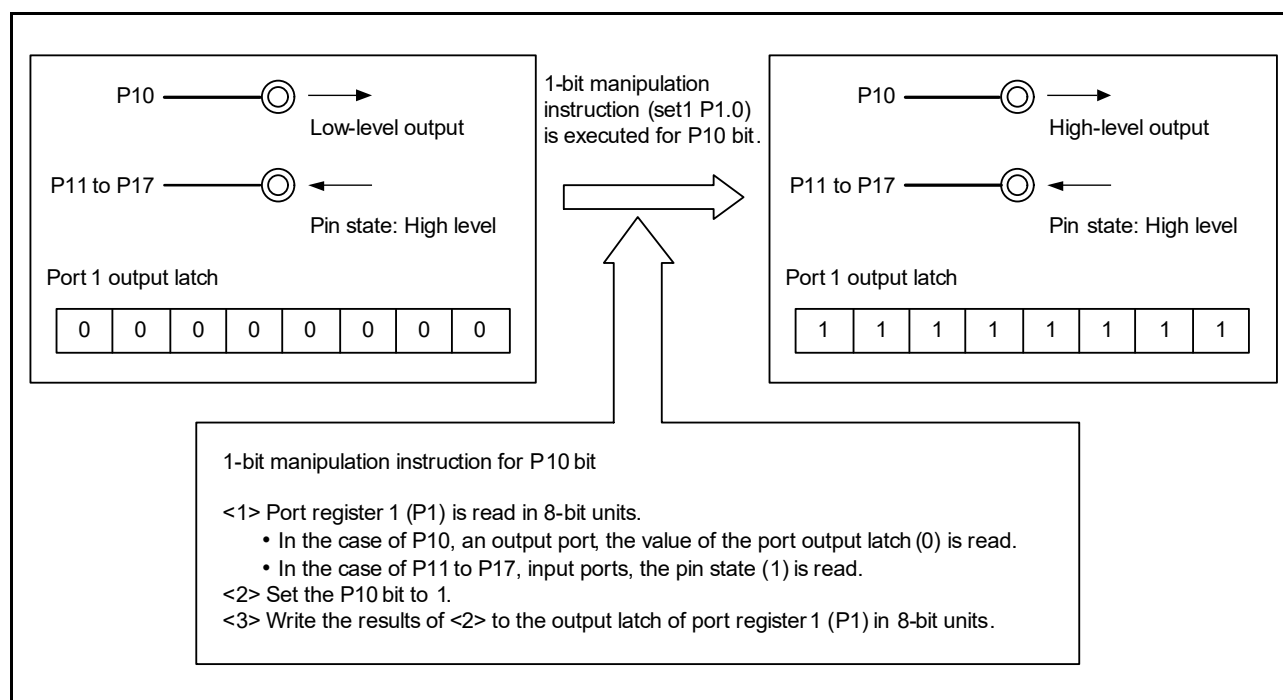
When P10 is an output port, P11 to P17 are input ports (all pins are at the high level), and the output latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 becomes FFH.

Explanation: The targets of writing to and reading from the Pm register of a port whose PMmn 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/L23.

- <1> The Pm register is read in 8-bit units.
- <2> The targeted one bit is manipulated.
- <3> The Pm 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 - 19 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 registers (PIORx). 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 release from the reset state. The mode can be changed by setting of the relevant register.

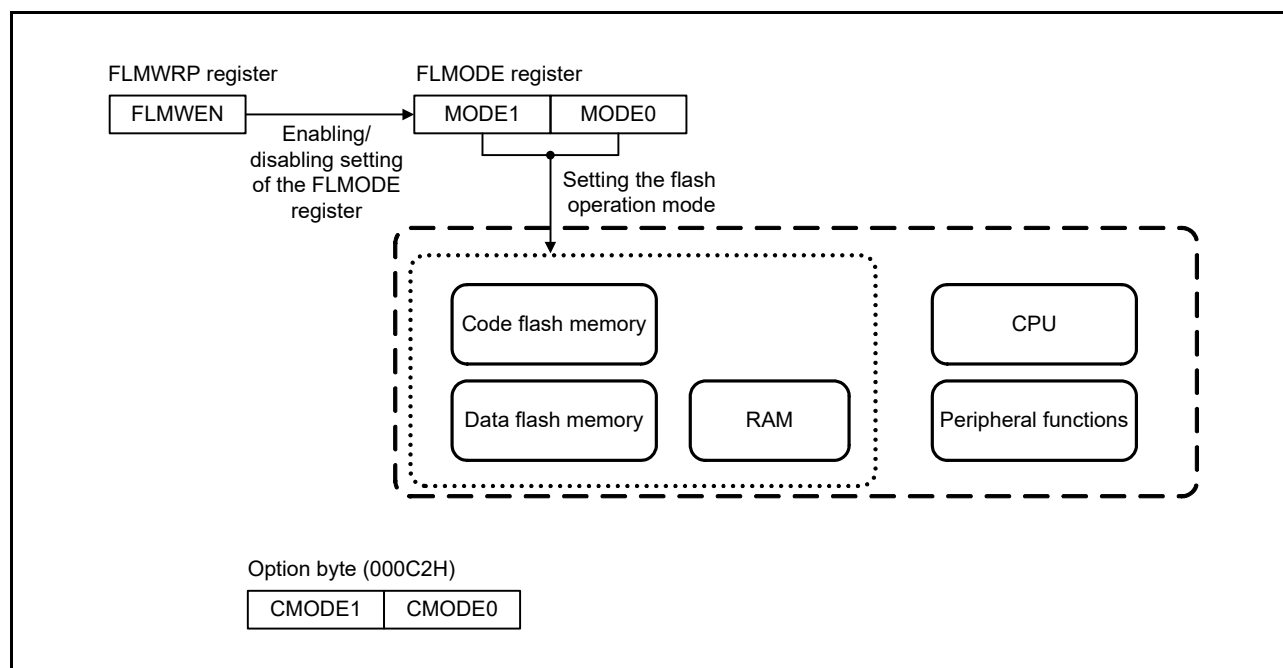
5.1 Configuration of Operation State Control

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 Note 1	1.6 to 1.8 V	1 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.
	1.6 to 1.8 V	1 to 2 MHz	
	1.8 to 5.5 V	1 to 32 MHz	
LS (low-speed main) mode Note 1	1.6 to 1.8 V	1 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.
	1.6 to 1.8 V	1 to 2 MHz	
	1.8 to 5.5 V	1 to 24 MHz	
LP (low-power main) mode Notes 2, 3	1.6 to 5.5 V	1 to 2 MHz (Rewriting of the flash memory is not possible.)	The CPU operates at 1 to 2 MHz in this mode. Low operating current is realized at 1 to 2 MHz.
SUB mode Note 2	1.6 to 5.5 V	32.768 kHz (Rewriting of the flash memory is not possible.)	CPU operation is driven by the subsystem clock. Note 4 This mode enables low-power operation.

Note 1. VDD must be within the range from 2.2 to 5.5 V when rewriting of the flash memory in the bank programming mode is required.

Note 2. Setting of the bank programming mode (FLMODE.BANKPGEN = 1) is prohibited.

Note 3. In LP mode, only the main on-chip oscillator clock (f_{IH} or f_{IM}) can be selected as the CPU/peripheral hardware clock (f_{CLK}).

Note 4. The subsystem clock can be derived from the subsystem clock X (f_{SX}) or the low-speed on-chip oscillator clock (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 register is an 8-bit register for controlling the 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 operating mode protect register (FLMWRP) is 0. The settings of the MODE1 and MODE0 bits following a reset are updated to those of the CMODE1 and CMODE0 bits in the option byte at address 000C2H.

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

Address: F00AAH
 After reset: 40H/80H/C0H^{Note 1}
 R/W: R/W

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

MODE1	MODE0	Selection of flash operation mode
0	0	Setting prohibited
0	1	LP (low-power main) mode ^{Note 2} This setting can be made in LS mode with the main on-chip oscillator clock (f _{IH} or f _{IM}) running at 1 or 2 MHz.
1	0	LS (low-speed main) mode This setting can be made in HS mode when 1 MHz ≤ f _{CLK} ≤ 24 MHz applies or in LP mode.
1	1	HS (high-speed main) mode This setting can be made in LS mode.

Note 1. 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).

Note 2. In LP mode, only the main on-chip oscillator clock (f_{IH} or f_{IM}) can be selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 1. The value of the FLMODE register can be changed when the FLMWEN bit in the flash operating 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 by using the DTC or SMS.

Caution 4. When changing the flash operation mode, make sure that operation is possible within the usable voltage range and operating frequency range for the flash operation mode following the transition 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 is placed in the wait state for the following time until the mode changes. Interrupt requests are held pending during this wait period.

Times for switching between the flash operation modes

Switching between the Flash Operation Modes	Transition 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 Cycles of the CPU/peripheral hardware clock (fCLK)

Caution 6. When rewriting the FLMODE register, insert at least one clock cycle of the CPU/peripheral hardware clock (fCLK) after having rewritten the FLMODE 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 operation 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 register is an 8-bit register for controlling access to the flash operating 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 operating mode select register (FLMODE)
0	Rewriting the FLMODE register is disabled.
1	Rewriting the FLMODE register is enabled.

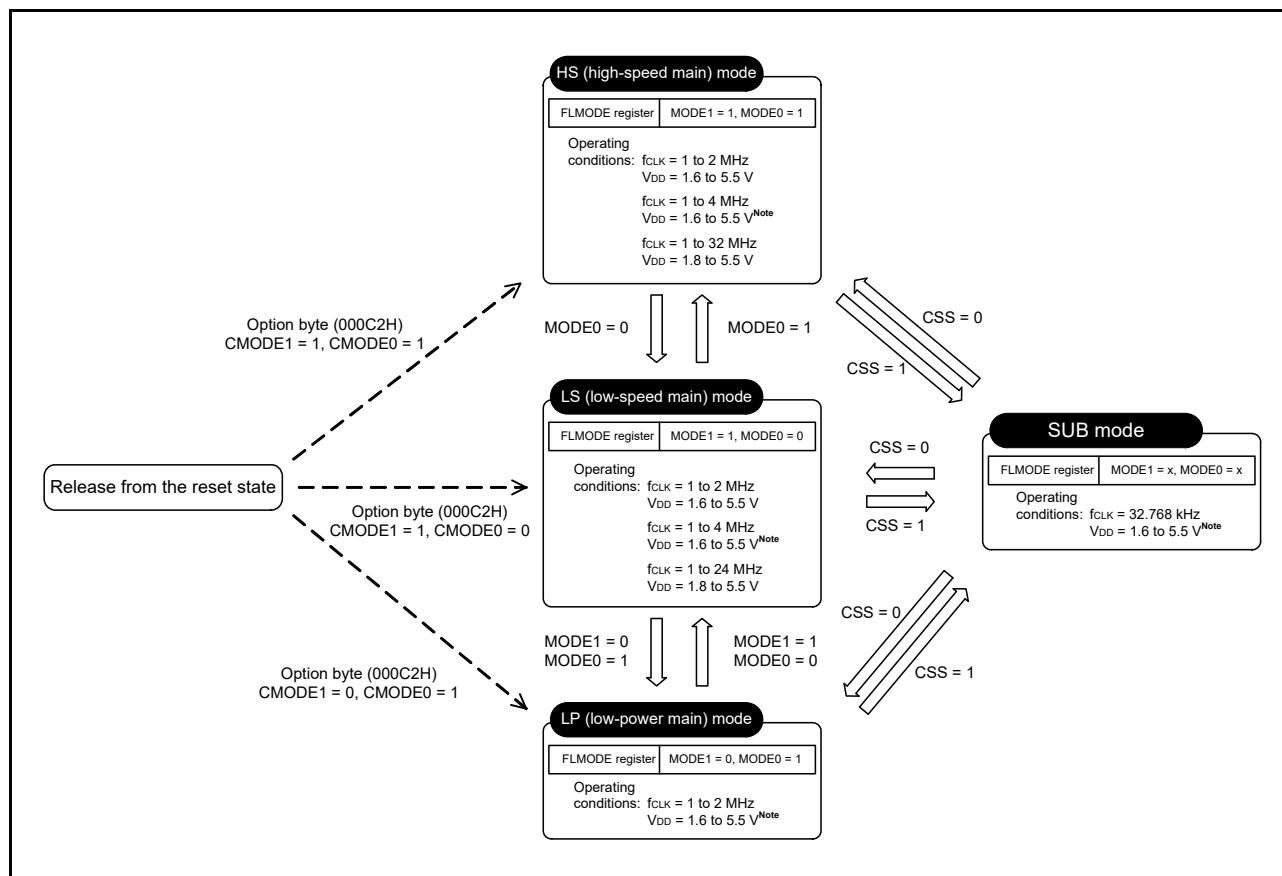
5.3 Initial Setting of Flash Operation Modes

The option byte (000C2H) is used to set the initial state of flash operation mode and the high-speed on-chip oscillator after release from the reset state. Set an appropriate flash operation mode according to the VDD voltage and the high-speed on-chip oscillator frequency at a reset release. When a reset is released, the value of the CMODE1 and CMODE0 bits is reflected in the MODE1 and MODE0 bits in the flash operating mode select register (FLMODE) and the value of the FRQSEL3 to FRQSEL0 bits is reflected in the high-speed on-chip oscillator frequency select register (HOCODIV). For details on the option byte (000C2H), see **Section 37 Option Bytes**.

5.4 Transitions between Flash Operation Modes

Setting of the CMODE1 and CMODE0 bits in the option byte (000C2H) determines the initial flash operation mode immediately after release from the reset state. The initial state can be selected from among HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode. The value of the CMODE1 and CMODE0 bits is reflected in the MODE1 and MODE0 bits in the flash operating 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 to 1 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 1.** When a reset is applied while the MCU operates, operation always starts in the flash operation mode set by the option byte after release from the reset state. Therefore, make sure that operation does not start outside the operating voltage range when a reset is released by setting the LVD detection voltage within the usable operating voltage range of the flash operation mode set in the option byte.
- Caution 2.** VDD must be within the range from 2.2 to 5.5 V when rewriting of the flash memory in the bank programming mode is required. The VDD voltage can be checked by using the voltage detector (LVD) in HS (high-speed main) mode or LS (low-speed main) mode. Setting of the bank programming mode (FLMODE.BANKPGEN = 1) is prohibited in LP (low-power main) mode and SUB mode.

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. The CPU can run in HS mode immediately after release from the reset state. Transition to this mode from LS (low-speed main) mode is possible. The suitable operating range for HS mode is when the supply voltage satisfies $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ and the operating frequency satisfies $24\text{ MHz} < f_{CLK} \leq 32\text{ MHz}$. When 24 MHz or a lower frequency is used for operation, another mode can be used as the suitable flash operation mode. Transition to the bank programming mode (FLMODE.BANKPGEN = 1) from the user mode (FLMODE.BANKPGEN = 0) in HS mode is possible. The supply voltage must satisfy $2.2\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ when rewriting of the flash memory in the bank programming mode is required. For details on the bank programming mode, see **Section 38 Flash Memory**.

Caution Wait for at least 20 μs after release of the CPU from the STOP mode before proceeding with either of the following if required.

- Using the 16-bit timers KB40, KB41, and KB42
- Rewriting of the code flash memory or data flash memory

Figure 5 - 5 Operating Range in HS Mode (User Mode (FLMODE.BANKPGEN = 0))

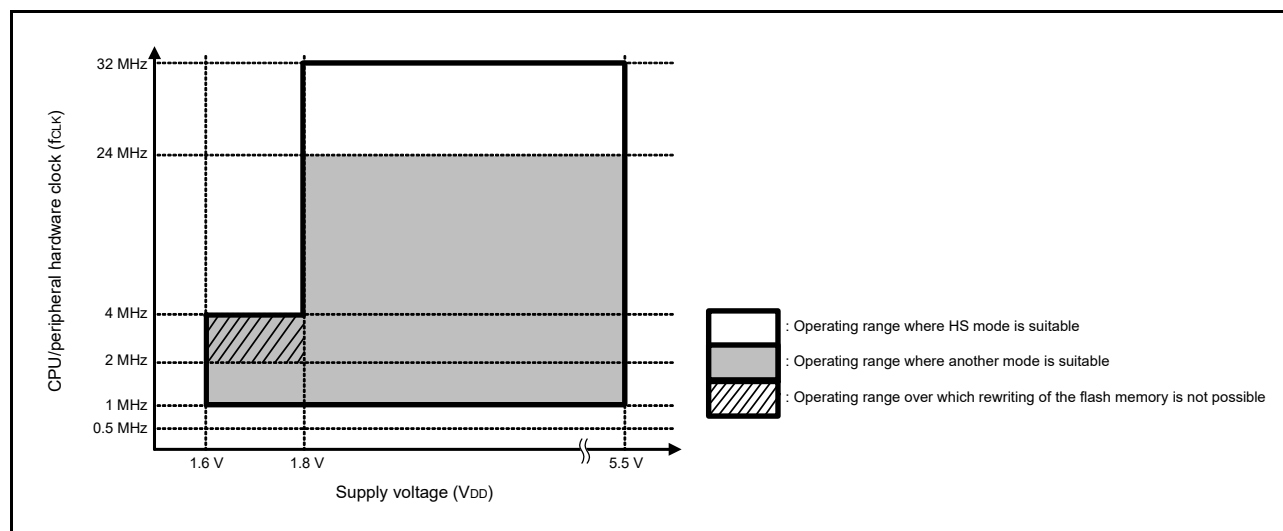
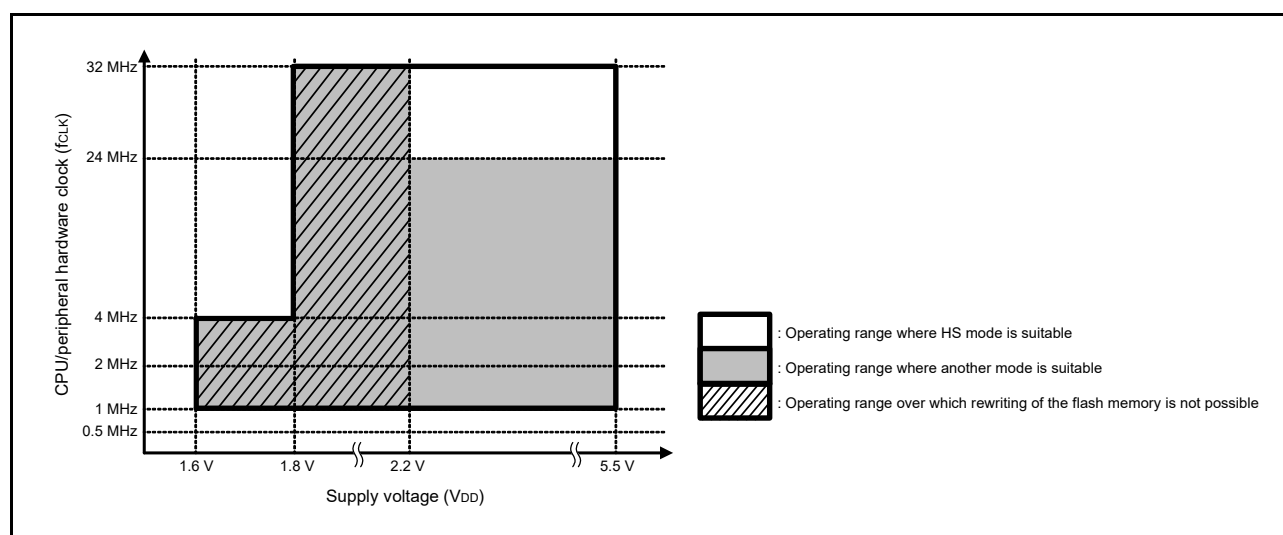


Figure 5 - 6 Operating Range in HS Mode (Bank Programming Mode (FLMODE.BANKPGEN = 1))



5.5.2 Details of LS (low-speed main) mode

LS (low-speed main) mode supports performance in terms of both CPU processing capacity and operating voltage, and is suitable for applications that require low-power consumption and are capable of running at 2 to 24 MHz. The CPU can run in LS mode immediately after release from the reset state. Transition to this mode from HS (high-speed main) mode or LP (low-power main) mode is possible. For a transition from HS mode to LS mode, make sure that the operating frequency satisfies $1 \text{ MHz} \leq f_{\text{CLK}} \leq 24 \text{ MHz}$. The suitable operating range for LS mode is when the supply voltage satisfies $1.8 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ and the operating frequency satisfies $2 \text{ MHz} < f_{\text{CLK}} \leq 24 \text{ MHz}$, or when the supply voltage satisfies $1.6 \text{ V} \leq V_{\text{DD}} < 1.8 \text{ V}$ and the operating frequency satisfies $2 \text{ MHz} < f_{\text{CLK}} \leq 4 \text{ MHz}$ ^{Note}. Transition to the bank programming mode (FLMODE.BANKPGEN = 1) from the user mode (FLMODE.BANKPGEN = 0) in LS mode is possible. When rewriting of the flash memory in the bank programming mode is required, the supply voltage and operating frequency must satisfy $2.2 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ and $1 \text{ MHz} < f_{\text{CLK}} \leq 24 \text{ MHz}$, respectively. For details on the bank programming mode, see **Section 38 Flash Memory**.

Note The operating frequency must satisfy $1 \text{ MHz} \leq f_{\text{CLK}} \leq 2 \text{ MHz}$ when rewriting of the flash memory is required.

Caution Wait for at least 20 μs after release of the CPU from the STOP mode before proceeding with either of the following if required.

- Using the 16-bit timers KB40, KB41, and KB42
- Rewriting of the code flash memory or data flash memory

Figure 5 - 7 Operating Range in LS Mode (User Mode (FLMODE.BANKPGEN = 0))

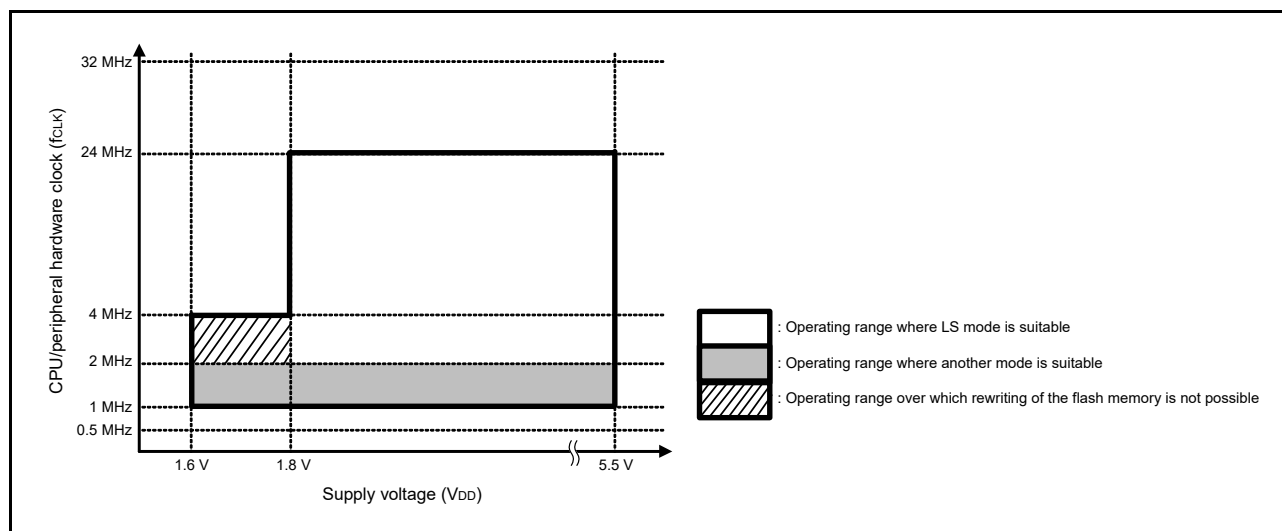
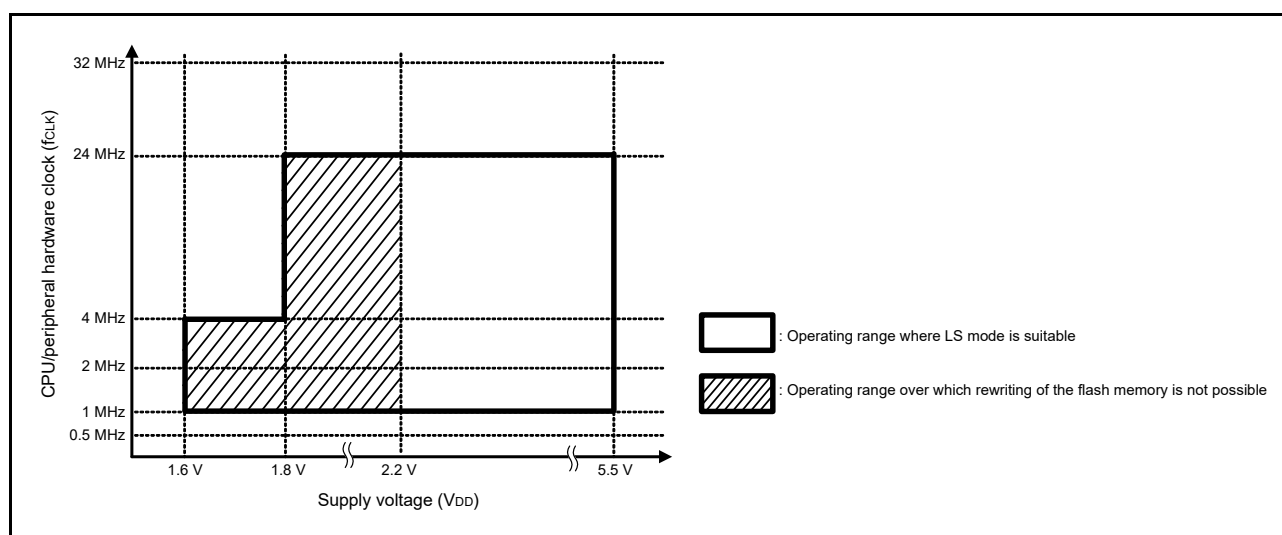


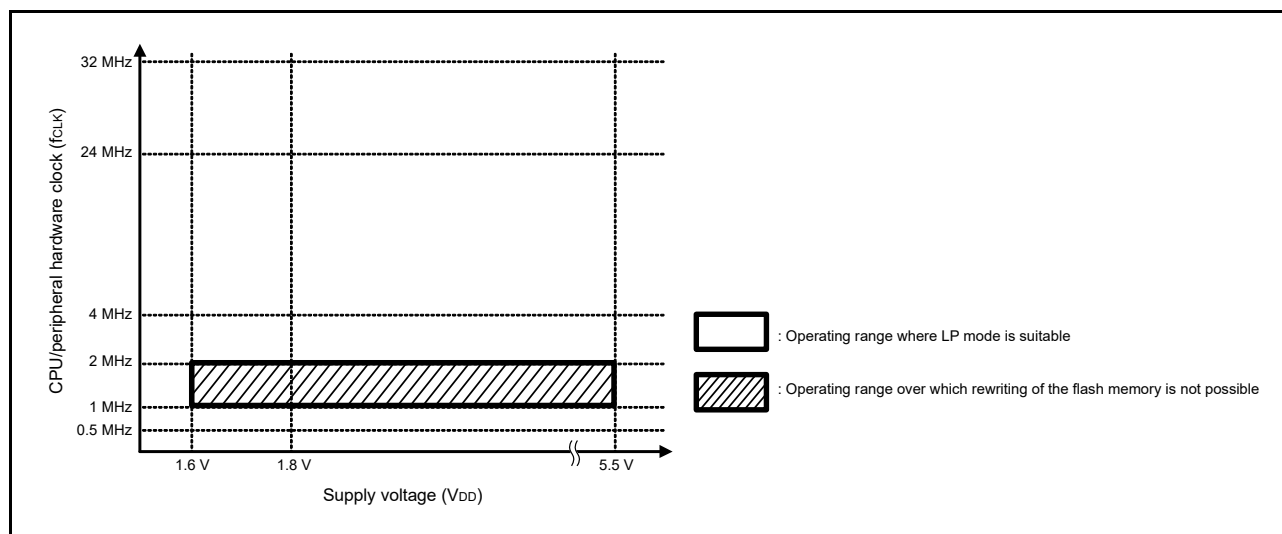
Figure 5 - 8 Operating Range in LS Mode (Bank Programming Mode (FLMODE.BANKPGEN = 1))



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 to 2 MHz. The CPU can run in LP mode immediately after release from the reset state. Transition to this mode from LS (low-speed main) mode is possible. For a transition from LS mode to LP mode, make sure that the operating frequency satisfies $1 \text{ MHz} \leq f_{\text{CLK}} \leq 2 \text{ MHz}$. The suitable operating range for LP mode is when the supply voltage satisfies $1.6 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$ and the operating frequency satisfies $1 \text{ MHz} \leq f_{\text{CLK}} \leq 2 \text{ MHz}$. In LP mode, only the main on-chip oscillator clock (f_{IH} or f_{IM}) can be selected as the CPU/peripheral hardware clock (f_{CLK}). When rewriting of flash memory is required, place the CPU in LS (low-speed main) mode.

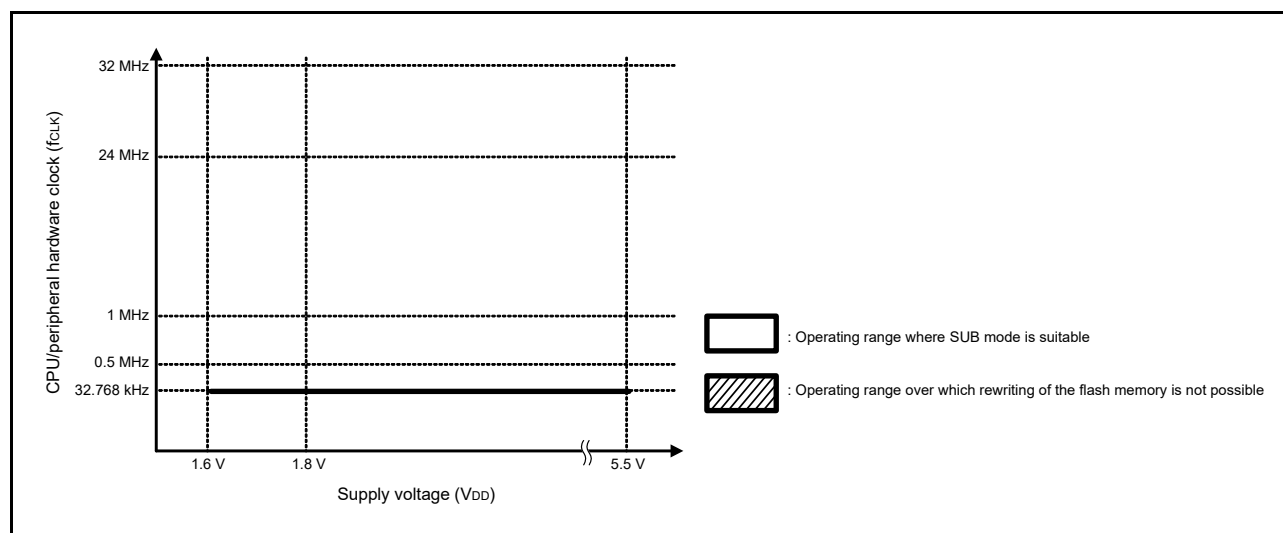
Figure 5 - 9 Operating Range in LP Mode



5.5.4 Details on SUB mode

In SUB mode, the CPU operates at a frequency of 32.768 kHz. Transition to SUB mode from HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode is possible. Setting the CSS bit in the system clock control register (CKC) to 1 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 - 10 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 to produce a clock at $f_x = 1$ 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 $f_{HOCO} = 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2$, or 1 MHz (typ.) by using a user option byte (000C2H). If 64 MHz or 48 MHz is selected for f_{HOCO} , f_{IH} is 32 MHz or 24 MHz, respectively. When 32 MHz or a lower frequency is selected for f_{HOCO} , f_{IH} is not divided, thus having the same frequency as f_{HOCO} . 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 - 13 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	48	64
$2.2\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	✓	✓	✓	✓	—	—	—	—	—	—	—	—

<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[1:0] bits (bits 1, 0 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 CMODE[1:0] bits of a user option byte (000C2H) (see **Section 37 Option Bytes**) or the flash operating mode select register (FLMODE).

2. Subsystem clock

<1> XT1 oscillator

This circuit oscillates to produce a clock at $f_{XT} = 32.768$ 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 to produce a clock at $f_{IL} = 32.768$ 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 a user 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)

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.

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

Remark

f_X : X1 clock oscillation frequency

f_{HOCO} : High-speed on-chip oscillator clock frequency (64 MHz max.)

f_{IH} : High-speed on-chip oscillator clock frequency (32 MHz max.)^{Note}

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

Note When f_{HOCO} is set to 64 or 48 MHz, f_{IH} is obtained by dividing f_{HOCO} by 2 under hardware control.

When 32 MHz or a lower frequency is selected for f_{HOCO} , f_{IH} is not divided under hardware control, thus having the same frequency as f_{HOCO} .

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) • Main oscillator oscillation mode select register (MODRV) • 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 to 2 (PER0 to PER2) • 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) • XT1 oscillator margin checking control register (SOMRG)
Oscillators	• X1 oscillator • XT1 oscillator • High-speed on-chip oscillator • Middle-speed on-chip oscillator • Low-speed on-chip oscillator

Remark	fx: X1 clock oscillation frequency fEX: External main system clock frequency fHOCO: High-speed on-chip oscillator clock frequency (64 MHz max.) fIH: High-speed on-chip oscillator clock frequency (32 MHz max.)Note fIHP: High-speed on-chip oscillator peripheral clock frequency (64 MHz when FRQSEL4 = 1 and FRQSEL3 = 1, 48 MHz when FRQSEL4 = 1 and FRQSEL3 = 0, 32 MHz when FRQSEL4 = 0 and FRQSEL3 = 1, 24 MHz when FRQSEL4 = 0 and 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 fSUBR: Subsystem clock R 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) Note When fHOCO is set to 64 or 48 MHz, fIH is obtained by dividing fHOCO by 2 under hardware control. When 32 MHz or a lower frequency is selected for fHOCO, fIH is not divided under hardware control, thus having the same frequency as fHOCO.

6.3 Registers for Controlling the Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Main oscillator oscillation mode select register (MODRV)
- 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 to 2 (PER0 to PER2)
- 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)
- XT1 oscillator margin checking control register (SOMRG)

Caution Which registers and bit functions are included depends on the product. Be sure to set bits to which no functions are allocated to their initial values.

6.3.1 Clock operation mode control register (CMC)

The CMC 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)

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

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

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

EXCLKS Note 1	OSCSELS 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 2}
0	1	Normal oscillation
1	0	Low power consumption oscillation 2 ^{Note 2}
1	1	Low power consumption oscillation 3 ^{Note 2}

AMPH	Control of the X1 clock oscillation frequency
0	1 MHz ≤ f _x ≤ 10 MHz
1	10 MHz < f _x ≤ 20 MHz

Note 1. The EXCLKS, OSCSELS, and AMPHS[1:0] bits are only initialized by a power-on reset; they retain the values when a reset caused by another source occurs.

Note 2. 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 and AMPHS[1:0] 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 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) 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.
 - In evaluating oscillation by the XT1 oscillator, bits 1 and 0 (SOSCMRG1 and SOSCMRG0) in the XT1 oscillator margin checking control register (SOMRG) can be used to make the settings for checking the lower- and upper-limit margins for oscillation by the XT1 oscillator.
 - 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 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) 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.

Remark fx: X1 clock oscillation frequency

6.3.2 Main oscillator oscillation mode select register (MODRV)

The MODRV register is used to select the low driving mode for the X1 clock. The oscillator requires a limiting resistor. Selecting the low driving mode suppresses the amplitude of oscillation and so minimizes the required resistance. The MODRV 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 - 3 Format of Main Oscillator Oscillation Mode Select Register (MODRV)

Address: F009EH
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MODRV	0	0	0	0	0	0	0	AMPH1

AMPH1	MOSC operating mode
0	Normal mode
1	Low driving mode

- Caution 1.** The MODRV 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 MODRV 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 MODRV register before X1 oscillation is started by setting the clock operation status control register (CSC).
- Caution 3.** Make the setting of the AMPH1 bit 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}).

6.3.3 System clock control register (CKC)

The CKC 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 - 4 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)							
CSS ^{Note 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) ^{Note 3}							
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. Only bits 7, 5, and 1 are readable.

Note 2. Changing the value of the MCM[1:0] bits is prohibited while the CSS bit is set to 1.

Note 3. This setting is prohibited in LP (low-power main) mode.

Caution 1. Be sure to set bits 3 and 2 of the CKC register to 0.

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

- 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, LCD controller/driver, serial interface UARTA, 32-bit interval timer, 8-bit interval timer, timer RJ, oscillation stop detector, external signal sampler, 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 42 Electrical Characteristics.
- 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 subsystem 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})

6.3.4 Clock operation status control register (CSC)

The CSC 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 - 5 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	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 by a power-on reset; it retains its value when a reset caused by another source occurs.

- 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.

6.3.5 Oscillation stabilization time counter status register (OSTC)

The OSTC 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 = 0 and OSCSEL = 1 in the CMC register → MSTOP = 0)
- When the STOP mode is released

Figure 6 - 6 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

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	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.

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

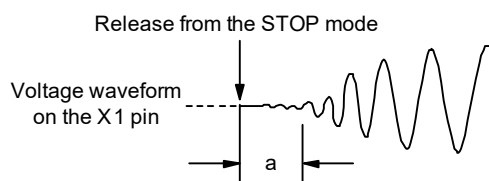
Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in the oscillation stabilization time select register (OSTS).

In the following cases, set the oscillation stabilization time of the OSTC 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 OSTC 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

6.3.6 Oscillation stabilization time select register (OSTS)

The OSTS 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 - 7 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	$2^8/fx$	25.6 μ s	12.8 μ s
0	0	1	$2^9/fx$	51.2 μ s	25.6 μ s
0	1	0	$2^{10}/fx$	102 μ s	51.2 μ s
0	1	1	$2^{11}/fx$	204 μ s	102 μ s
1	0	0	$2^{13}/fx$	819 μ s	409 μ s
1	0	1	$2^{15}/fx$	3.27 ms	1.63 ms
1	1	0	$2^{17}/fx$	13.1 ms	6.55 ms
1	1	1	$2^{18}/fx$	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.

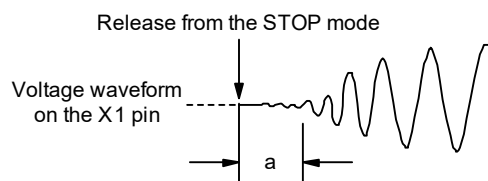
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register after the oscillation starts.

- 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 and Remark are listed on the next page.)

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

6.3.7 Peripheral enable registers 0 to 2 (PER0 to PER2)

The PER0 to PER2 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 IICA
- A/D converter
- Serial array unit
- Timer array unit
- D/A converter
- SNOOZE mode sequencer
- Comparator
- 32-bit interval timer
- DTC
- Serial interface UARTA
- Capacitive sensing unit
- External signal sampler
- Oscillation stop detector
- KB4 16-bit timer and forced output stop function of the KB4 16-bit timer
- RJ timer

The PER0 to PER2 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 - 8 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	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN
	RTCWEN	Control of supply of an input clock to the realtime clock						
	0	- The SFRs used by the realtime clock cannot be written. - The realtime clock can operate.						
	1	- The SFRs used by the realtime clock can be read and written. - The realtime clock can operate.						
	IICA1EN	Control of supply of an input clock to the IICA1 serial interface						
	0	Stops supply of an input clock. The SFRs used by the IICA1 serial interface cannot be written.Note						
	1	Enables supply of an input clock. The SFRs used by the IICA1 serial interface 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.Note						
	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.Note						
	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.Note						
	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.Note						
	1	Enables supply of an input clock. The SFRs used by serial array unit 0 can be read and written.						

Figure 6 - 8 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.Note
1	Enables supply of an input clock. • The SFRs used by timer array unit 0 can be read and written.

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to set bit 1 to 0.

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 - 9 Format of Peripheral Enable Register 1 (PER1) (1/2)

Address: F00FAH
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUN

DACEN	Control of supply of an input clock to the D/A converter
0	Stops supply of an input clock. • The SFRs used by the D/A converter cannot be written. Note
1	Enables supply of an input clock. • The SFRs used by the D/A converter can be read and written.

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. Note
1	Enables supply of an input clock. • The SFRs used by the SNOOZE mode sequencer can be read and written.

CMPEN	Control of supply of an input clock to the comparator
0	Stops supply of an input clock. • The SFRs used by the comparator cannot be written. Note
1	Enables supply of an input clock. • The SFRs used by the comparator 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. Note
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.

UTA0EN	Control of supply of an input clock to the UARTA0 serial interface
0	Stops supply of an input clock. • The SFRs used by the UARTA0 serial interface cannot be written. Note
1	Enables supply of an input clock. • The SFRs used by the UARTA0 serial interface can be read and written.

Figure 6 - 9 Format of Peripheral Enable Register 1 (PER1) (2/2)

CTSUN	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. Note
1	Enables supply of an input clock. • The SFRs used by the capacitive sensing unit can be read and written.

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to set bit 1 to 0.

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.

Figure 6 - 10 Format of Peripheral Enable Register 2 (PER2) (1/2)

Address: F00FCH

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN Note 1	UTA2EN Note 2	UTA1EN	TRJ1EN Note 3	TRJ0EN
	EXSDEN	Control of supply of an input clock to the external signal sampler						
	0	Stops supply of an input clock. • The SFRs used by the external signal sampler cannot be written. Note 4						
	1	Enables supply of an input clock. • The SFRs used by the external signal sampler can be read and written.						
	OSDCEN	Control of supply of an input clock to the oscillation stop detector						
	0	Stops supply of an input clock. • The SFRs used by the oscillation stop detector cannot be written. Note 4						
	1	Enables supply of an input clock. • The SFRs used by the oscillation stop detector can be read and written.						
	TKBEN Note 5	Control of supply of an input clock to the KB4 16-bit timer and forced output stop function of the KB4 16-bit timer						
	0	Stops supply of an input clock. • The SFRs used by the KB4 16-bit timer and forced output stop function of the KB4 16-bit timer cannot be written. Note 4						
	1	Enables supply of an input clock. • The SFRs used by the KB4 16-bit timer and forced output stop function of the KB4 16-bit timer can be read and written.						
	UTA3EN	Control of supply of an input clock to the UARTA3 serial interface						
	0	Stops supply of an input clock. • The SFRs used by the UARTA3 serial interface cannot be written. Note 4 • The UARTA3 serial interface cannot operate.						
	1	Enables supply of an input clock. • The SFRs used by the UARTA3 serial interface can be read and written. • The UARTA3 serial interface can operate.						
	UTA2EN	Control of supply of an input clock to the UARTA2 serial interface						
	0	Stops supply of an input clock. • The SFRs used by the UARTA2 serial interface cannot be written. Note 4 • The UARTA2 serial interface cannot operate.						
	1	Enables supply of an input clock. • The SFRs used by the UARTA2 serial interface can be read and written. • The UARTA2 serial interface can operate.						

Figure 6 - 10 Format of Peripheral Enable Register 2 (PER2) (2/2)

UTA1EN	Control of supply of an input clock to the UARTA1 serial interface
0	Stops supply of an input clock. • The SFRs used by the UARTA1 serial interface cannot be written. Note 4 • The UARTA1 serial interface cannot operate.
1	Enables supply of an input clock. • The SFRs used by the UARTA1 serial interface can be read and written. • The UARTA1 serial interface can operate.

TRJ1EN	Control of supply of an input clock to the RJ1 timer
0	Stops supply of an input clock. • The SFRs used by the RJ1 timer cannot be written. Note 4
1	Enables supply of an input clock. • The SFRs used by the RJ1 timer can be read and written.

TRJ0EN	Control of supply of an input clock to the RJ0 timer
0	Stops supply of an input clock. • The SFRs used by the RJ0 timer cannot be written. Note 4
1	Enables supply of an input clock. • The SFRs used by the RJ0 timer can be read and written.

Note 1. This bit is only present in the 80- and 100-pin products.

Note 2. This bit is only present in the 48- to 100-pin products with 256 or 512 Kbytes of flash memory.

Note 3. This bit is only present in the 44- to 64-pin products with 256 or 512 Kbytes of flash memory and 80- and 100-pin products.

Note 4. When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Note 5. When the setting of the FRQSEL4 bit of the user option byte (000C2H) is 1, select f_{IH} as f_{CLK} before setting bit 5 (TKBEN) of peripheral enable register 2 (PER2).

When changing the source clock of f_{CLK} to a clock other than f_{IH}, clear bit 5 (TKBEN) of peripheral enable register 2 (PER2) before doing so.

Caution Do not change the value of a bit of the PER2 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.

6.3.8 Subsystem clock supply mode control register (OSMC)

The OSMC 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, clock output/buzzer output function, and LCD controller/driver is stopped in STOP mode or in HALT mode while the CPU is operating with the subsystem clock. In addition, the OSMC register can be used to select the operating clock for the realtime clock, serial interface UARTA, clock output/buzzer output controller, LCD controller/driver, 32-bit interval timer, 8-bit interval timer, and external signal sampler. 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^{Note 1}.

Note The RTCLPC and WUTMMCK0 bits have the value 0 and the HIPREC bit has the value 1.

Figure 6 - 11 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
 After reset: Undefined^{Note 1}
 R/W: R/W^{Note 2}

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	HIPREC
	RTCLPC ^{Note 3}	Setting in STOP mode or in HALT mode while the CPU is operating with the subsystem clock X						
	0	Enables supply of the subsystem clock X to peripheral functions (See Table 27 - 1 Operating States in the HALT Mode (1) to Table 27 - 4 Operating States in the SNOOZE Mode for peripheral functions whose operations are enabled.)						
	1	Stops supply of the subsystem clock to peripheral functions other than the realtime clock, clock output/buzzer output function, LCD controller/driver.						
	WUTMMCK0	Selection of the operating clock for the realtime clock, serial interface UARTA, clock output/buzzer output controller, LCD controller/driver, 32-bit interval timer, 8-bit interval timer, and external signal sampler.						
	0	Subsystem clock X						
	1	Low-speed on-chip oscillator clock ^{Notes 4, 5}						
	HIPREC	State of the high-speed on-chip oscillator clock ^{Notes 6, 7}						
	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 8}						
	1	The high-speed on-chip oscillator clock is operating with high precision.						

Note 1. The RTCLPC and WUTMMCK0 bits have the value 0 and the HIPREC bit has the value 1.

Note 2. Be sure to set bits 6, 5, and 1 to 0. Only bits 3, 2, and 0 are readable. Writing to these bits is ignored.

Note 3. 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 (f_{SUB}) is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock (f_{SUB}) is not stopped.

Note 4. Do not set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.

Note 5. 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, serial interface UARTA, clock output/buzzer output controller, LCD controller/driver, 32-bit interval timer, 8-bit interval timer, and external signal sampler are stopped.

Note 6. Undefined while the high-speed on-chip oscillator is stopped.

Note 7. For frequency accuracy of the high-speed on-chip oscillator clock, see **Section 42 Electrical Characteristics**.

(Note and Caution are listed on the next page.)

Note 8. When the RL78/L23 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.

6.3.9 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 - 12 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 clockNote							

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 fMAIN) and confirm that the value of the CLS bit is 0 before doing so.

6.3.10 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 a user option byte (000C2H). Note that the selectable frequencies depend on the values of the FRQSEL[4:3] bits of the user 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 the FRQSEL[2:0] bits of the user option byte (000C2H).

Figure 6 - 13 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H
 After reset: The value set in the FRQSEL[2:0] bits of a user 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			
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f _{IH} = 24 MHz	f _{IH} = 32 MHz	f _{IH} = 24 MHz f _{HOCO} = 48 MHz	f _{IH} = 32 MHz f _{HOCO} = 64 MHz
0	0	1	f _{IH} = 12 MHz	f _{IH} = 16 MHz	f _{IH} = 12 MHz f _{HOCO} = 24 MHz	f _{IH} = 16 MHz f _{HOCO} = 32 MHz
0	1	0	f _{IH} = 6 MHz	f _{IH} = 8 MHz	f _{IH} = 6 MHz f _{HOCO} = 12 MHz	f _{IH} = 8 MHz f _{HOCO} = 16 MHz
0	1	1	f _{IH} = 3 MHz	f _{IH} = 4 MHz	f _{IH} = 3 MHz f _{HOCO} = 6 MHz	f _{IH} = 4 MHz f _{HOCO} = 8 MHz
1	0	0	Setting prohibited	f _{IH} = 2 MHz	Setting prohibited	f _{IH} = 2 MHz f _{HOCO} = 4 MHz
1	0	1	Setting prohibited	f _{IH} = 1 MHz	Setting prohibited	f _{IH} = 1 MHz f _{HOCO} = 2 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.
- Note that even if a value set in the HOCODIV register is the same as its current value, up to three cycles of waiting following the setting are required for stable operation of the CPU/peripheral hardware clock.
- Caution 4.** When using the HOCODIV register to change f_{HOCO} to 64 or 48 MHz (FRQSEL4 = 1 and HOCODIV[2:0] = 000B), clear bit 5 (TKBEN) of peripheral enable register 2 (PER2) before doing so.

6.3.11 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 - 14 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

MOCODIV 1	MOCODIV 0	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.

6.3.12 High-speed system clock division register (MOSCDIV)

The MOSCDIV 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 - 15 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.

6.3.13 High-speed on-chip oscillator trimming register (HIOTRM)

The HIOTRM 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 - 16 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/G23 High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN5555)**.

6.3.14 Middle-speed on-chip oscillator trimming register (MIOTRM)

The MIOTRM 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 - 17 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 42 Electrical Characteristics**.

6.3.15 Low-speed on-chip oscillator trimming register (LIOTRM)

The LIOTRM 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 - 18 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

Low-speed on-chip oscillator

Minimum speed

Initial value

Maximum speed

Remark For details about the accuracy adjustment resolution of the low-speed on-chip oscillator clock, see **Section 42 Electrical Characteristics**.

6.3.16 Standby mode release setting register (WKUPMD)

The WKUPMD 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 - 19 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. ^{Notes 3, 4}

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/L23 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 27 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 42 Electrical Characteristics**.

Note 4. This is only specifiable when the setting of the FRQSEL4 bit of the user option byte (000C2H) is 0.

6.3.17 XT1 oscillator margin checking control register (SOMRG)

The SOMRG register is used to check the margins of oscillation in evaluating oscillation by the XT1 oscillator.

The SOSCMRG1 and SOSCMRG0 bits can be used to make the settings for checking the lower- and upper-limit margins for oscillation by the XT1 oscillator. In evaluating oscillation by the XT1 oscillator and a crystal resonator, checking the lower- and upper-limit margins of oscillation in addition to the normal oscillation improves reliability in evaluating oscillation by the XT1 oscillator.

Figure 6 - 20 Format of XT1 Oscillator Margin Checking Control Register (SOMRG)

Address: F0217H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SOMRG	0	0	0	0	0	0	SOSCMRG1	SOSCMRG0

SOSCMRG1	SOSCMRG0	Setting for checking a margin of the XT1 oscillator
0	0	Normal oscillation state
0	1	The lower-limit margin of oscillation by the XT1 oscillator is checked. Note
1	0	The upper-limit margin of oscillation by the XT1 oscillator is checked. Note
1	1	Setting prohibited

Note This is only specifiable when oscillation by the XT1 oscillator is to be evaluated.

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.

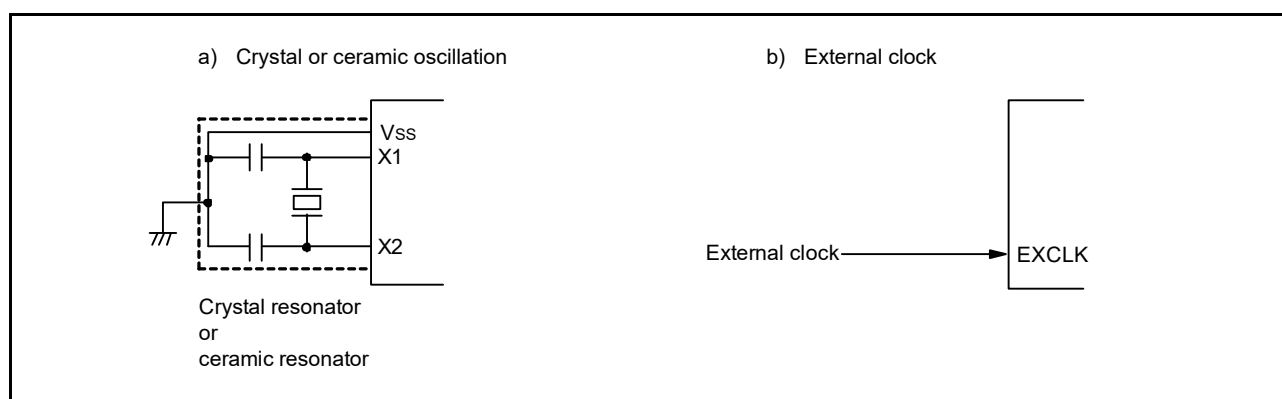
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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 - 1 Connections of Unused Pins**.

Figure 6 - 21 shows examples of external circuits for the X1 oscillator.

Figure 6 - 21 Examples of External Circuits for the X1 Oscillator



Caution When using the X1 oscillator, wire as follows in the area enclosed in broken lines 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.

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.

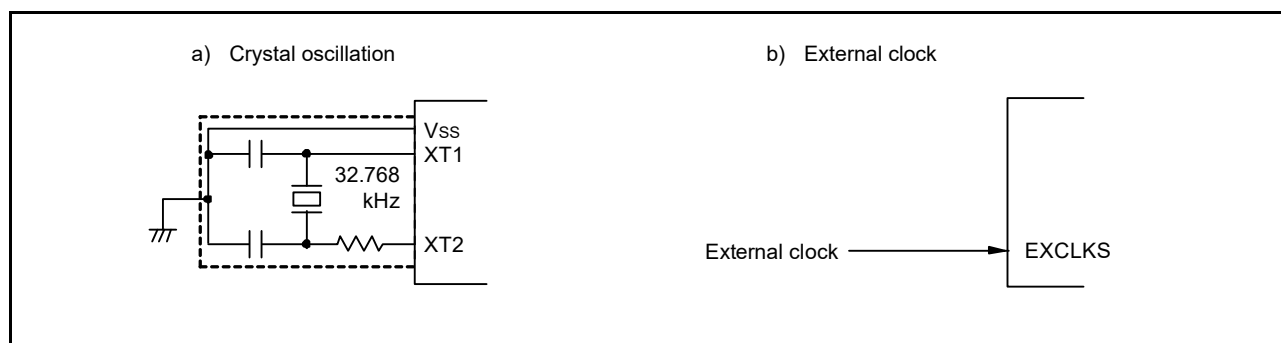
- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

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 - 1 Connections of Unused Pins**.

Figure 6 - 22 shows examples of external circuits for the XT1 oscillator.

Figure 6 - 22 Examples of External Circuits for the XT1 Oscillator



Caution 1. When using the XT1 oscillator, wire as follows in the area enclosed in broken lines to avoid an adverse effect from wiring capacitance.

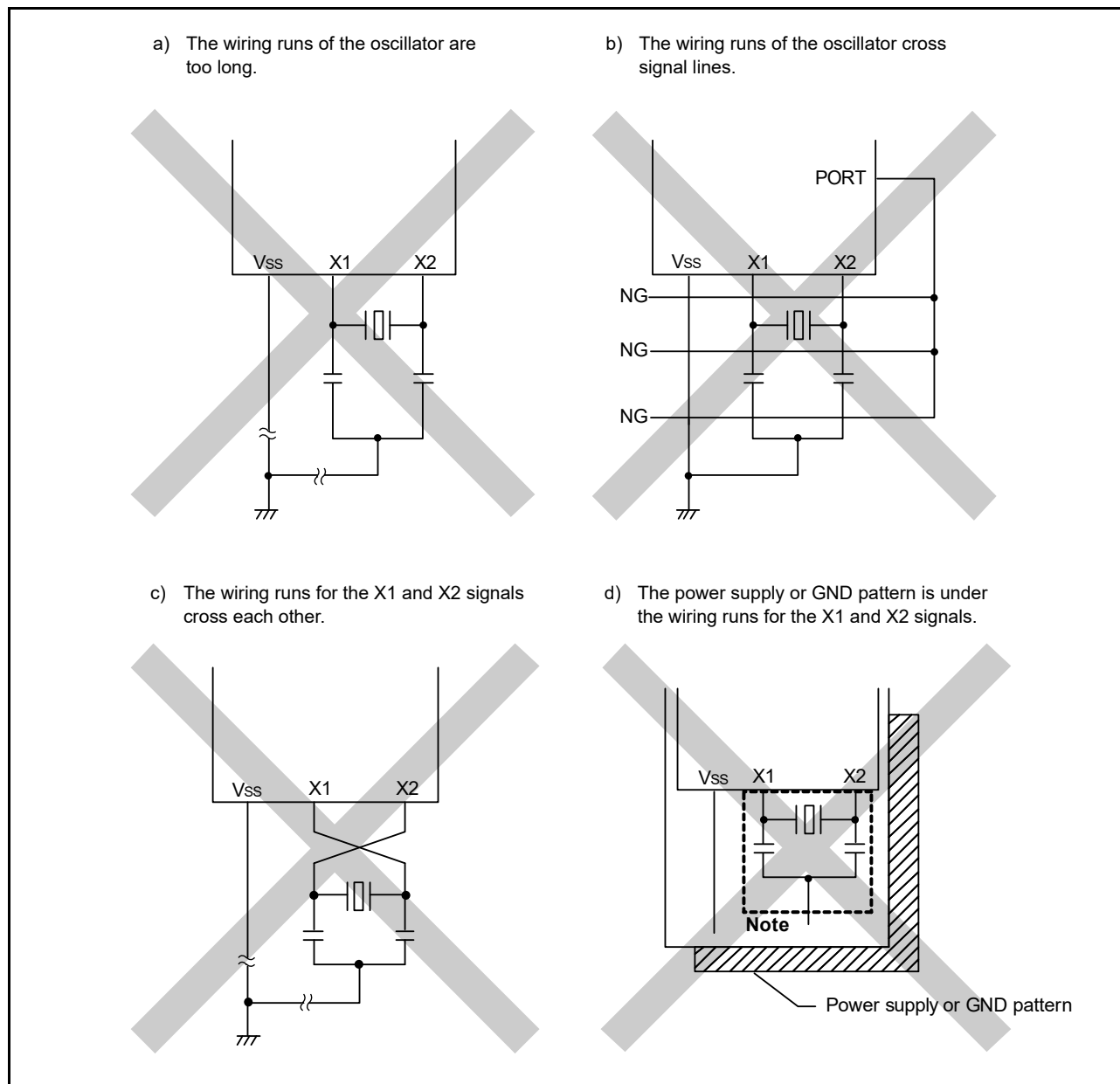
- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a strong fluctuating current flows.
- Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.
- Do not bring a signal line out from the oscillator.

Caution 2. 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 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) 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.
- In evaluating oscillation by the XT1 oscillator, bits 1 and 0 (SOSCMRG1 and SOSCMRG0) in the XT1 oscillator margin checking control register (SOMRG) can be used to make the settings for checking the lower- and upper-limit margins for oscillation by the XT1 oscillator.
- 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 (AMPHS[1:0] = 10B) or low power consumption oscillation 3 (AMPHS[1:0] = 11B) 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 - 23 shows examples of incorrect resonator connection.

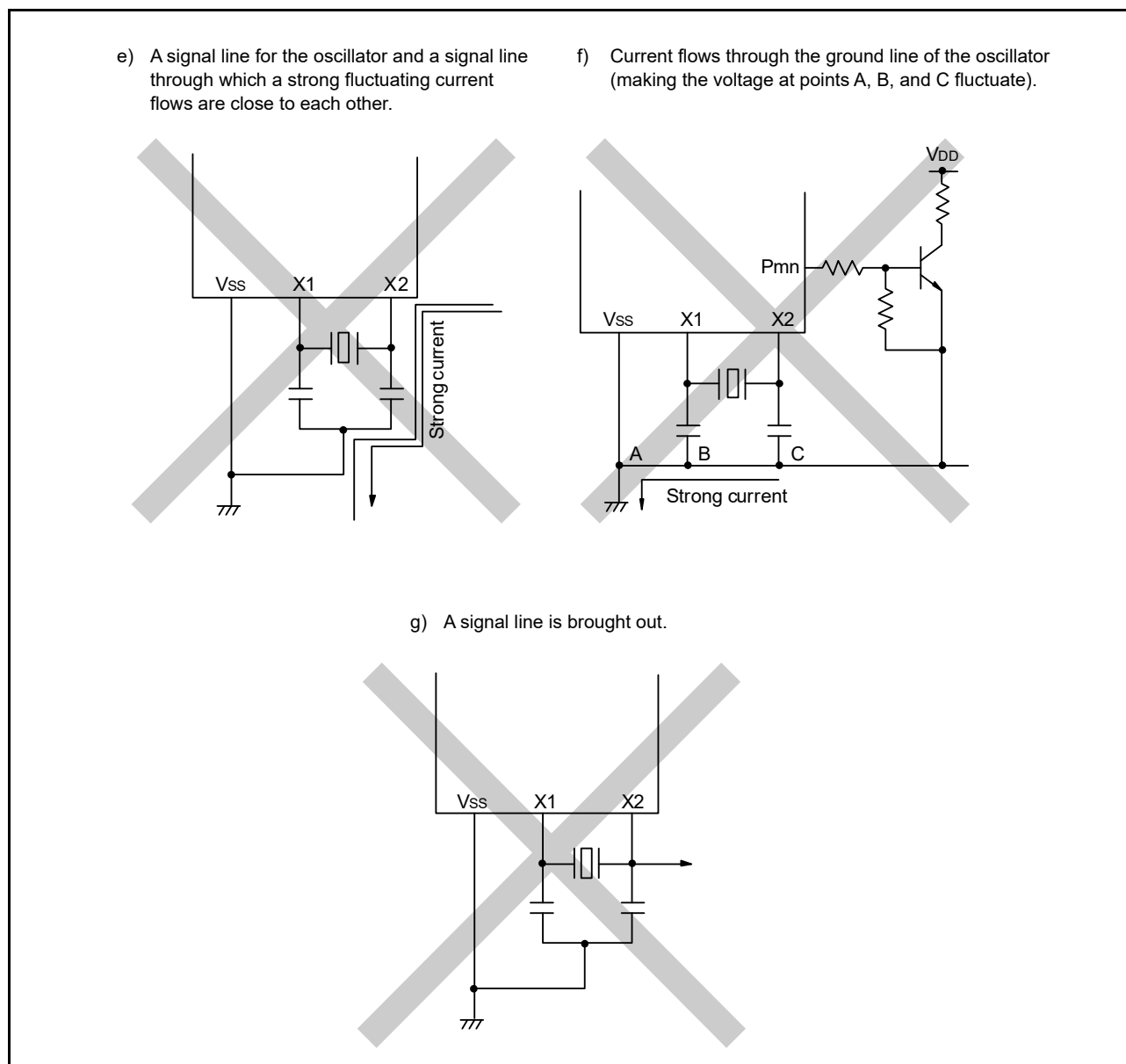
Figure 6 - 23 Examples of Incorrect Resonator Connection (1/2)



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 - 23 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/L23. The frequency can be selected from among 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using a user option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSSTOP) 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/L23. 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/L23.

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.
- f_{IL} 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

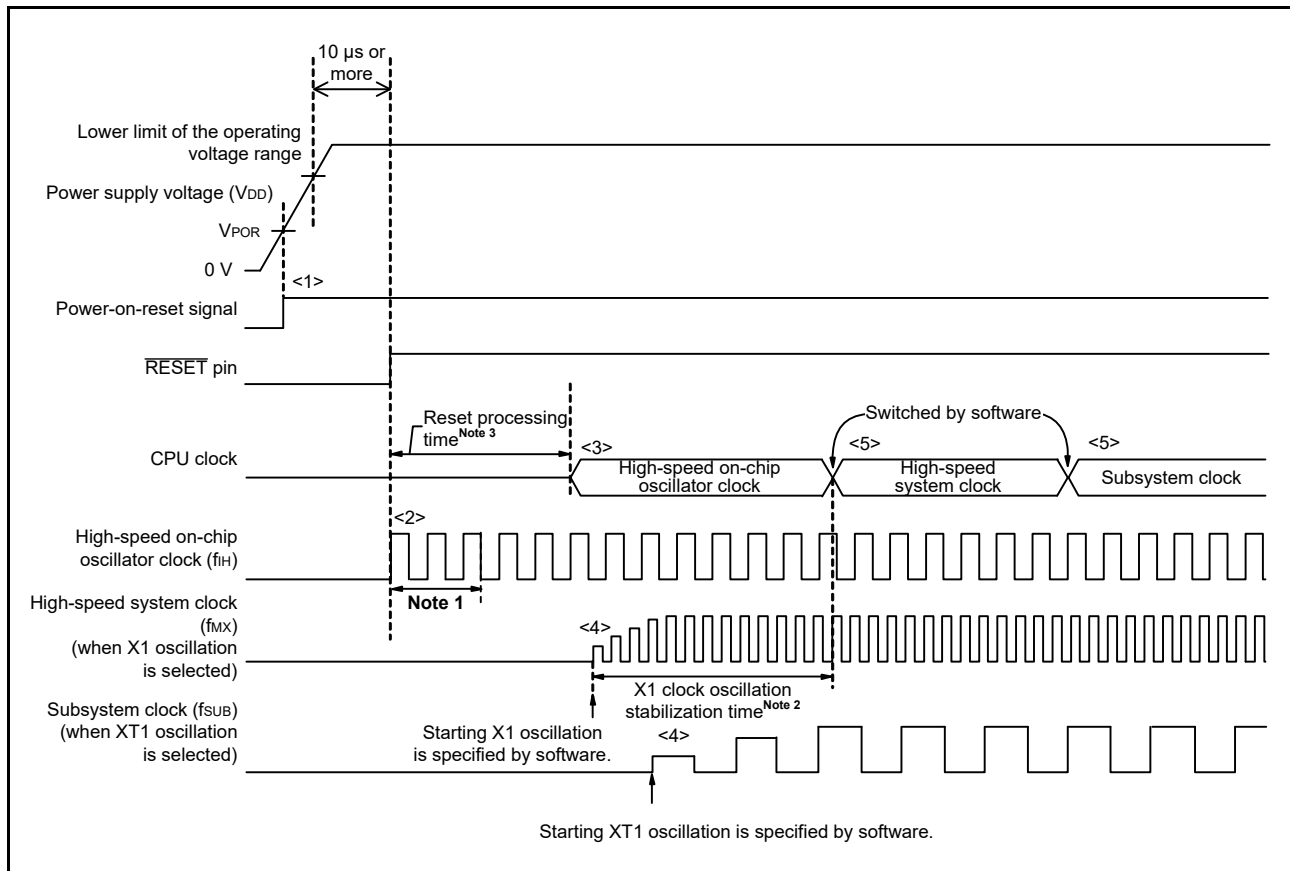
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 1 Block Diagram of Clock Generator**).

- 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_{XS})
 - 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/L23.

Figure 6 - 24 shows the clock generator operation when power supply voltage is turned on.

Figure 6 - 24 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 **42.4 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 29 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 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the FRQSEL[4:0] bits of a user option byte (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[User option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
User option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					fHOCO	fIH
1	1	0	0	0	64 MHz	32 MHz
1	0	0	0	0	48 MHz	24 MHz
0	1	0	0	0	32 MHz	32 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than above					Setting prohibited	

[Settings of the high-speed on-chip oscillator frequency select register (HOCODIV)]

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			
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f _{IH} = 24 MHz	f _{IH} = 32 MHz	f _{IH} = 24 MHz f _{HOCO} = 48 MHz	f _{IH} = 32 MHz f _{HOCO} = 64 MHz
0	0	1	f _{IH} = 12 MHz	f _{IH} = 16 MHz	f _{IH} = 12 MHz f _{HOCO} = 24 MHz	f _{IH} = 16 MHz f _{HOCO} = 32 MHz
0	1	0	f _{IH} = 6 MHz	f _{IH} = 8 MHz	f _{IH} = 6 MHz f _{HOCO} = 12 MHz	f _{IH} = 8 MHz f _{HOCO} = 16 MHz
0	1	1	f _{IH} = 3 MHz	f _{IH} = 4 MHz	f _{IH} = 3 MHz f _{HOCO} = 6 MHz	f _{IH} = 4 MHz f _{HOCO} = 8 MHz
1	0	0	Setting prohibited	f _{IH} = 2 MHz	Setting prohibited	f _{IH} = 2 MHz f _{HOCO} = 4 MHz
1	0	1	Setting prohibited	f _{IH} = 1 MHz	Setting prohibited	f _{IH} = 1 MHz f _{HOCO} = 2 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 $f_x > 10$ MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 0	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 When using the system clock control register (CKC) to change the main system clock (fMAIN), do so while the voltage is within the usable range for the flash operation mode set in the option byte (000C2H) and in the flash operating mode select register (FLMODE) both before and after changing the clock. 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, clock output/buzzer output control function, LCD controller/driver with the subsystem clock (for ultra-low current) in STOP mode or in HALT mode while the CPU is 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 1

- <2> Set the OSCSELS bit of the CMC register to 1 to make the XT1 oscillator operate.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS[1:0] 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 - 25 shows the state transitions of the CPU clock in this product.

Figure 6 - 25 State Transitions of the CPU Clock

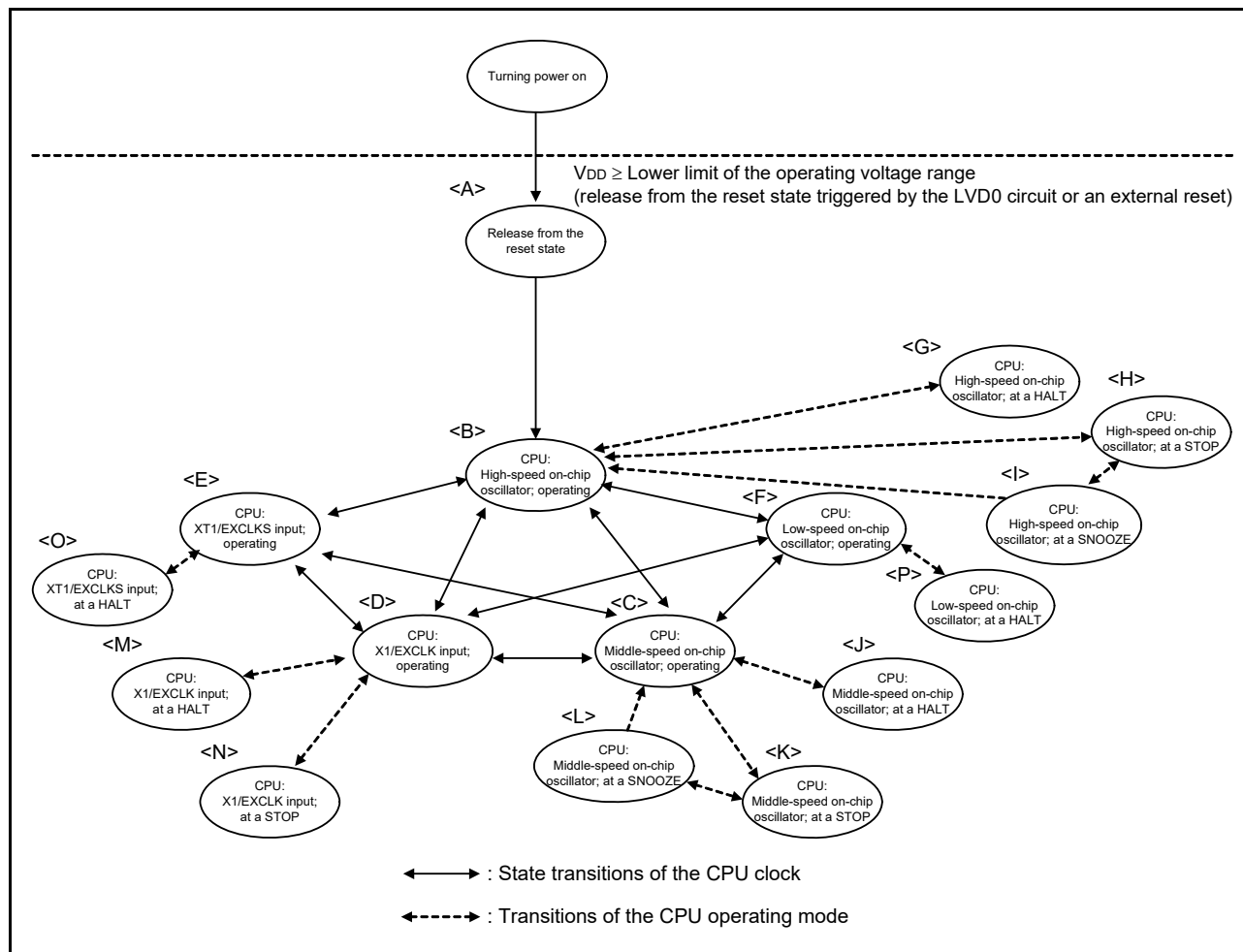


Table 6 - 2 (1/3) to Table 6 - 2 (3/3) show examples of transitions of the CPU clock and SFR settings.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (1/3)

1. Transition to state 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> →

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 where the CPU is operating with the high-speed on-chip oscillator clock

Scope of state transitions: <C> → , <D> → , <E> → , <F> →

(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: → <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 - 25 State Transitions of the CPU Clock.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (2/3)

4. Transitions to state <D> where the CPU is operating with the high-speed system clock

Scope of state transitions: → <D>, <C> → <D>, <E> → <D>, <F> → <D>

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CMC Register ^{Note 1}			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
X1 clock: 1 MHz ≤ f _x ≤ 10 MHz	0	1	0	Note 2	0	Must be checked	0	1
X1 clock: 10 MHz < f _x ≤ 20 MHz	0	1	1	Note 2	0	Must be checked	0	1
External main clock	1	1	x	Note 2	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. 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.

Note 2. Set the oscillation stabilization time as follows.

- Desired oscillation stabilization time counted by the oscillation stabilization time counter status register (OSTC) ≤ 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 42 Electrical Characteristics).

5. Transitions to state <E> where the CPU is operating with the subsystem clock

Scope of state transitions: → <E>, <C> → <E>, <D> → <E>

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CMC Register ^{Note}				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	0	Necessary	1

Unnecessary if these bits are already set

Unnecessary if the subsystem clock is already running

Note 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.

Remark 1. x: Don't care

Remark 2. <A> to <P> in Table 6 - 2 correspond to <A> to <P> in Figure 6 - 25 State Transitions of the CPU Clock.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (3/3)

6. Transitions to state <F> where the CPU is operating with the low-speed on-chip oscillator clock

Scope of state transitions: → <F>, <C> → <F>, <D> → <F>

(Sequence of setting the SFRs) →

SFR Bit to be Set Clock after Transition	CKSEL	Waiting for Oscillation Accuracy Stabilization	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

7. Transitions from the CPU operating mode , <C>, <D>, <E>, or <F> to the HALT mode <G>, <J>, <M>, <O>, or <P>

Scope of state transitions: → <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 , <C>, or <D> to the STOP mode <H>, <K>, or <N>

Scope of state transitions: → <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 27 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 - 25 State Transitions of the CPU Clock.

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 (1/4)

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 • 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	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	
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 • 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	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (2/4)

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	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	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	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	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 (3/4)

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	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	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.	
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	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	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 (4/4)

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 • 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 • MCS = 1	
	XT1 clock	Transition is not possible.	—
	External subsystem clock	Transition is not possible.	

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 $f_{\text{MAIN}} \leftrightarrow f_{\text{SUB}}$

Set Value before Switchover	Set Value after Switchover	
CSS	CSS	
	0 (fCLK = fMAIN)	1 (fCLK = fSUB)
0 (fCLK = fMAIN)	3 cycles	1 + 2 fMAIN/fSUB cycles
1 (fCLK = fSUB)		

Remark 1. The number of clock cycles listed in **Table 6 - 5** to **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** to **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, $f_{\text{IH}} = 8 \text{ MHz}$, $f_{\text{MX}} = 10 \text{ MHz}$)

$$1 + f_{\text{IH}}/f_{\text{MX}} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2 \text{ cycles}$$

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.

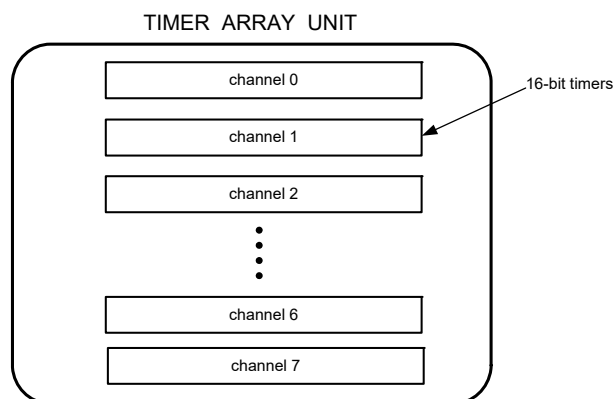
Figure 6 - 26 Examples of External Circuits



Section 7 Timer Array Unit (TAU)

The single timer array unit has 8 channels.

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) - Remote-control output function (refer to 7.9.4)

Note Only channel 0 of unit 0 supports this.

Each of the 16-bit timer channels 1 and 3 of unit 0 can be used as two 8-bit timers, represented by the higher- and lower-order bytes. The following functions can be realized by using channels 1 and 3 as 8-bit timers:

- 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

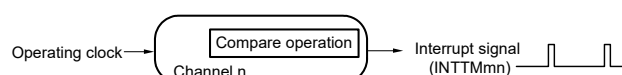
Timer array unit has the following functions.

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.

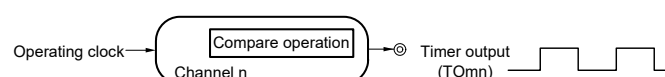
1. Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



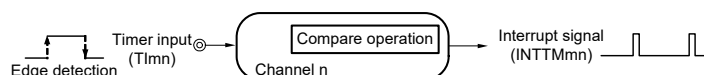
2. Square wave output

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).



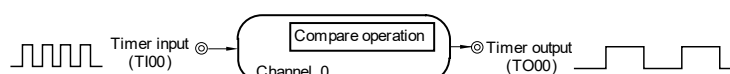
3. External event counter

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.



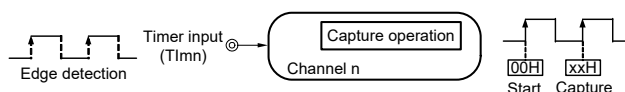
4. Divider function (only channel 0 of unit 0)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



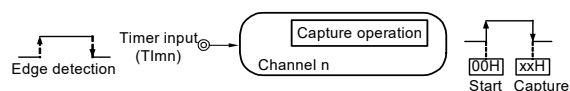
5. Input pulse interval measurement

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.



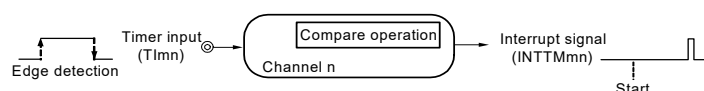
6. Measurement of high-/low-level width of input signal

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)

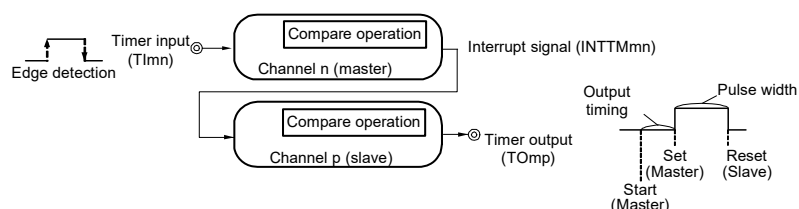
Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

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.

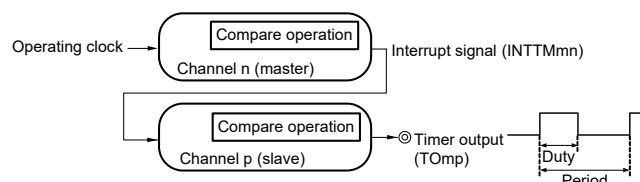
1. One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



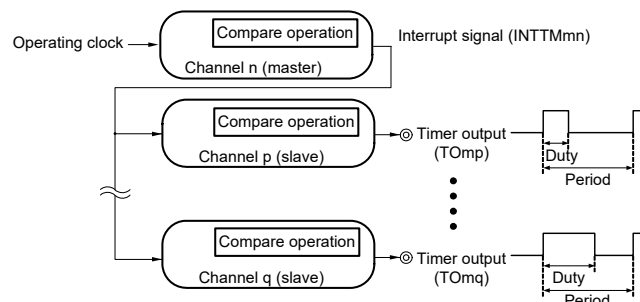
2. PWM (Pulse Width Modulation) output

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



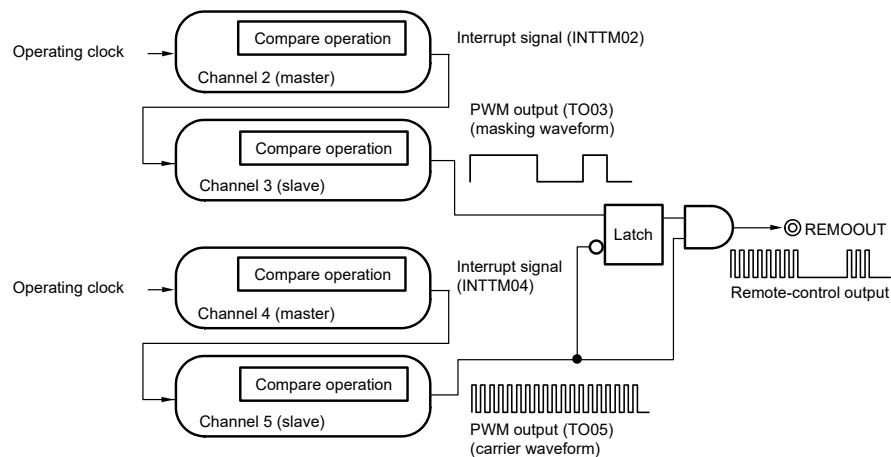
3. Multiple PWM (Pulse Width Modulation) output

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.



4. Remote-control output function

The PWM output function is used with the set of channels 2 and 3 and channels 4 and 5. The PWM signal output by channel 3 is used as the masking waveform, the PWM signal output by channel 5 is used as the carrier waveform, and the logical AND of these signals is output as a remote-control signal.



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 ($n < p < q \leq 7$)

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 7.3.16 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 pins ^{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 select register (TOS) • 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 registers (PMxx)^{Note 2} • Port registers (Pxx)^{Note 2} • Port output mode registers (POMxx)^{Note 2} • Port mode control A registers (PMCAxx)^{Note 2} • Port mode control T registers (PMCTxx)^{Note 2} • Port mode control E registers (PMCExx)^{Note 2} • Port function output enable registers (PFOEx)^{Note 2} • LCD port function registers (PFSEGx)^{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 registers to be set depend 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

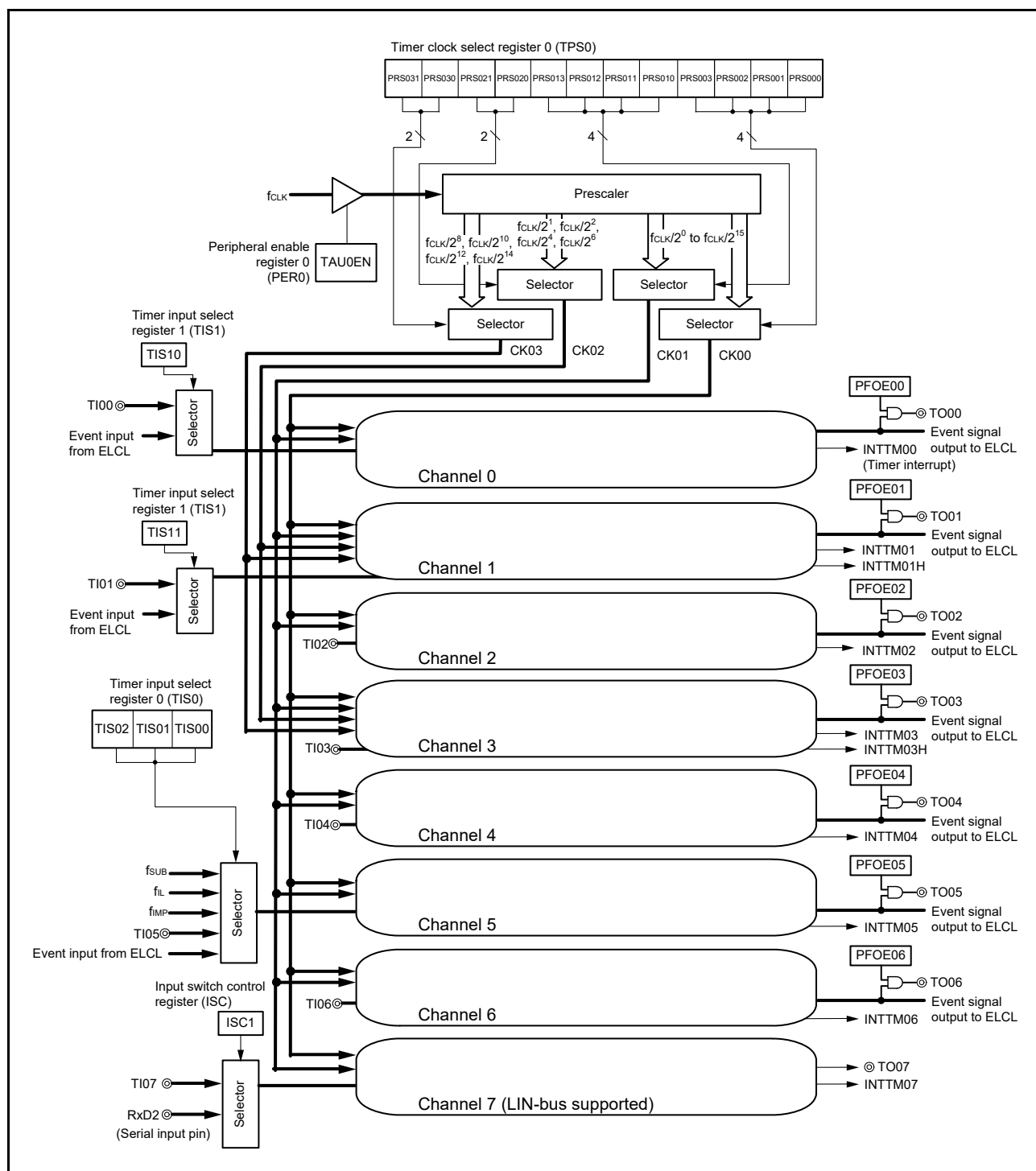
Timer array unit channels		I/O Pins of Each Product					
		44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
Unit 0	Channel 0	(TI00/TO00) Note			TI00/TO00		
	Channel 1	(TI01/TO01) Note					TI01/TO01
	Channel 2	(TI02/TO02) Note	TI02/TO02				
	Channel 3	(TI03/TO03) Note			TI03/TO03		
	Channel 4	(TI04/TO04) Note		TI04/TO04			
	Channel 5	(TI05/TO05) Note			TI05/TO05		
	Channel 6	(TI06/TO06) Note				TI06/TO06	
	Channel 7	(TI07/TO07) Note			TI07/TO07		

Note Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

Remark When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Figure 7 - 1 shows the block diagram of the timer array unit.

Figure 7 - 1 Entire Configuration of Timer Array Unit 0 (Example: 64-pin Products)



Remark fsUB: Subsystem clock frequency
 fL: Low-speed on-chip oscillator clock frequency
 fMP: Middle-speed on-chip oscillator peripheral clock frequency

Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

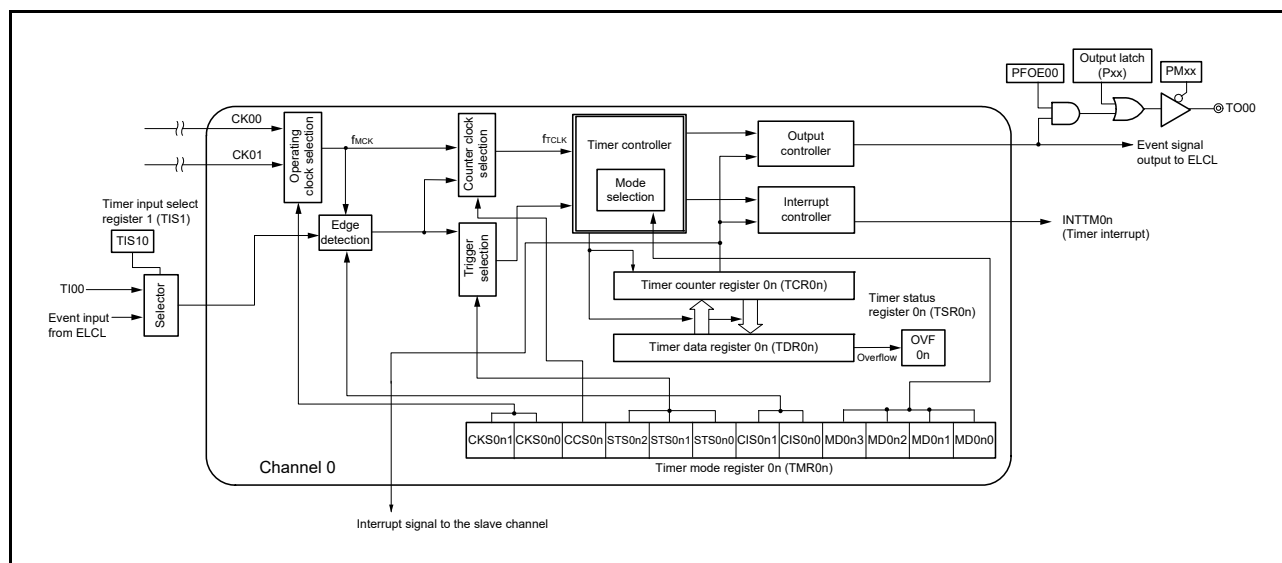


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

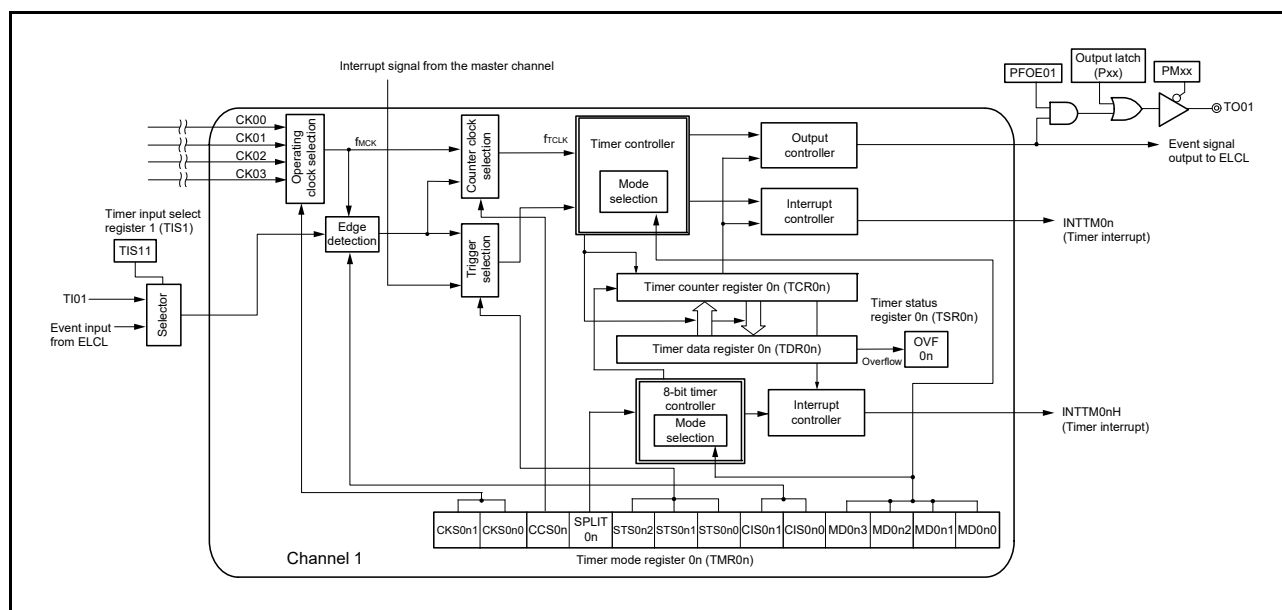
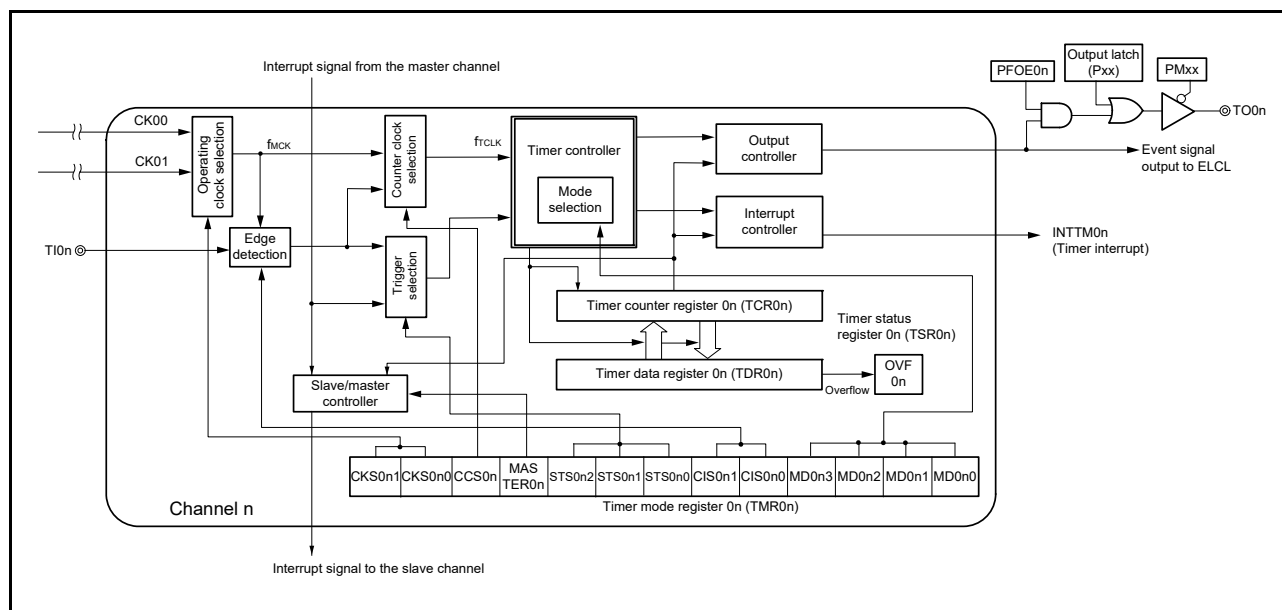


Figure 7 - 4 Internal Block Diagram of Channels 2, 4, and 6 of Timer Array Unit 0



Remark n = 2, 4, 6

Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0

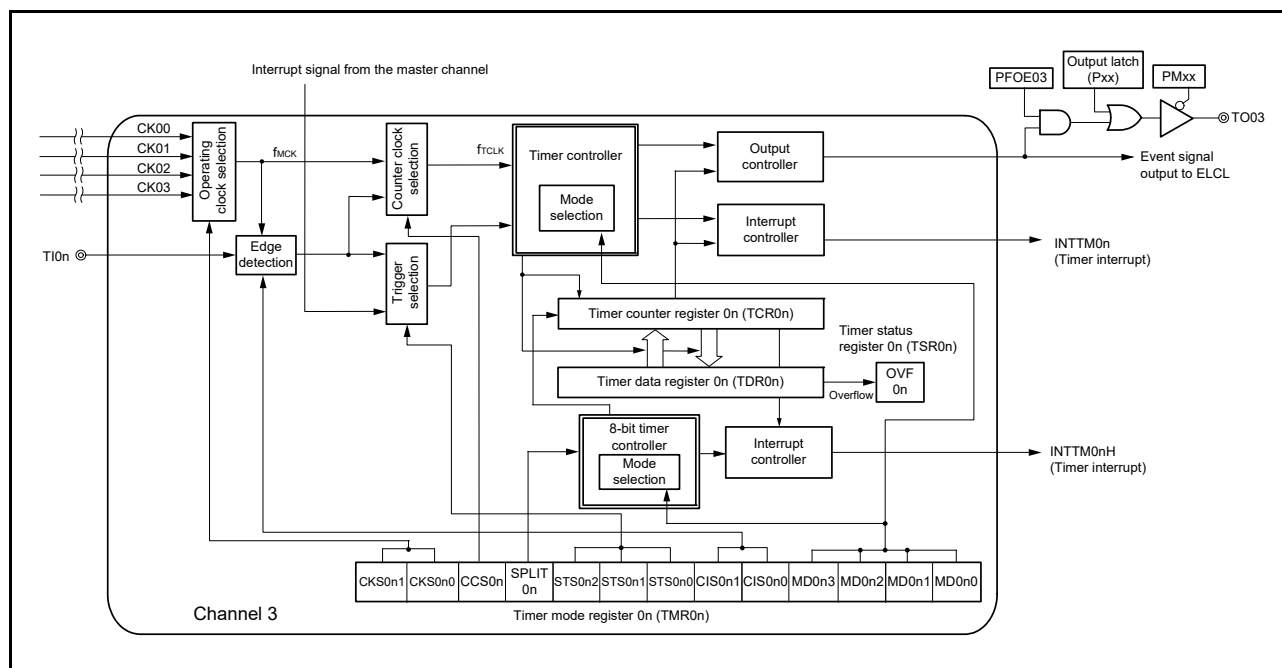


Figure 7 - 6 Internal Block Diagram of Channel 5 of Timer Array Unit 0

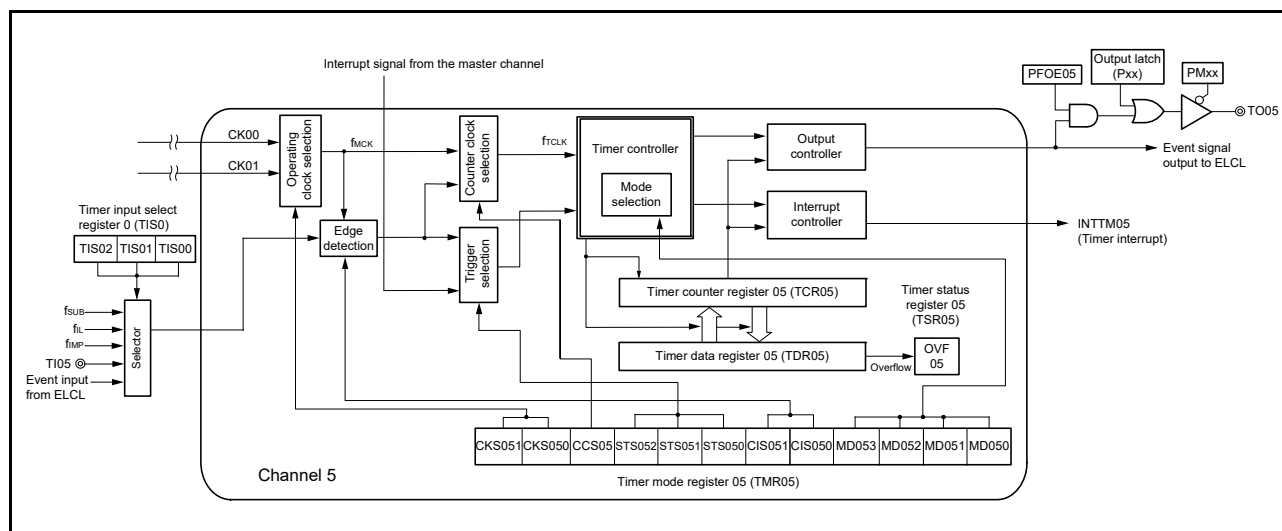
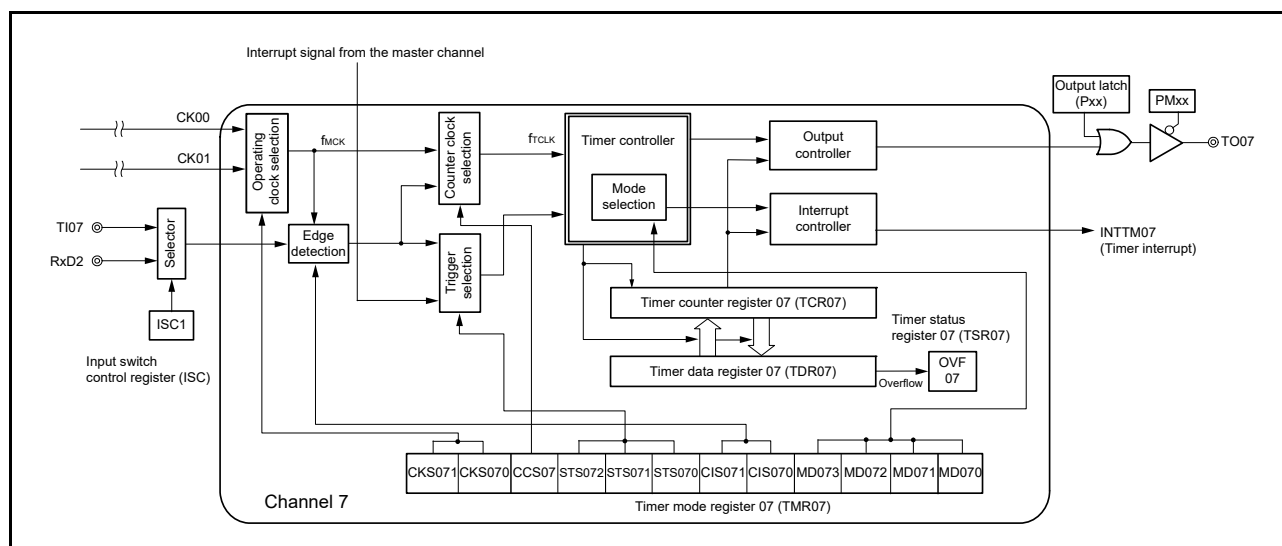


Figure 7 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit 0



7.2.1 Timer counter register mn (TCRmn)

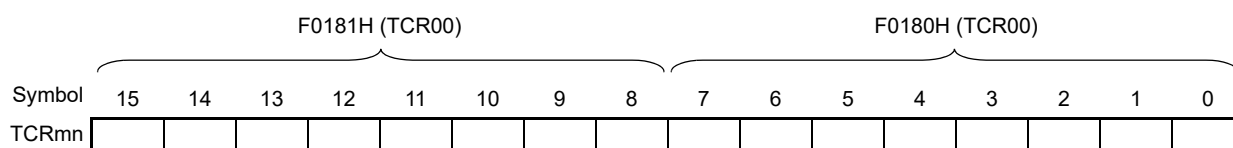
The TCRmn register is a 16-bit read-only register and is used to count the number of input clock cycles. The value of this counter is incremented or decremented in synchronization with the rising edge of each cycle of the counter clock. Whether the counter is incremented or decremented depends on the 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



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 counter 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

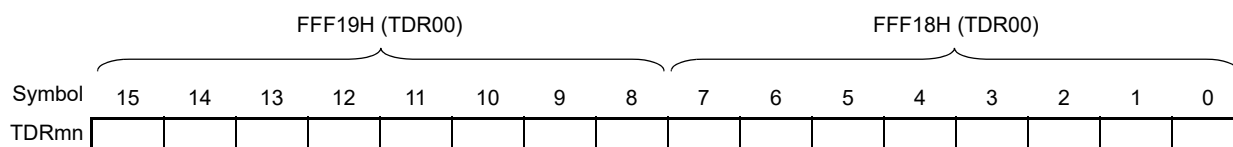
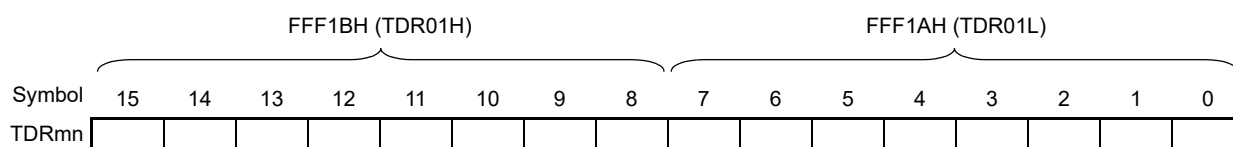


Figure 7 - 10 Format of Timer Data Register mn (TDRmn) (n = 1, 3)

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 (INTTmn) 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 select register (TOS)
- 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 registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- LCD port function registers (PFSEGx)

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	IICA1EN	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 a timer array unit, make sure that the setting of the TAUmEN bit is 1 before setting the following registers. If TAUmEN = 0, the values of the registers which control the timer array unit are cleared to 00H and writing to them is ignored.

- 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)

Note, however, that writing to the following registers is valid.

- Timer input select registers 0 and 1 (TIS0 and TIS1)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- LCD port function registers (PFSEGx)

Caution 2. Be sure to clear bit 1 to 0.

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) to 1 to reset timer array unit 0. 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	IICA1RES	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 Be sure to clear bits 7 and 1 to 0.

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.

Figure 7 - 13 Format of Timer Clock Select Register m (TPSm) (1/2)

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		
PRSmk3	PRSmk2	PRSmk1	PRSmk0		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).

Caution 1. Be sure to clear bits 15, 14, 11, and 10 to 0.

Caution 2. If fCLK (no division) is selected as the operating clock (CKmk) and TDRmn is set to 0000H (m = 0; n = 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 Counter 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

PRS m31	PRS m30	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, and 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 TPSP register, see **7.5.1 Counter clock (fTCLK)**.

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 counter 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	0 ^{Note 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 counter 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 counter 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 and 1, 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
Counter 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)>

MAS TER mn	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)>

SPLI Tmn	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.

STS mn2	STS mn1	STS mn0	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

CIS mn1	CIS mn0	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)	MD mn0	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 read-only and its value is fixed to 0. Writing to this bit is ignored.

Note 2. In one-count mode, interrupt output (INTTmn) when starting a count operation and TCRmn 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 counter clock (fTCLK).

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 TSmL. 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 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn) 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 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn) 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 Operations from the Count Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn) 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 and Remarks are listed on the next page.)

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 (TNFENmn = 1): Four cycles of the operating clock (fMCK)

When the TImn pin noise filter is disabled (TNFENmn = 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 ELCL
0	1	1	Middle-speed on-chip oscillator peripheral clock (fMP)
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 ELCL

Remark n = 0, 1

7.3.11 Timer output select register (TOS)

The TOS register is used to enable the remote-control output function. The remote-control output is generated by the PWM output signal (carrier waveform) generated by channels 4 and 5 being masked by the PWM output signal (masking waveform) generated by channels 2 and 3. The TOS register can only be rewritten before counting starts (TE02, TE03, TE04, TE05 = 0). The TOS 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 - 21 Format of Timer Output Select Register (TOS)

Address: F007AH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TOS	0	0	0	0	0	0	0	TOS0

TOS0	Set the remote-control output
0	Disabled (Output from channels 2 to 5 is timer output.)
1	Enabled (The remote-control signal is output to the REMOOUT pin.)

Caution When the remote-control output is enabled (TOS0 = 1), channels 2 to 5 cannot be used for other functions.

7.3.12 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 - 22 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)

7.3.13 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. If the port pin function that is multiplexed with the input and output of a timer is to be used as a port 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 - 23 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
	Timer output of channel n							
	Timer output value is 0.							
	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)

7.3.14 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 - 24 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)

7.3.15 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 - 25 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.**)

7.3.16 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 - 26 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.

7.3.17 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 (fMCK) 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 (fMCK) 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 the following sections.

- **7.5.1 Counter clock (fTCLK), 2. When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**
- **7.5.2 Timing of the start of counting**
- **7.7 Timer Input (TImn) Control**

Figure 7 - 27 Format of Noise Filter Enable Register 1 (NFEN1)

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.						
	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.

7.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of timers

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the timer array unit.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.14 Port function output enable registers (PFOEx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

7.4 Basic Rules of Timer Array Unit

7.4.1 Basic rules of simultaneous channel operation function

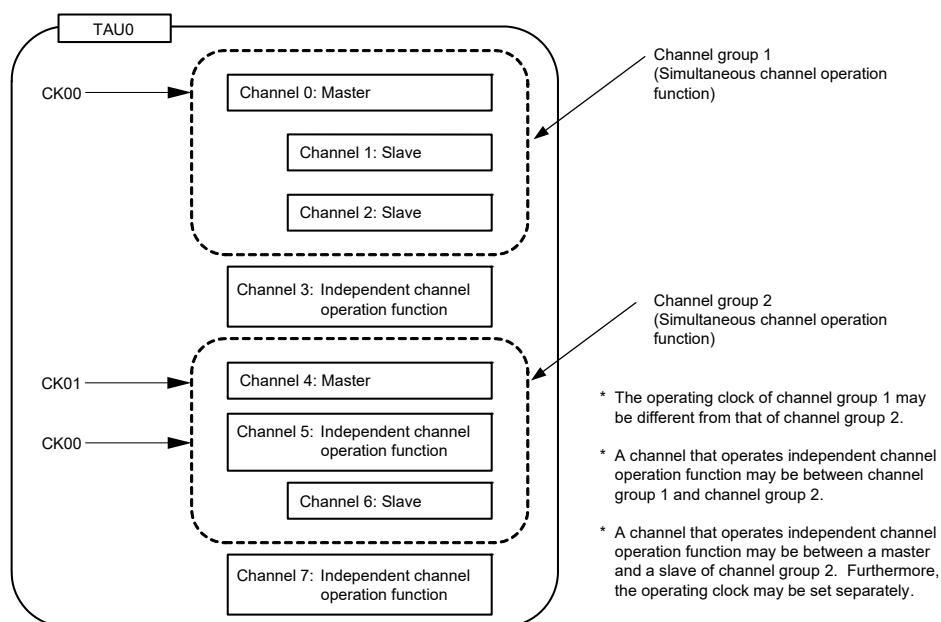
When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

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 counter clock to the lower channels.
8. A slave channel can use INTTMmn (interrupt), a start software trigger, or the counter clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or counter clock to channels with lower channel numbers.
9. A master channel cannot use INTTMmn (interrupt), a start software trigger, or the counter clock from the other higher master channel as a source clock.
10. To simultaneously start channels that operate in combination, the operation enable (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 operation 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 counter 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 Counter clock (f_{CLK})

The counter clock (f_{CLK}) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

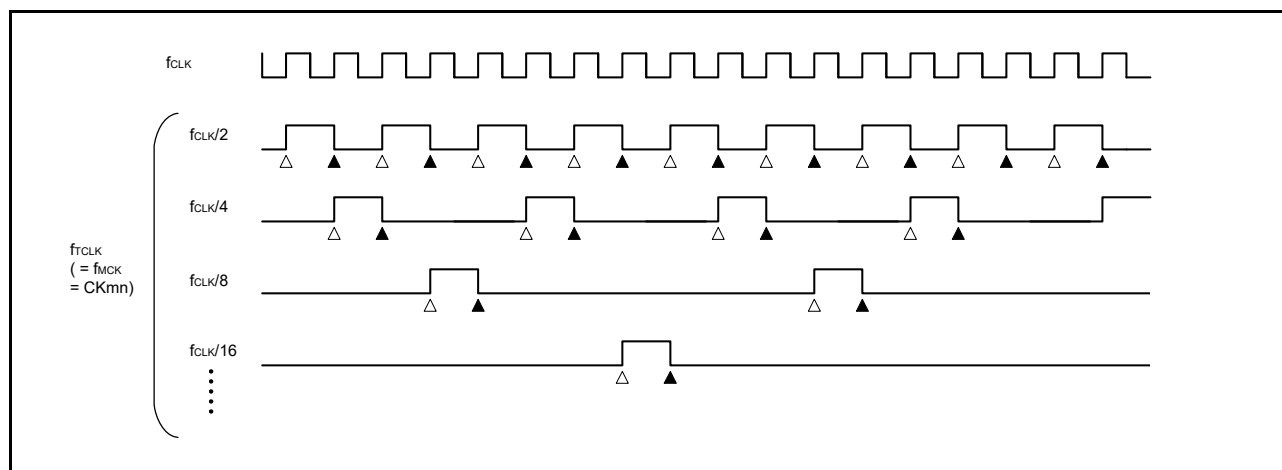
- Operating clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the counter clock (f_{CLK}) are shown below.

1. When operating clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The counter clock (f_{CLK}) 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 counter clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the counter clock”, as a matter of convenience.

Figure 7 - 28 Timing of f_{CLK} and Counter Clock (f_{CLK}) (When CCSmn = 0)

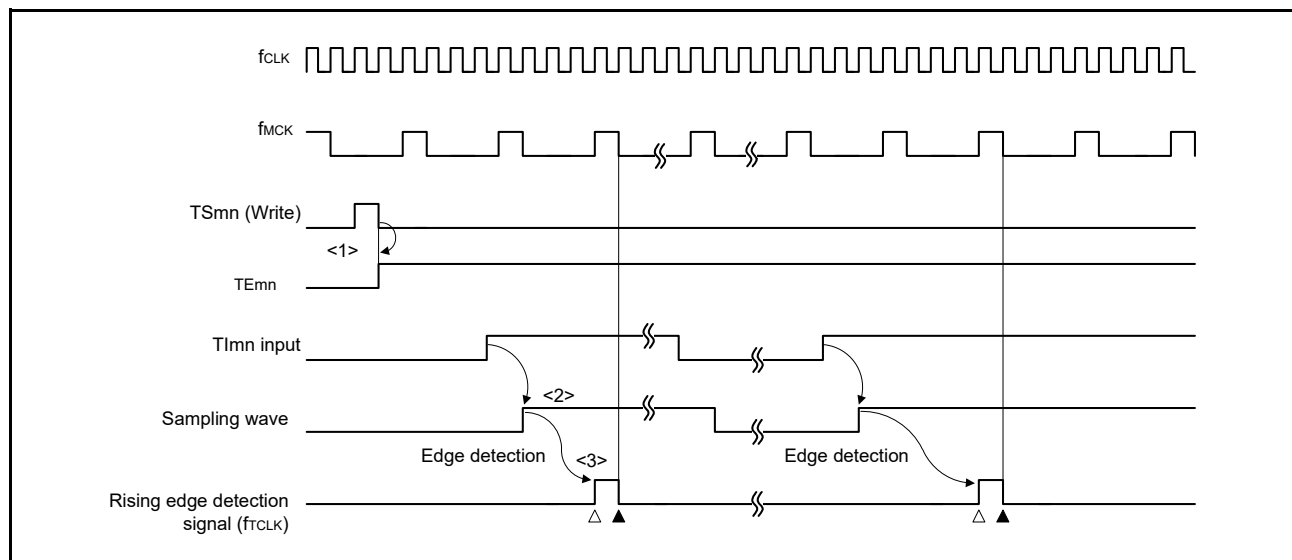


- Remark 1.** Δ : Rising edge of the counter clock
 ▲ : Synchronization, increment/decrement of counter
- Remark 2.** f_{CLK}: CPU/peripheral hardware clock

2. When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The counter clock (fCLK) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising fMCK. The counter clock (fCLK) is delayed for 1 to 2 period of fMCK 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 fCLK from rising edge of the counter clock, because of synchronization with fCLK. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 7 - 29 Timing of fCLK and Counter Clock (fTCLK) (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 fMCK.

<3> The edge is detected by the rising of the sampled signal and the detection signal (counter clock) is output.

Remark 1. Δ : Rising edge of the counter clock
 $\blacktriangle$: Synchronization, increment/decrement of counter

Remark 2. fCLK: CPU/peripheral hardware clock
 fMCK: 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 counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see 7.5.3 Operations of Counters, 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 counter clock performs count down operation (see 7.5.3 Operations of Counters, 2. Operation in event counter mode).
• Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see 7.5.3 Operations of Counters, 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 counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see 7.5.3 Operations of Counters, 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 counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see 7.5.3 Operations of Counters, 5. Operation in capture & one-count mode (high-level width measurement)).

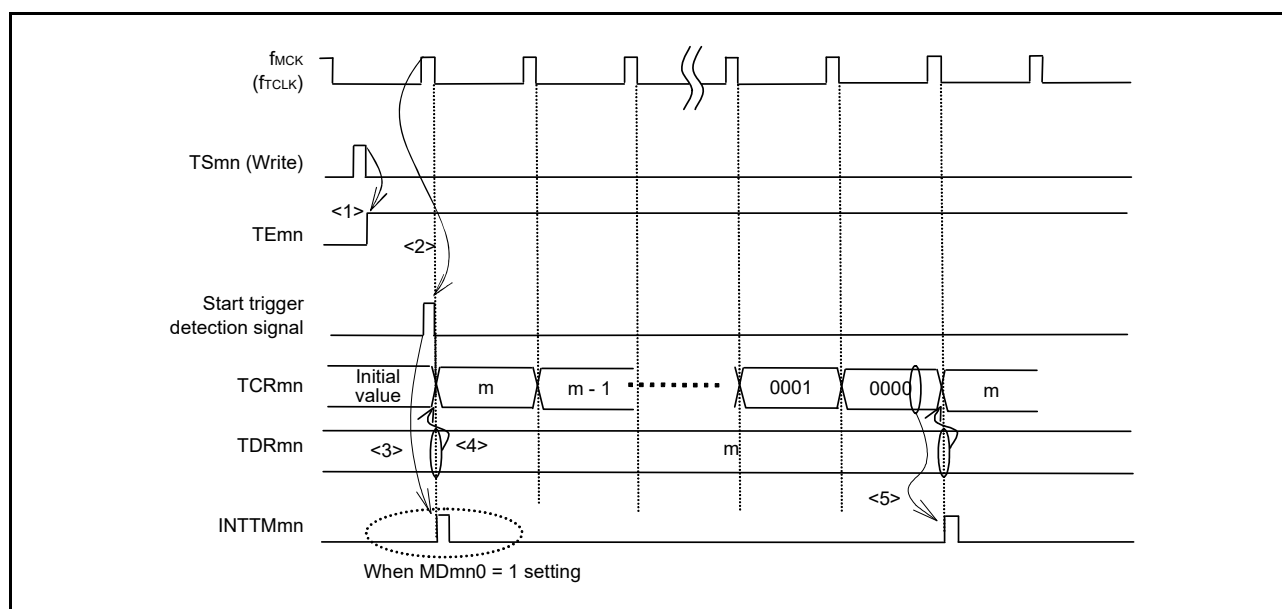
7.5.3 Operations of Counters

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 counter clock generation.
- <2> A start trigger is generated at the first counter 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 counter 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 - 30 Timing during Operation in Interval Timer Mode



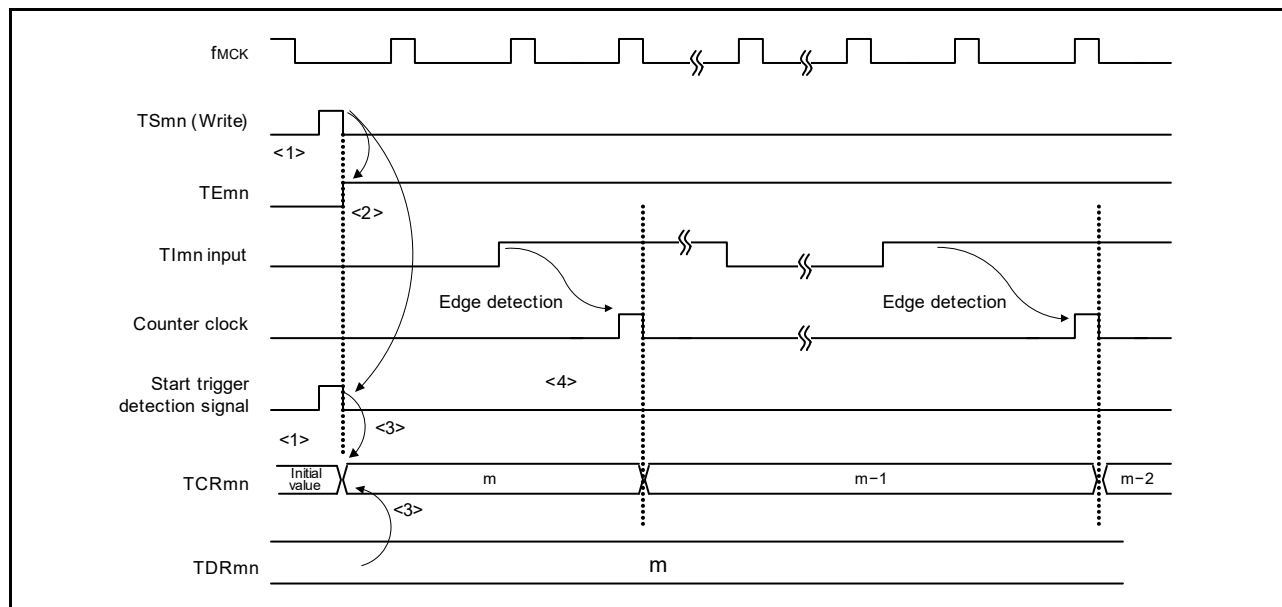
Caution In the first cycle operation of counter clock after writing the TS_{mn} bit, an error at a maximum of one clock is generated since count start delays until counter clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by 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 (TEmn = 0).
- <2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
- <3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn 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 counter clock of the valid edge of the TImn input.

Figure 7 - 31 Timing during Operation in Event Counter Mode

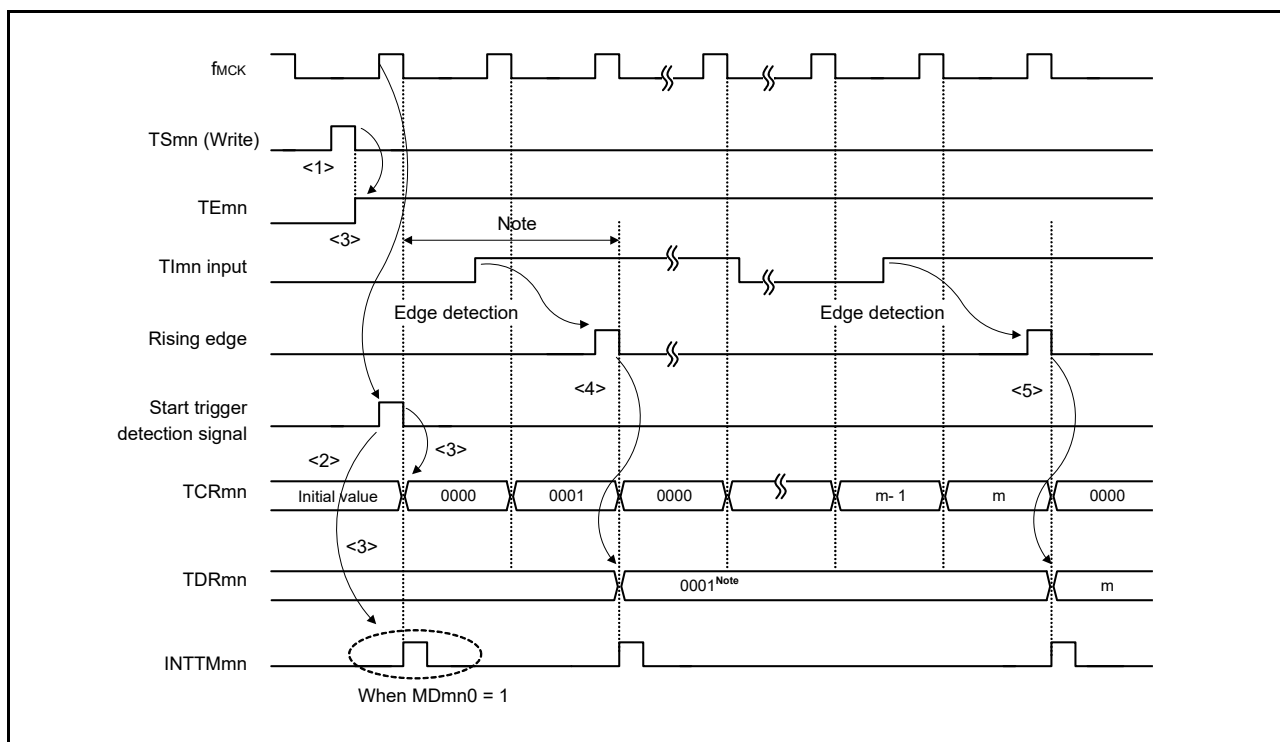


Remark Figure 7 - 31 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the counter clock (fMCK).

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 counter clock generation.
- <3> A start trigger is generated at the first counter 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 - 32 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.

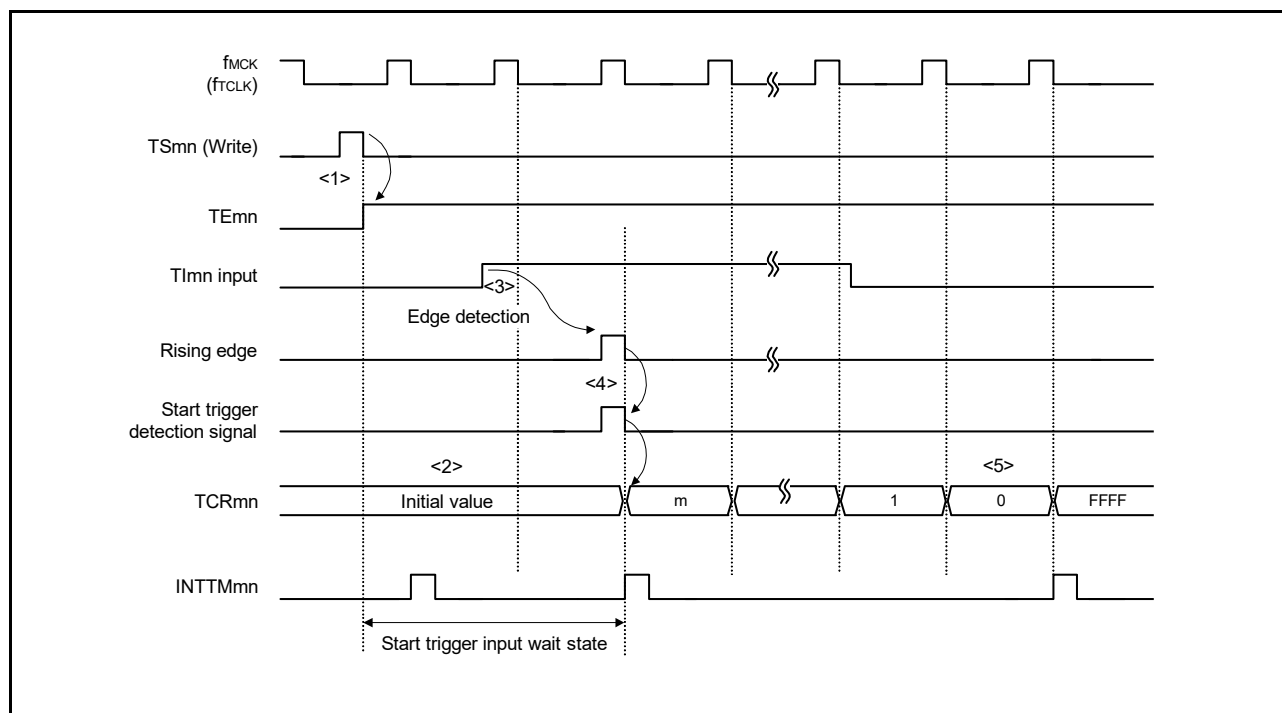
Caution In the first cycle operation of counter clock after writing the TS_{mn} bit, an error at a maximum of one clock is generated since count start delays until counter clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting $MD_{mn0} = 1$.

Remark Figure 7 - 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 counter 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, $INTTM_{mn}$ is generated and the value of the TCR_{mn} register becomes FFFFH and counting stops.

Figure 7 - 33 Timing during Operation in One-count Mode

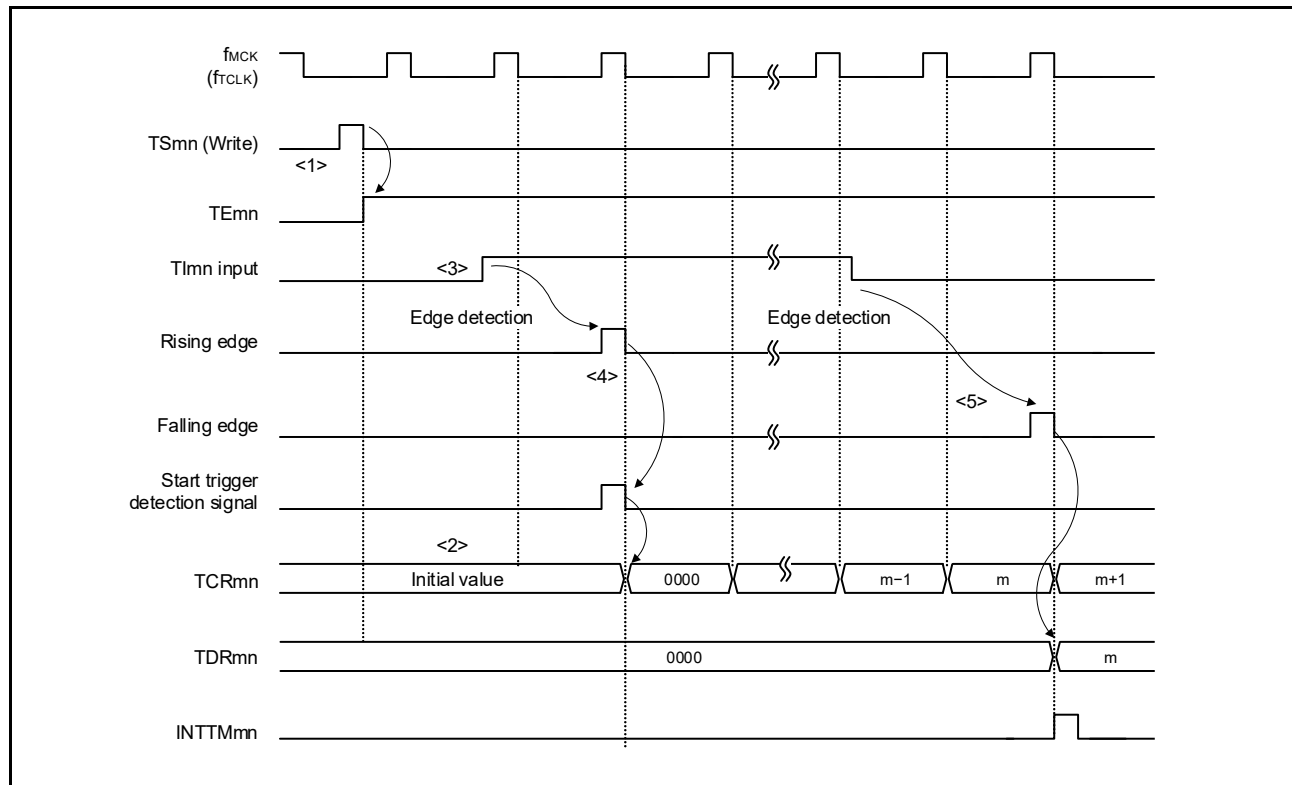


Remark Figure 7 - 33 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the counter 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 - 34 Timing during Operation in Capture & One-count Mode (High-level Width Measurement)

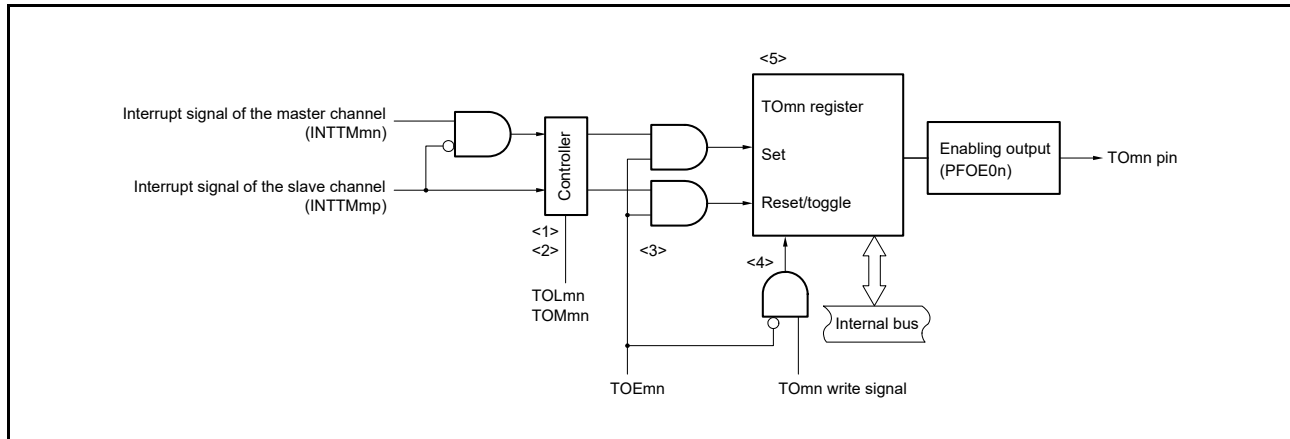


Remark Figure 7 - 34 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI_{mn} input. The error per one period occurs be the asynchronous between the period of the TI_{mn} input and that of the counter clock (f_{MCK}).

7.6 Channel Output (TOMn Pin) Control

7.6.1 TOMn pin output circuit configuration

Figure 7 - 35 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.

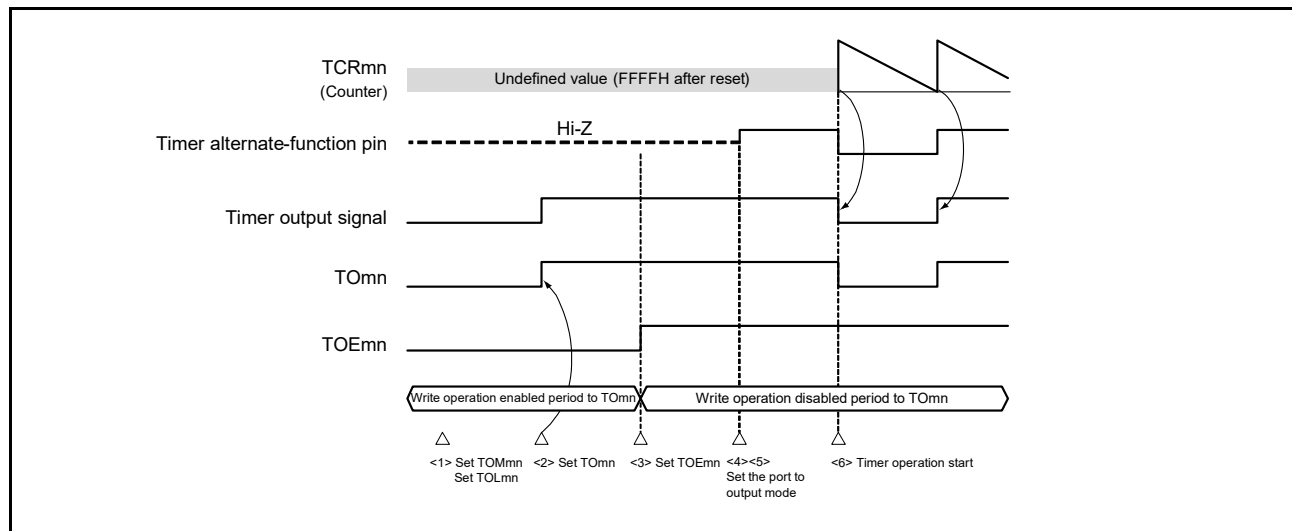
- <3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOM register. Writing to the TOM register (TOMn write signal) becomes invalid. When TOEmn = 1, the TOMn pin output never changes with signals other 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 - 36 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. See **7.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of timers.**

<5> The port I/O setting is set to output. See **7.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of timers.**

<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 Timer Input (TImn) Control** and **7.8 Independent Channel Operation Function of Timer Array Unit**. 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)

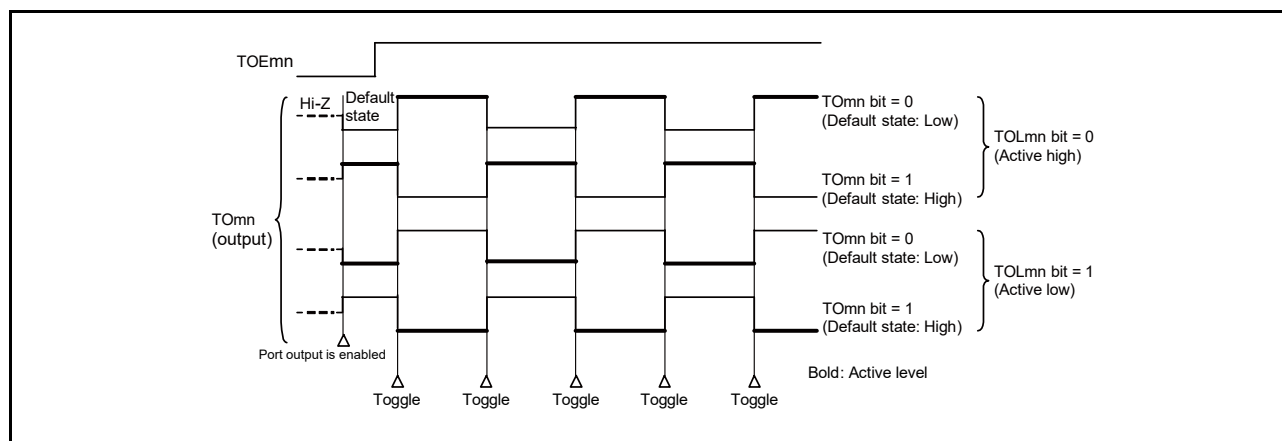
2. Default level of TOMn pin and output level after timer operation start

The change in the output level of the TOMn pin when timer output register m (TOM) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOMn pin is inverted.

Figure 7 - 37 TOMn Pin Output States with Toggled Output (TOMmn = 0)



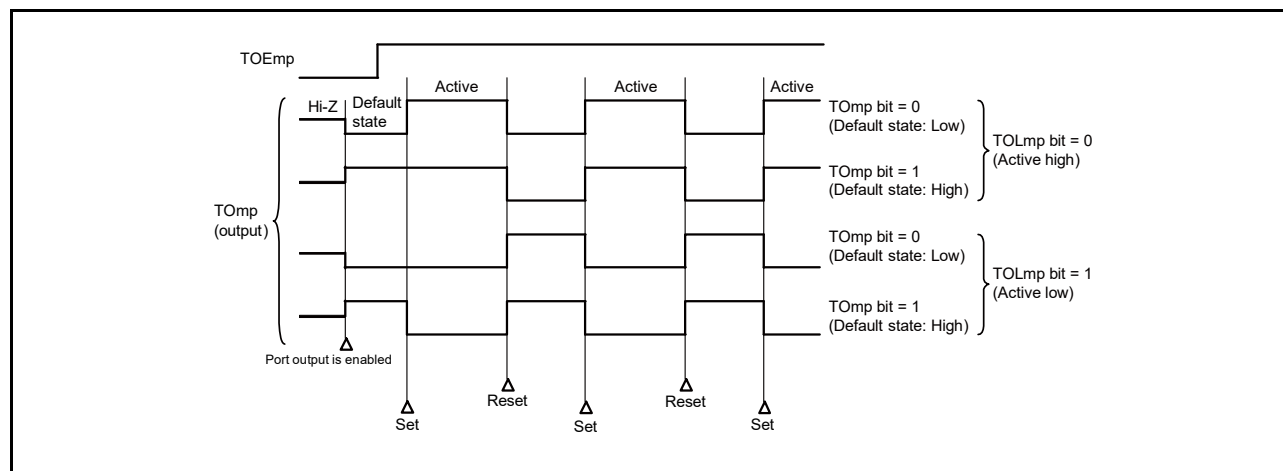
Remark 1. Toggle: Toggle signal to invert the output on the TOMn pin

Remark 2. 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 - 38 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.

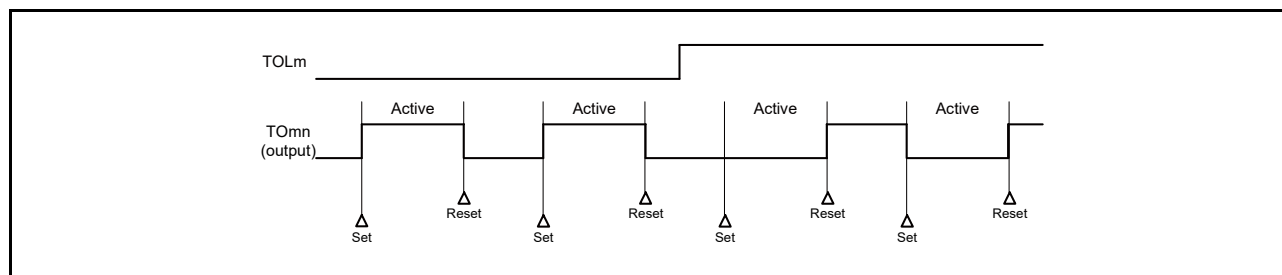
Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

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 - 39 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

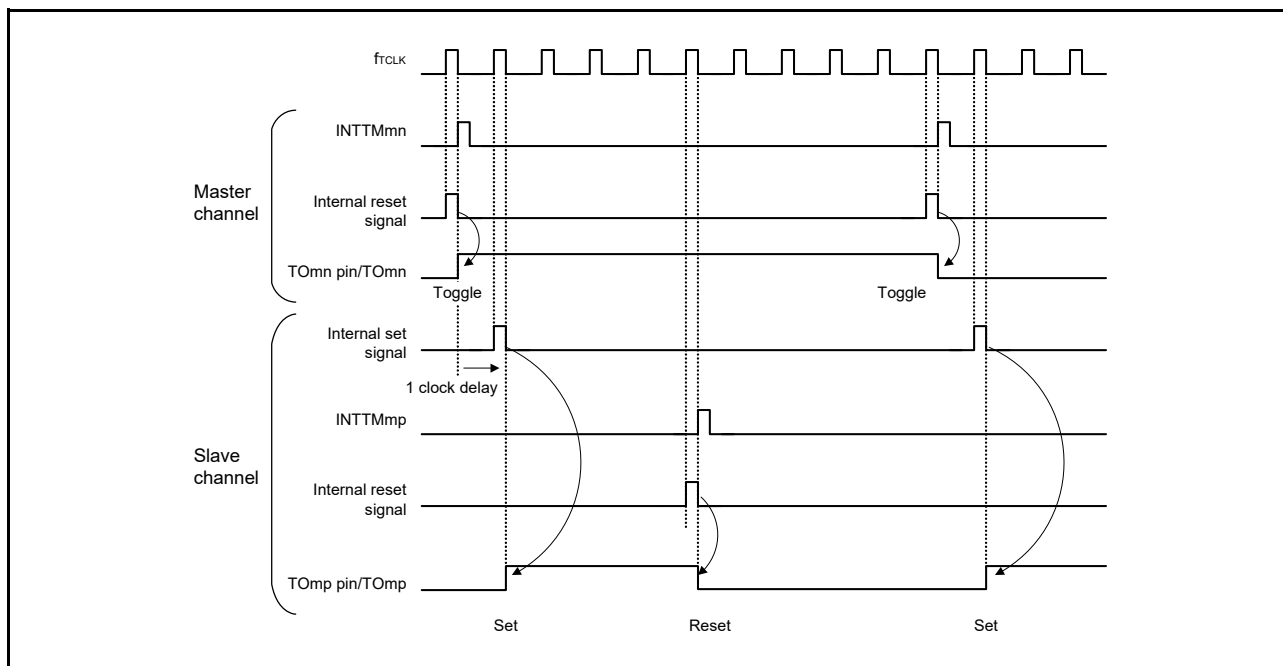
To realize 0%/100% output at PWM output, the TOMn pin/TOMn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 counter clock by the slave channel. If the set condition and reset condition are generated at the same time, a higher priority is given to the latter. **Figure 7 - 40** 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

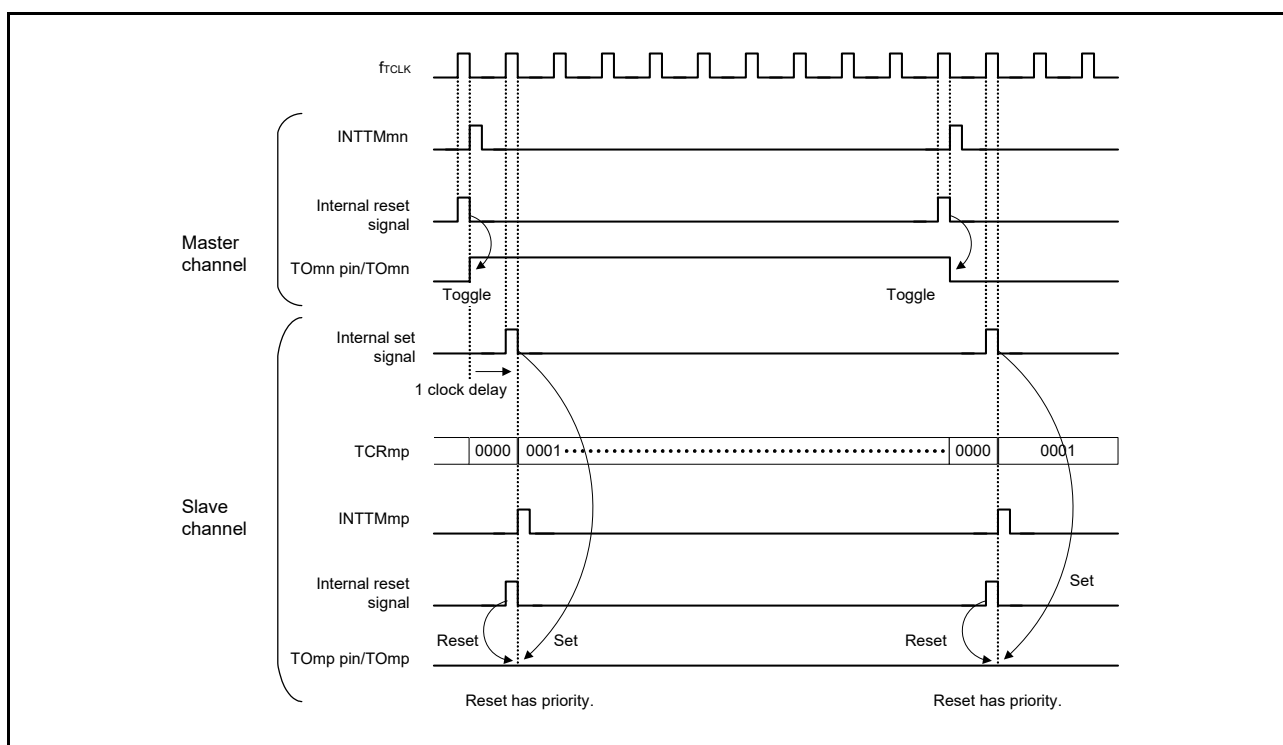
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 7 - 40 States of Operation following Set and Reset Signals

1. Basic timing during operation



2. Timing during operation with the duty cycle set to 0%



(Remarks are listed on the next page.)

Remark 1. Internal reset signal: TOn pin reset/toggle signal
Internal set signal: TOn 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 ≤ 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 - 41 Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	0	0	0	0	0	0	TO07	0	TO06	0	TO05	1	TO04	0	TO03	0	TO02	0	TO01	1	TO00	0
TOE0	0	0	0	0	0	0	0	TOE07	0	TOE06	0	TOE05	1	TOE04	0	TOE03	1	TOE02	1	TOE01	1	TOE00	1

Data to be written

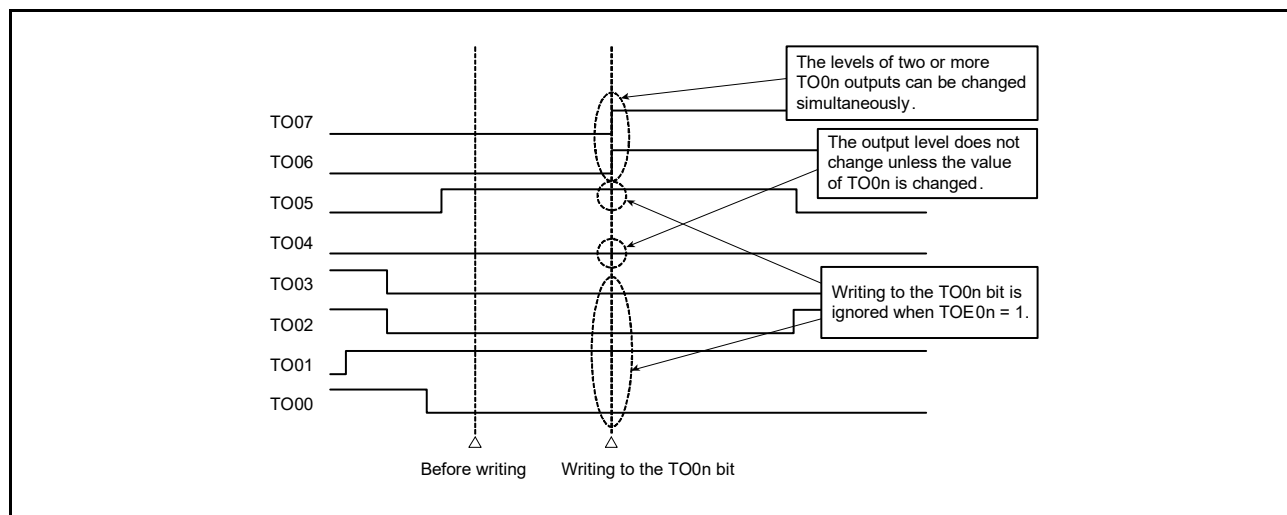
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
								↓	↓	×	↓	×	×	×	×	×	×

After writing

TO0	0	0	0	0	0	0	0	TO07	1	TO06	1	TO05	1	TO04	0	TO03	0	TO02	0	TO01	1	TO00	0

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 - 42 TO0n Pin States by Collective Manipulation of TO0n Bits

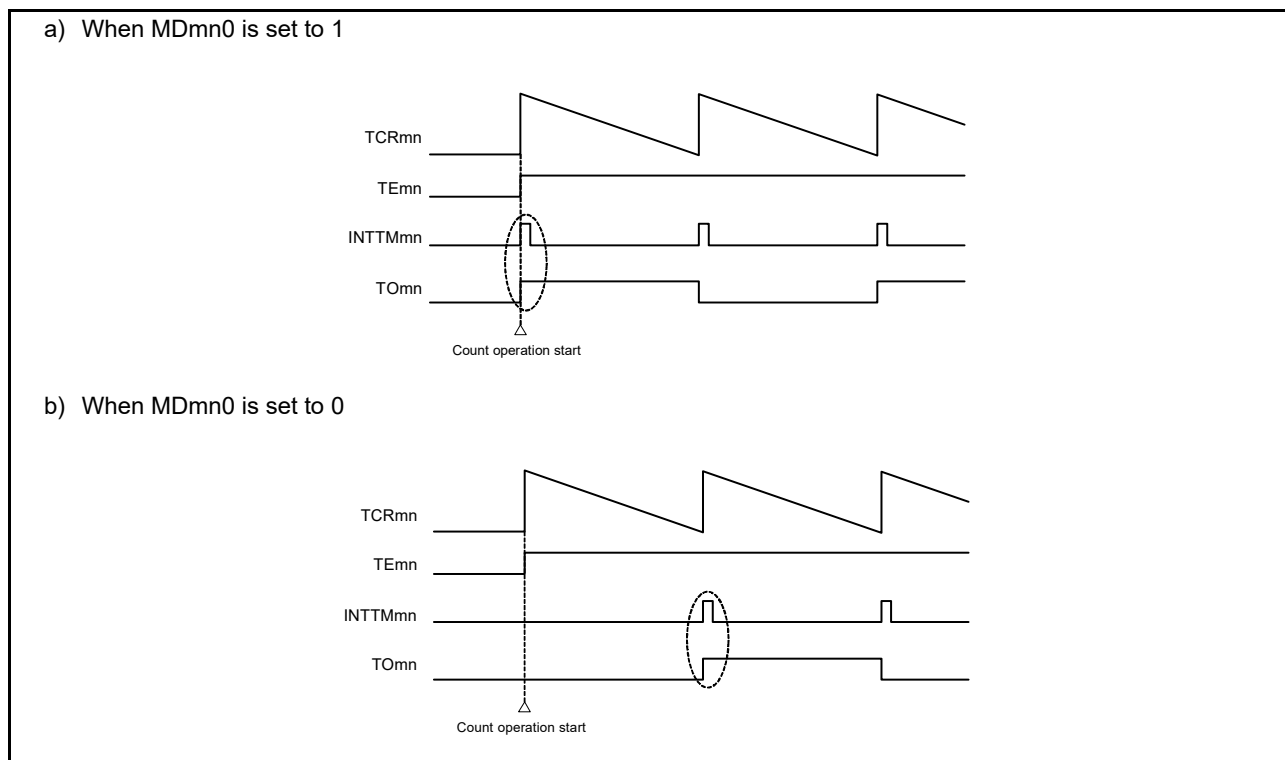


Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 - 43** shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 7 - 43 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.

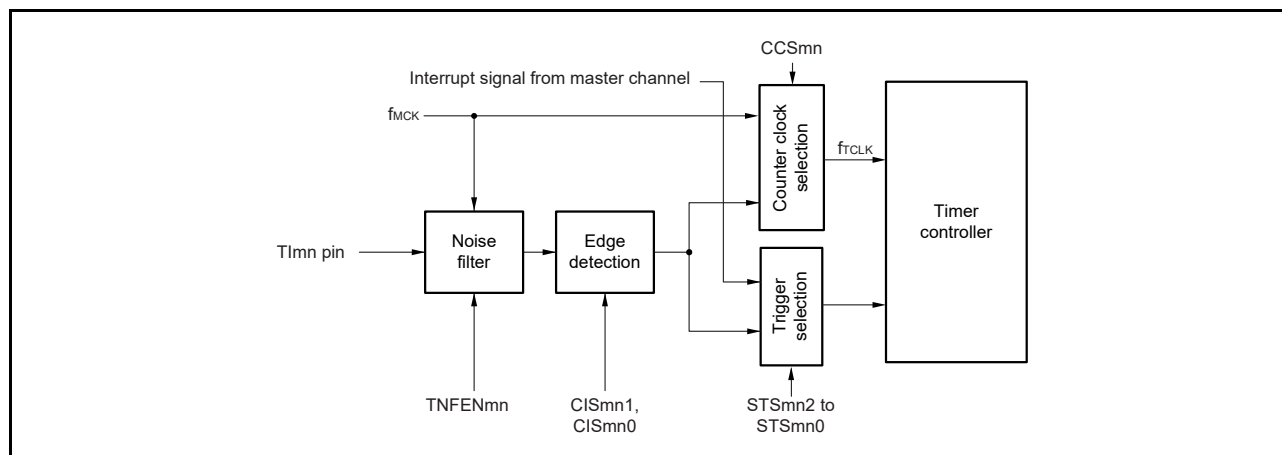
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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.

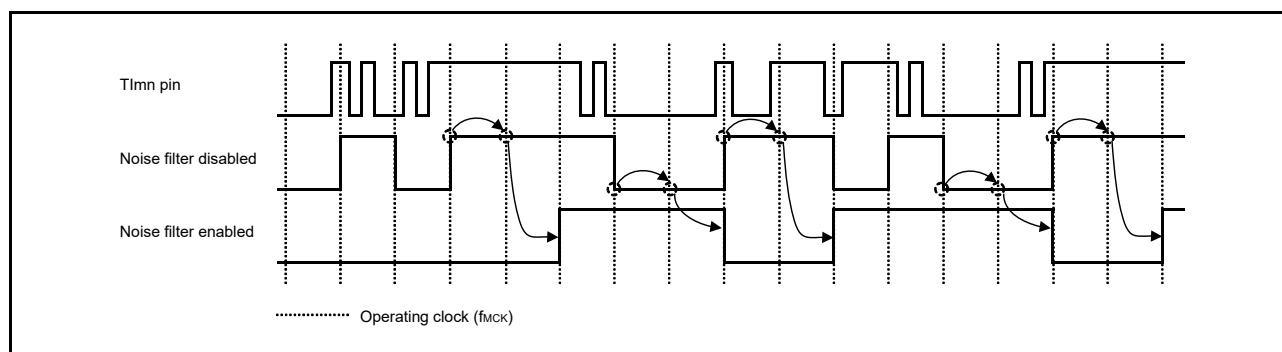
Figure 7 - 44 Input Circuit Configuration



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 - 45 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.

$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of counter clock} \times (\text{Set value of TDRmn} + 1)$

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.

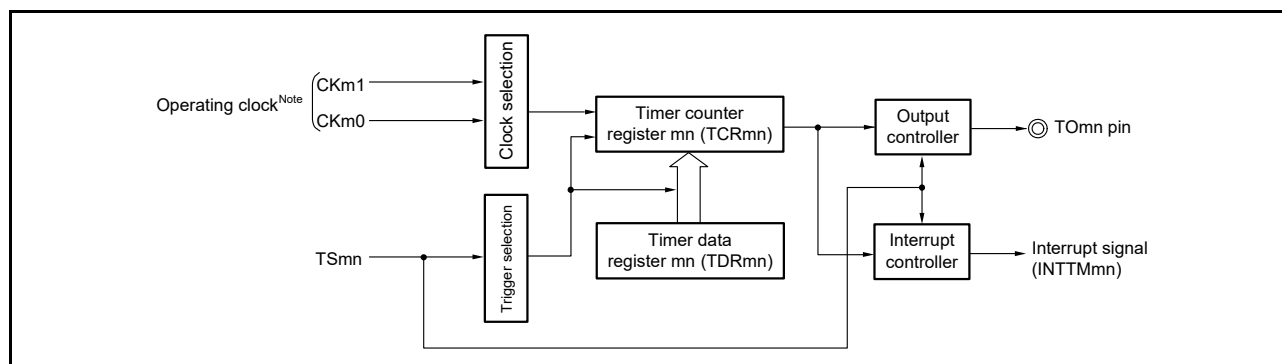
$\bullet \text{ Period of square wave output from TOmn} = \text{Period of counter clock} \times (\text{Set value of TDRmn} + 1) \times 2$

$\bullet \text{ Frequency of square wave output from TOmn} = \text{Frequency of counter clock} / \{(\text{Set value of TDRmn} + 1) \times 2\}$

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 counter clock after the operation enable (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 counter clock. When TCRmn = 0000H, INTTMmn is output and the output on TOmn is toggled at the next counter 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.

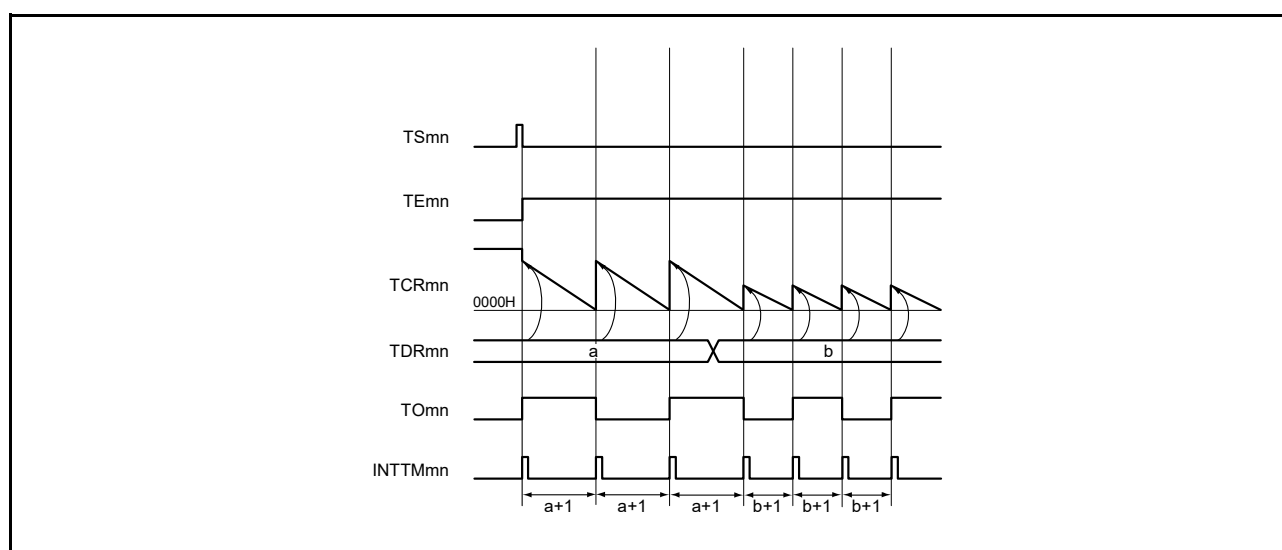
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 46 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 - 47 Example of Basic Timing during Operation as an Interval Timer or for Square Wave Output (MDmn0 = 1)

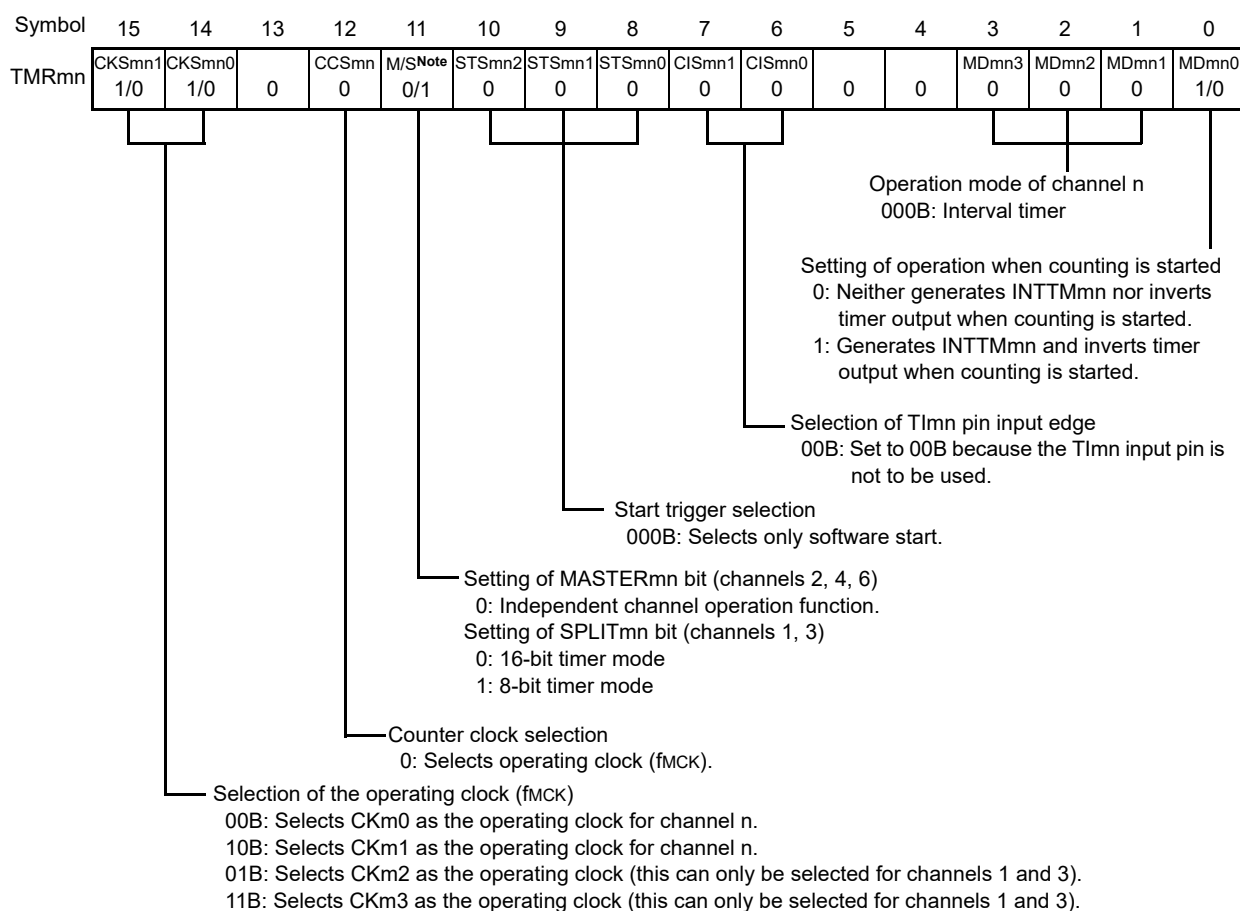


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

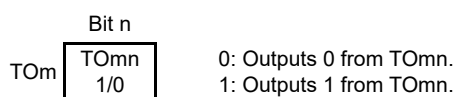
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
 TE mn: Bit n of timer channel enable status register m (TEm)
 TCRmn: Timer counter register mn (TCRmn)
 TDRmn: Timer data register mn (TDRmn)
 TOMn: TOMn pin output signal

Figure 7 - 48 Example of Register Settings for Operation as an Interval Timer or for Square Wave Output

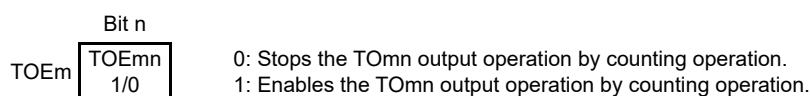
a) Timer mode register mn (TMRmn)



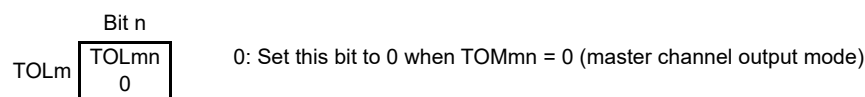
b) Timer output register m (TOM)



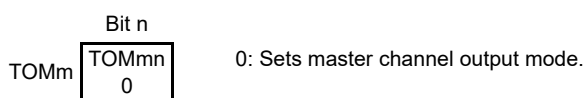
c) Timer output enable register m (TOEm)



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 - 49 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)

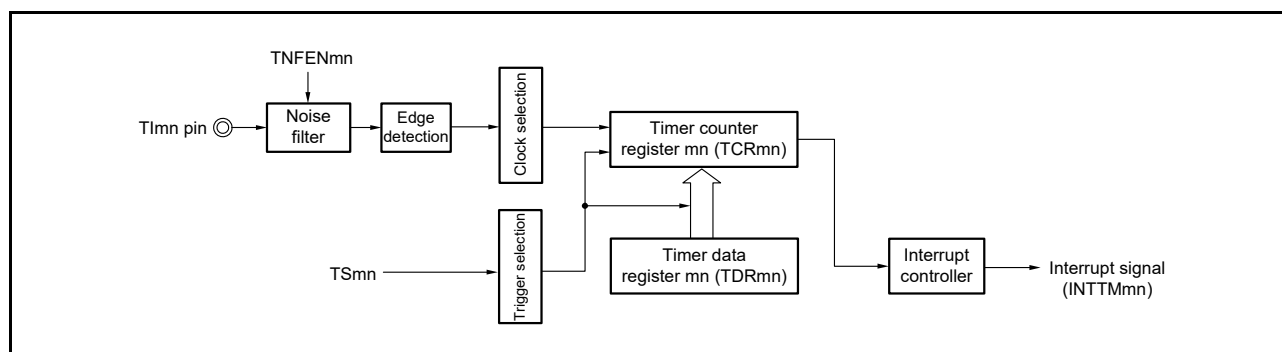
7.8.2 Operation as an external event counter

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.

$$\text{Specified number of counts} = \text{Set value of TDRmn} + 1$$

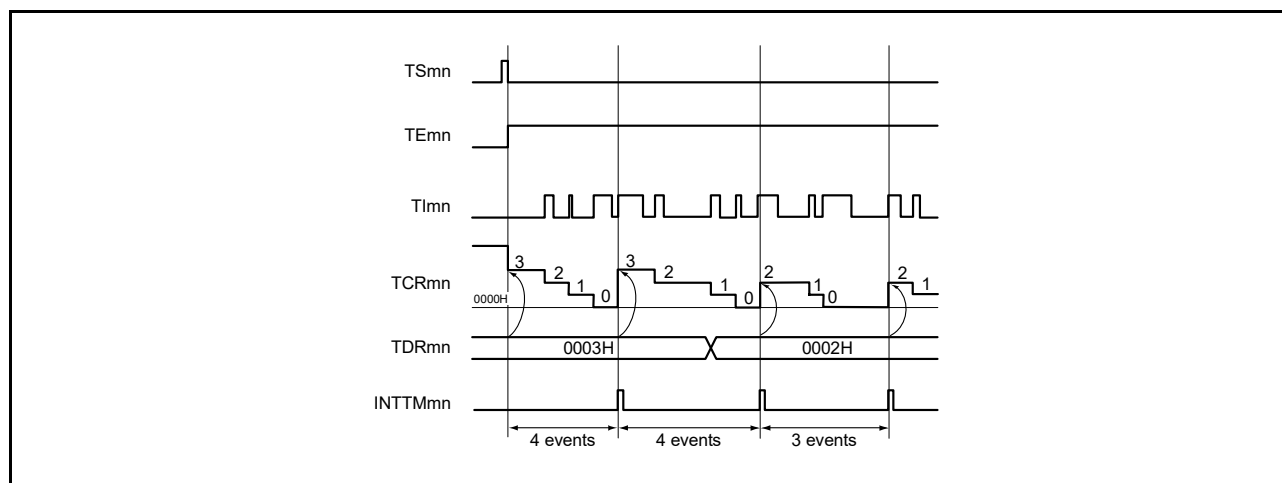
Timer counter register mn (TCRmn) operates as a down counter in the event counter mode. The TCRmn register loads the value of timer data register mn (TDRmn) by setting any operation enable (start) trigger bit (TSmn) of timer channel start register m (TSm) to 1. The TCRmn register counts down each time 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 - 50 Block Diagram for Operation as an External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 7 - 51 Example of Basic Timing during Operation as an External Event Counter

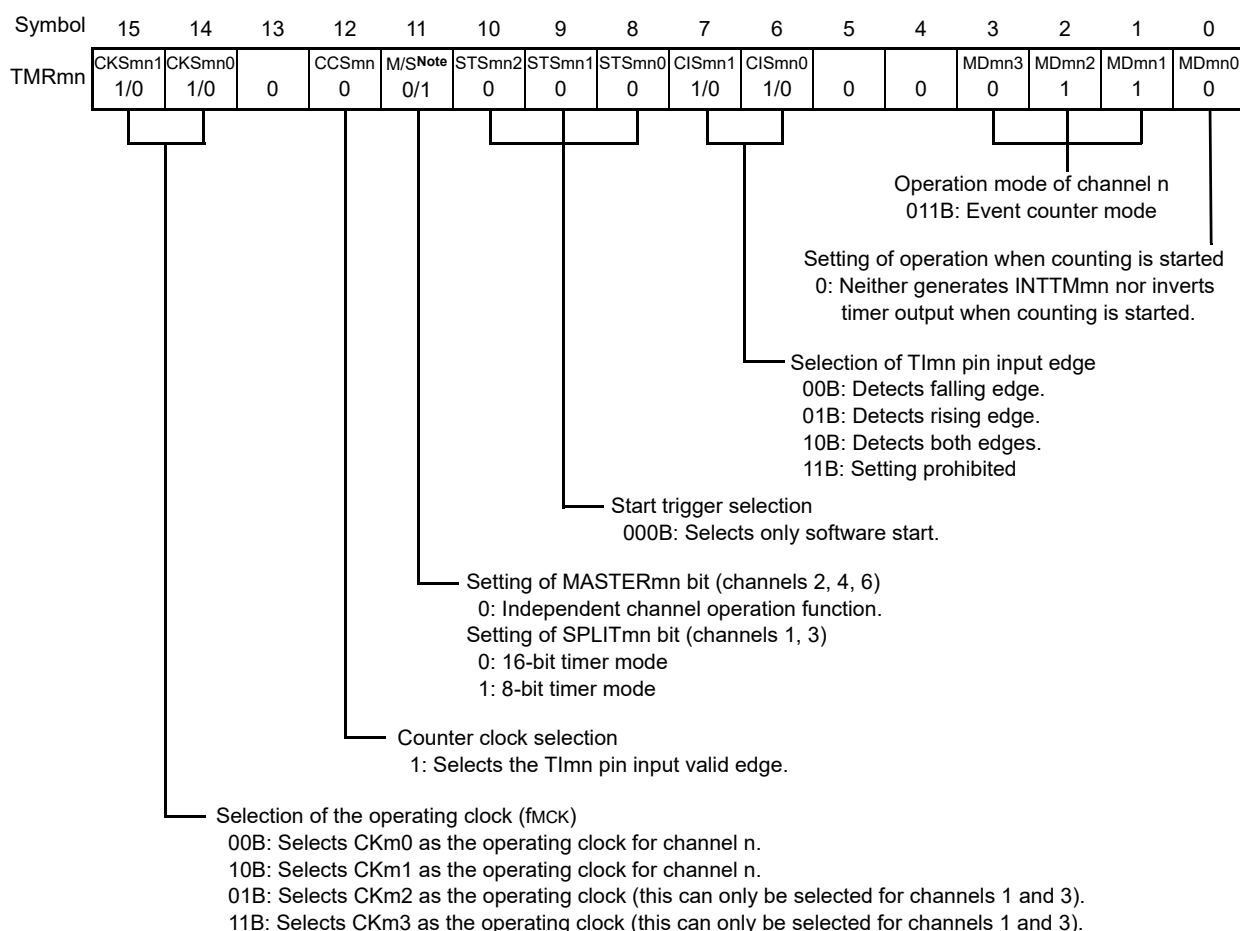


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

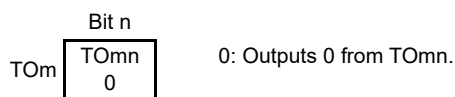
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
 TEmn: Bit n of timer channel enable status register m (TEm)
 TImn: TImn pin input signal
 TCRmn: Timer counter register mn (TCRmn)
 TDRmn: Timer data register mn (TDRmn)

Figure 7 - 52 Example of Register Settings in External Event Counter Mode

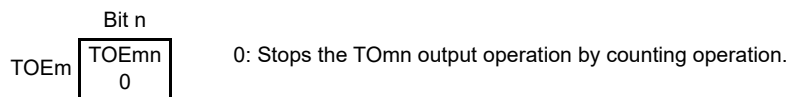
a) Timer mode register mn (TMRmn)



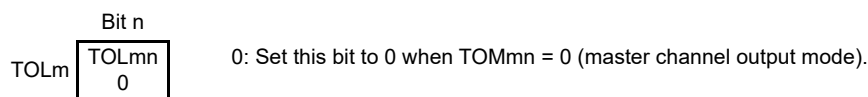
b) Timer output register m (TOm)



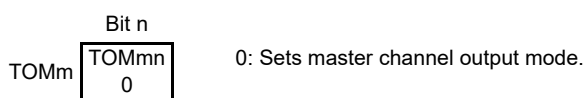
c) Timer output enable register m (TOEm)



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 - 53 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 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 TImn 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 TImn 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 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:

$$\text{Divided clock frequency} = \text{Input clock frequency} / \{(\text{Set value of TDR00} + 1) \times 2\}$$
- When both edges are selected:

$$\text{Divided clock frequency} \approx \text{Input clock frequency} / (\text{Set value of TDR00} + 1)$$

Timer counter register 00 (TCR00) operates as a down counter in the interval timer mode. After the operation enable (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 - 54 Block Diagram for Operation as a Frequency Divider

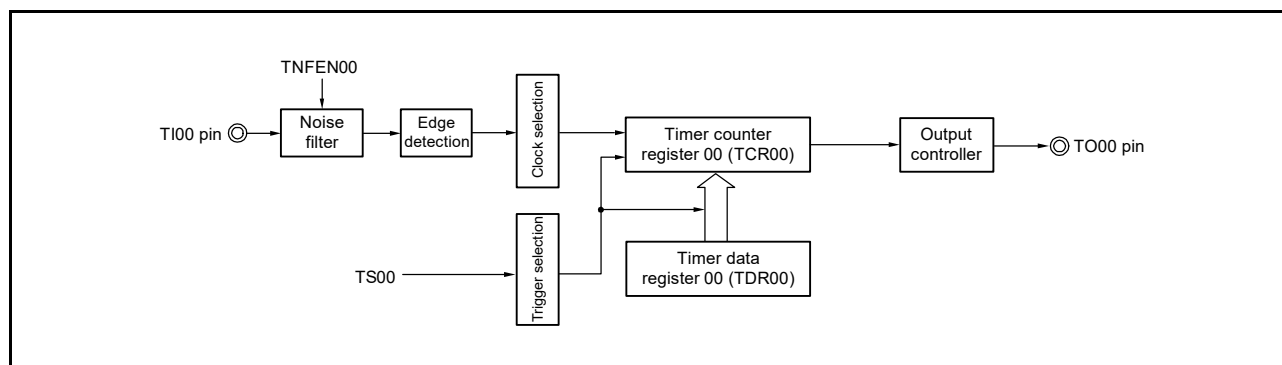
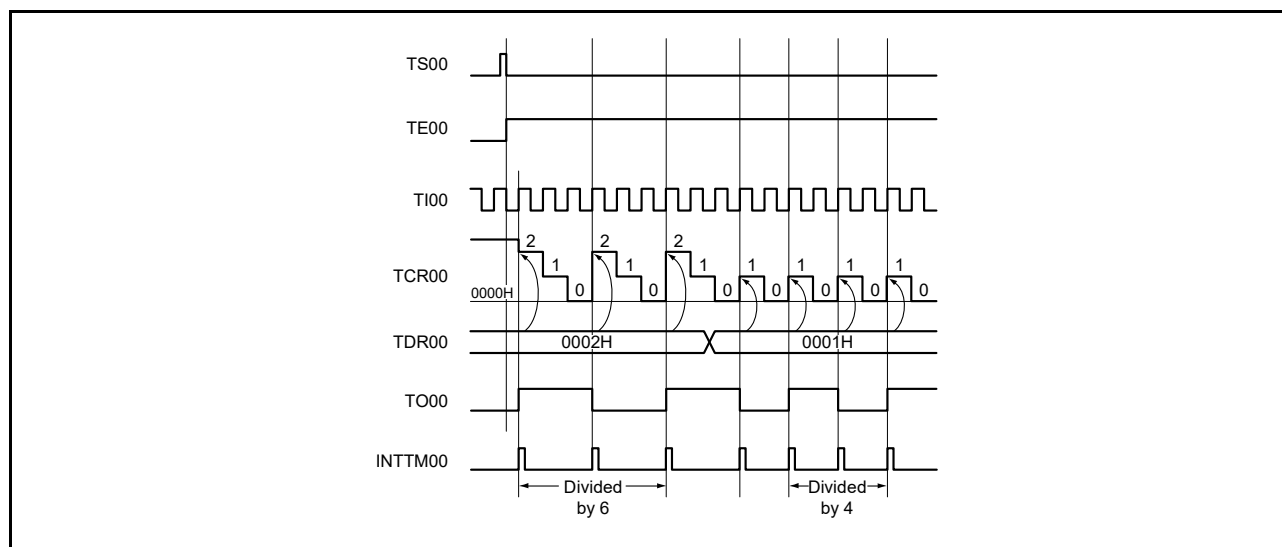


Figure 7 - 55 Example of Basic Timing during Operation as a Frequency Divider (MD000 = 1)

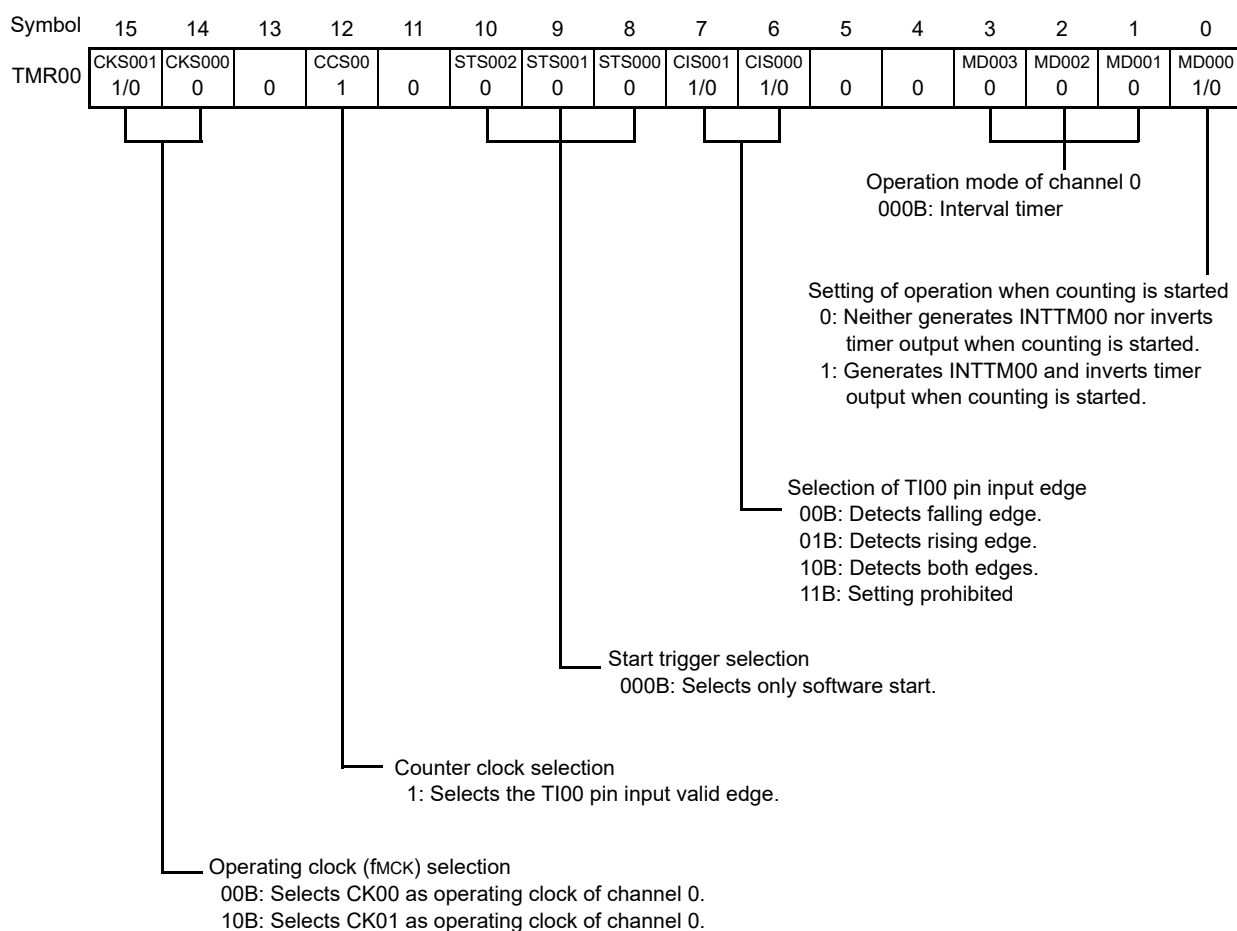


Remark

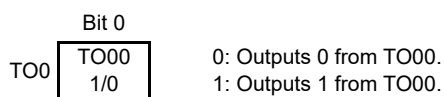
- TS00: Bit 0 of timer channel start register 0 (TS0)
- TE00: Bit 0 of timer channel enable status register 0 (TE0)
- TI00: T100 pin input signal
- TCR00: Timer counter register 00 (TCR00)
- TDR00: Timer data register 00 (TDR00)
- TO00: TO00 pin output signal

Figure 7 - 56 Example of Register Settings for Operation as a Frequency Divider

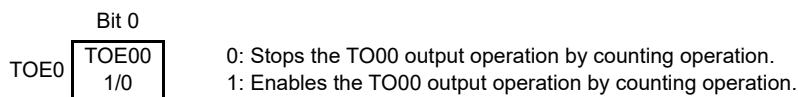
a) Timer mode register 00 (TMR00)



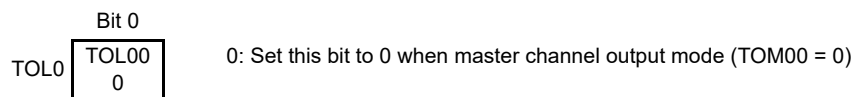
b) Timer output register 0 (TO0)



c) Timer output enable register 0 (TOE0)



d) Timer output level register 0 (TOL0)



e) Timer output mode register 0 (TOM0)

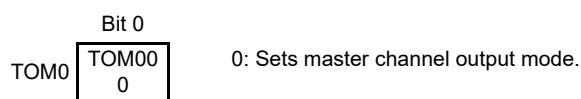


Figure 7 - 57 Procedure for Operations When the Frequency Divider Function Is to be Used (1/2)

	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 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 - 57 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

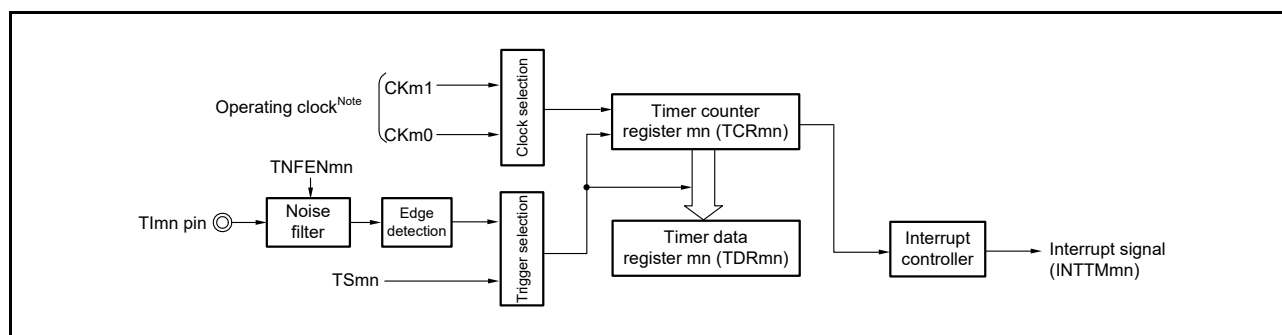
The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEMn bit is set to 1. The pulse interval can be calculated by the following expression.

$$\text{TImn input pulse interval} = \text{Period of counter clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Captured value of TDRmn} + 1))$$

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 cycle of the operating clock occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture mode. When the operation enable (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 counter 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.

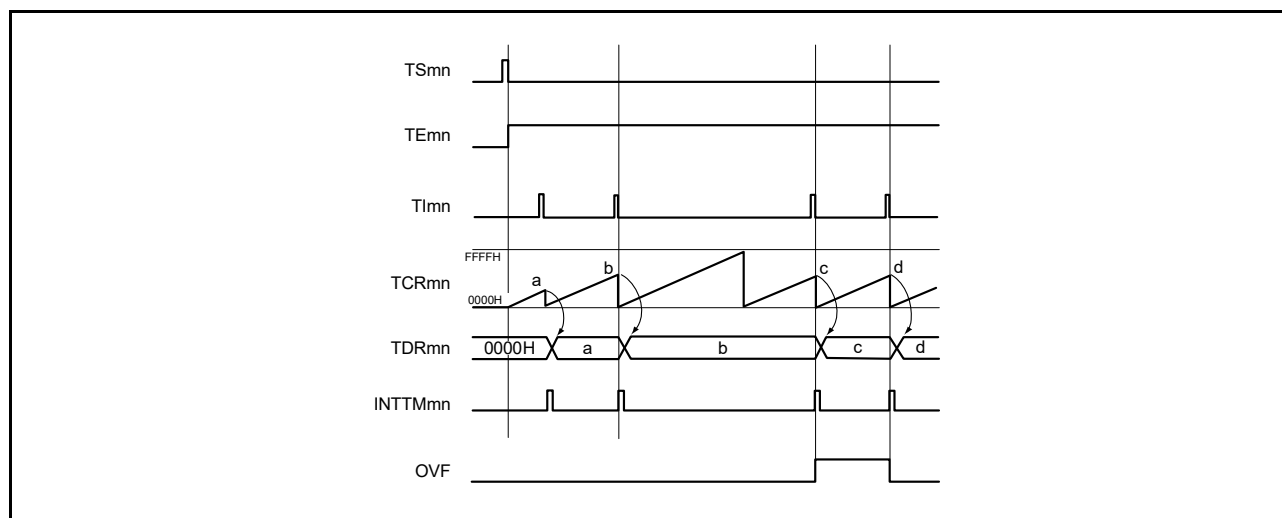
Figure 7 - 58 Block Diagram for Operation for Input Pulse Interval Measurement



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 - 59 Example of Basic Timing during Operation for Input Pulse Interval Measurement (MDmn0 = 0)

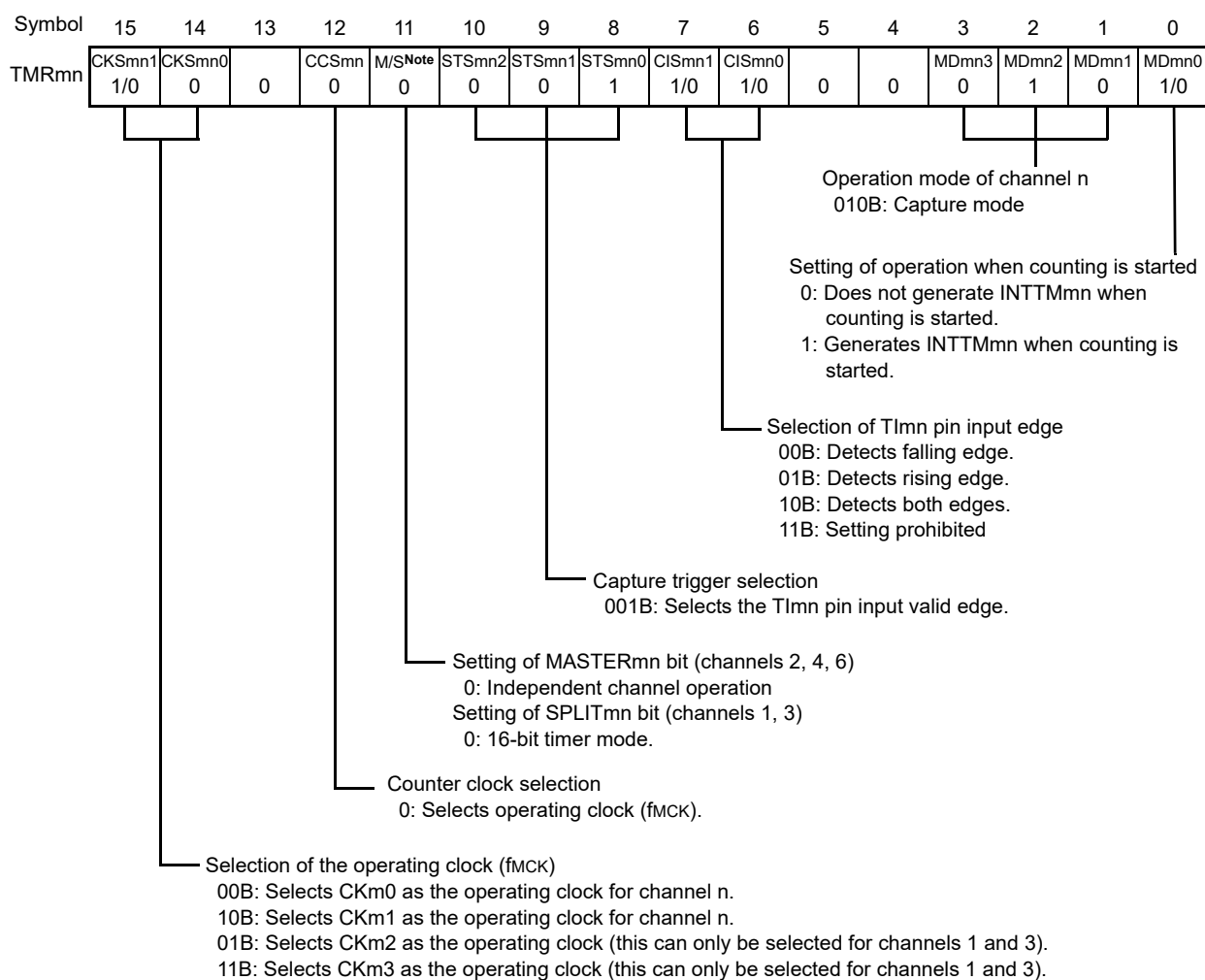


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

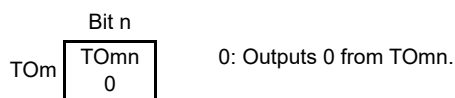
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
 TEmn: Bit n of timer channel enable status register m (TEm)
 TImn: TImn pin input signal
 TCRmn: Timer counter register mn (TCRmn)
 TDRmn: Timer data register mn (TDRmn)
 OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 60 Example of Register Settings for Operation for Input Pulse Interval Measurement

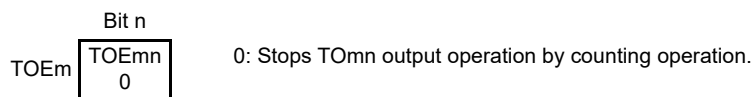
a) Timer mode register mn (TMRmn)



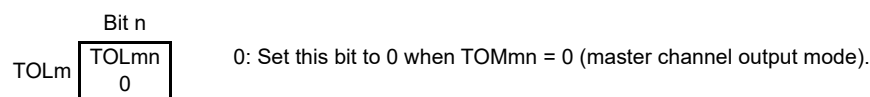
b) Timer output register m (TOM)



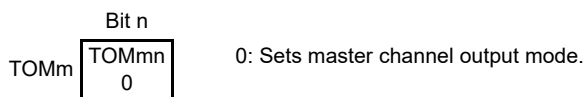
c) Timer output enable register m (TOEm)



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 - 61 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 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.

$$\text{Signal width of TImn input} = \text{Period of counter clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Captured value of TDRmn} + 1))$$

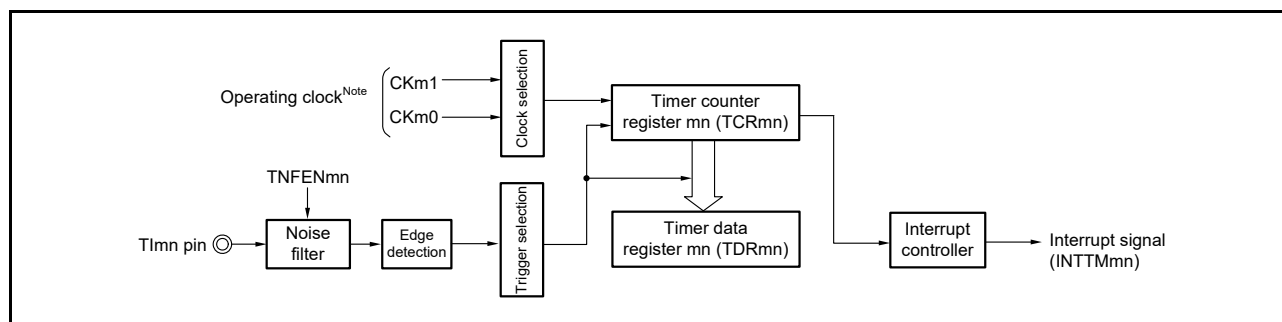
Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one cycle of the operating clock occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture & one-count mode. When the operation enable (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 counter 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.

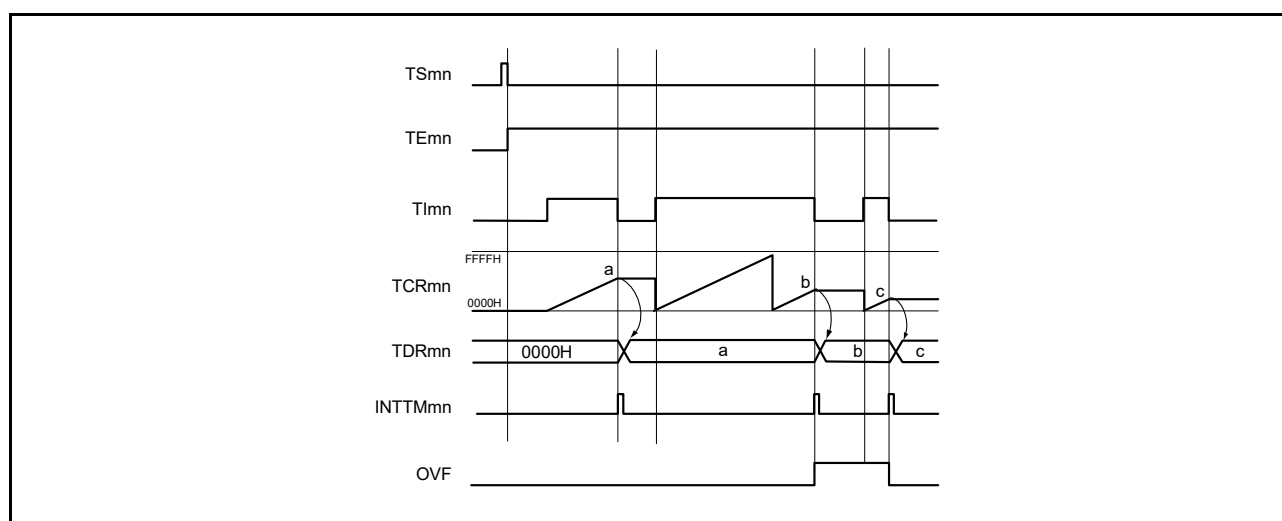
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 7 - 62 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 - 63 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)

TE mn: Bit n of timer channel enable status register m (TE m)

TImn: TImn pin input signal

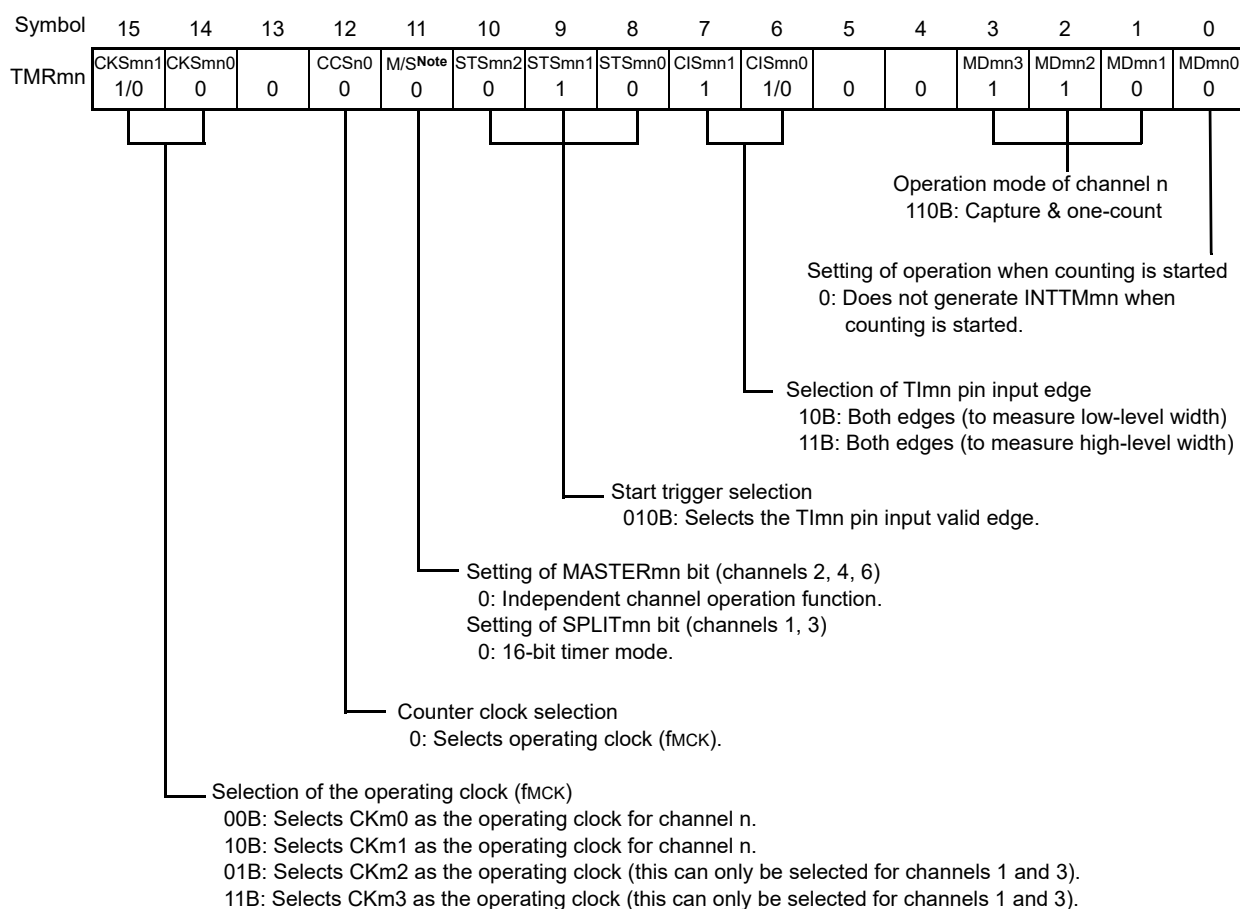
TCRmn: Timer counter register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

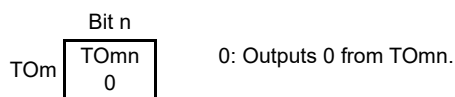
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 64 Example of Register Settings for Operation for Input Signal High-/Low-level Width Measurement

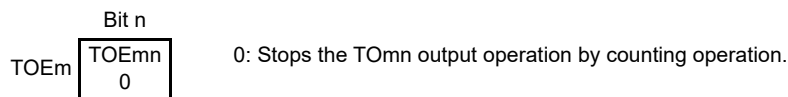
a) Timer mode register mn (TMRmn)



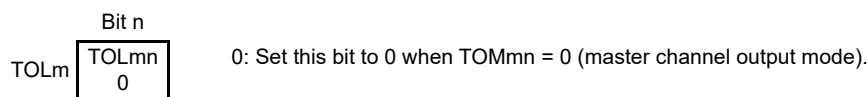
b) Timer output register m (TOm)



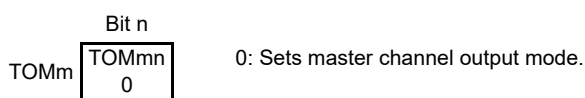
c) Timer output enable register m (TOEm)



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 - 65 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 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. →	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.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

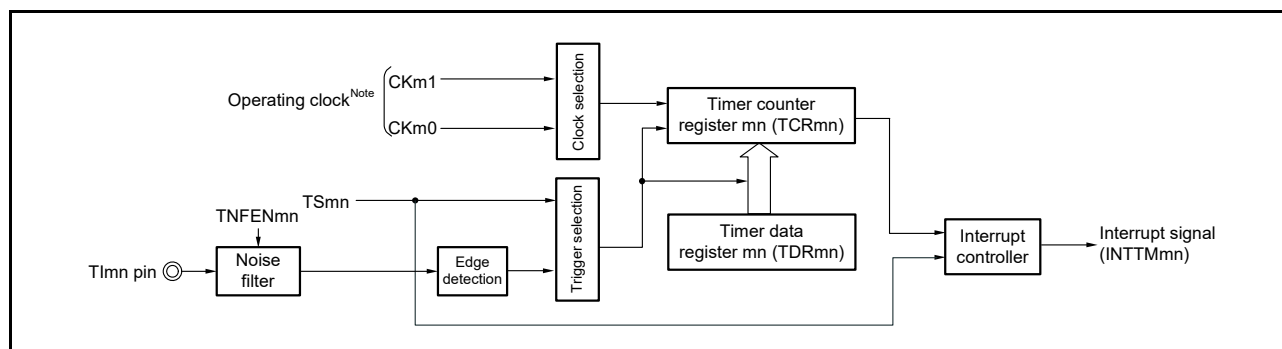
7.8.6 Operation as a delay counter

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.

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of counter clock} \times (\text{Set value of TDRmn} + 1)$$

Timer counter register mn (TCRmn) operates as a down counter in the one-count mode. When the operation enable (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 counter 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.

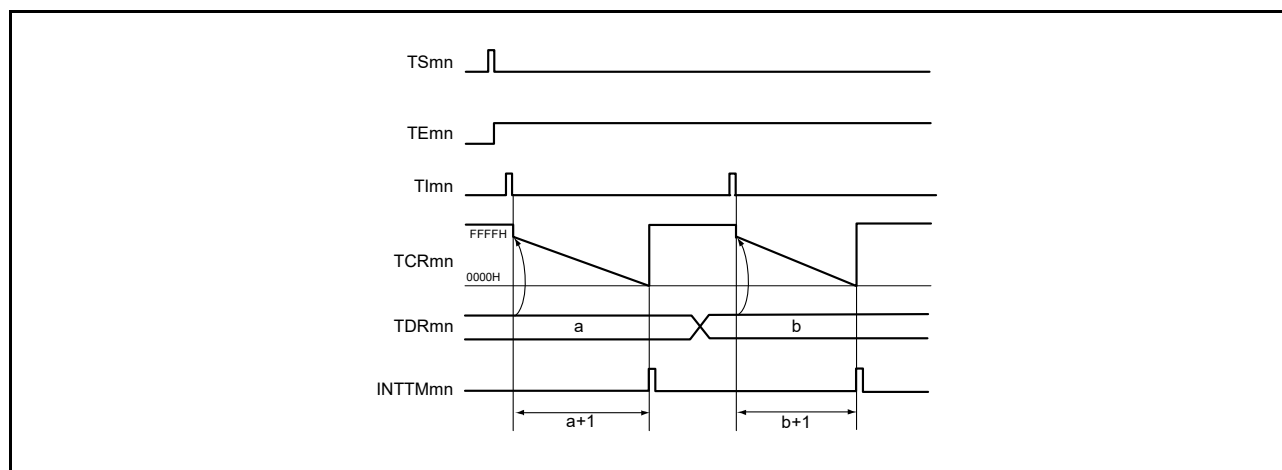
Figure 7 - 66 Block Diagram for Operation as a Delay Counter



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 - 67 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)

TE mn: Bit n of timer channel enable status register m (TE m)

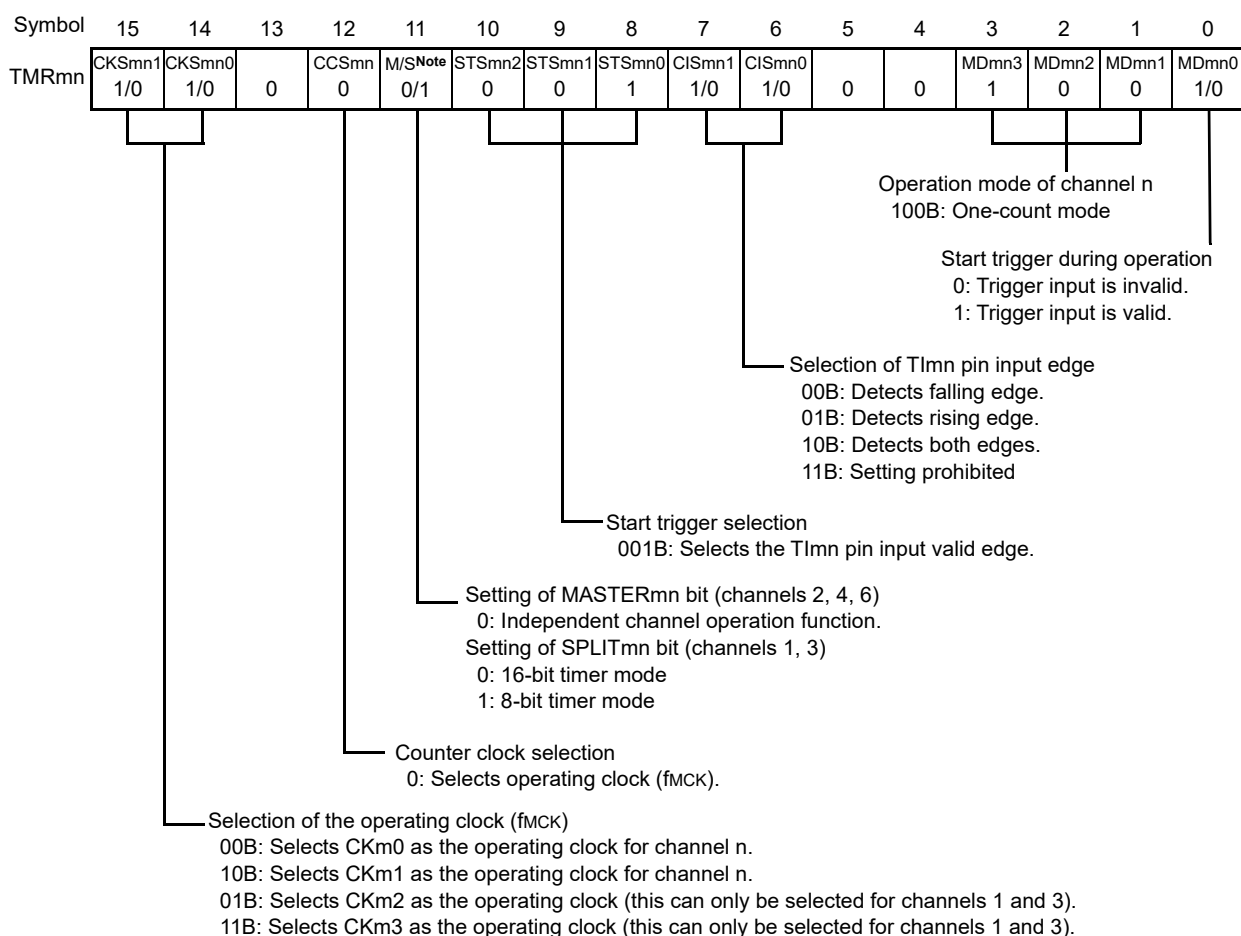
TImn: TImn pin input signal

TCRmn: Timer counter register mn (TCRmn)

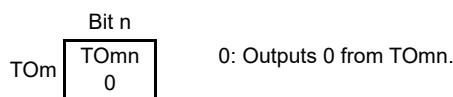
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 68 Example of Register Settings for Operation as a Delay Counter

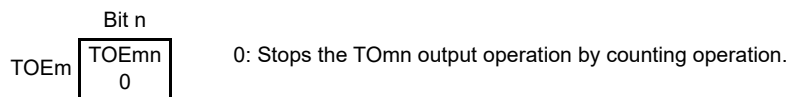
a) Timer mode register mn (TMRmn)



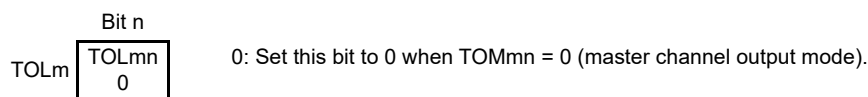
b) Timer output register m (TOm)



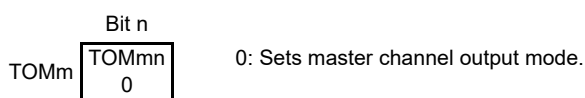
c) Timer output enable register m (TOEm)



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 - 69 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 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.

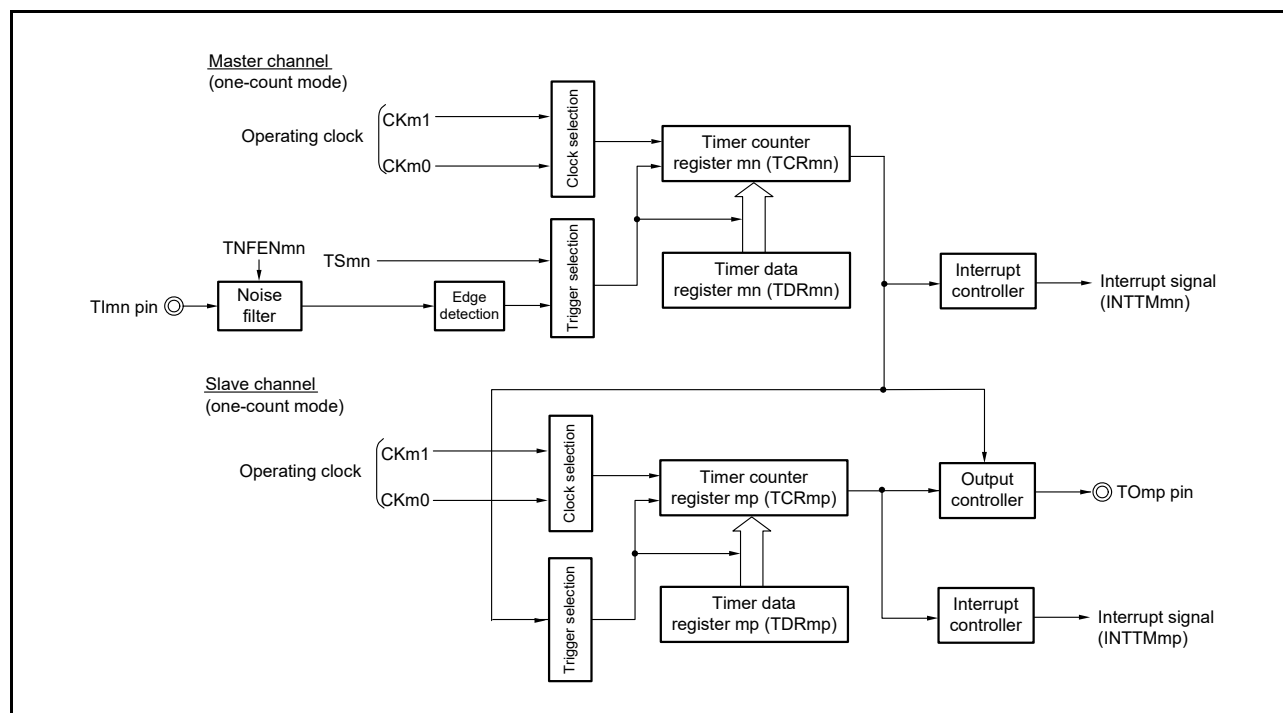
$$\begin{aligned}\text{Delay time} &= \{\text{Set value of TDRmn (master)} + 2\} \times \text{Counter clock period} \\ \text{Pulse width} &= \{\text{Set value of TDRmp (slave)}\} \times \text{Counter clock period}\end{aligned}$$

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 counter 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 counter 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.

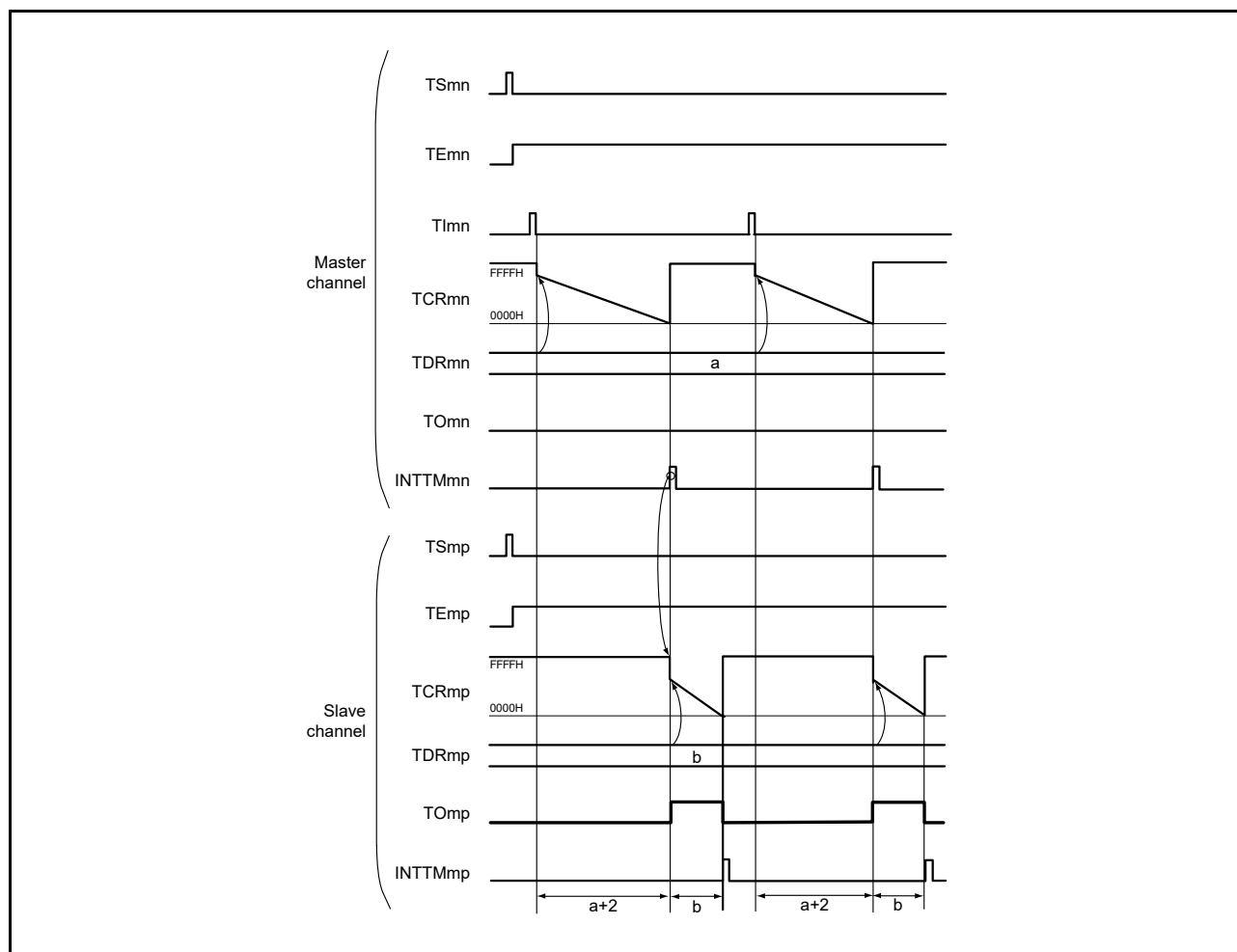
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 70 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 ($n < p \leq 7$)

Figure 7 - 71 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, TEmP: Bits n and p of timer channel enable status register m (TEM)

TImn, TImp: Signals on the TImn and TImp input pins

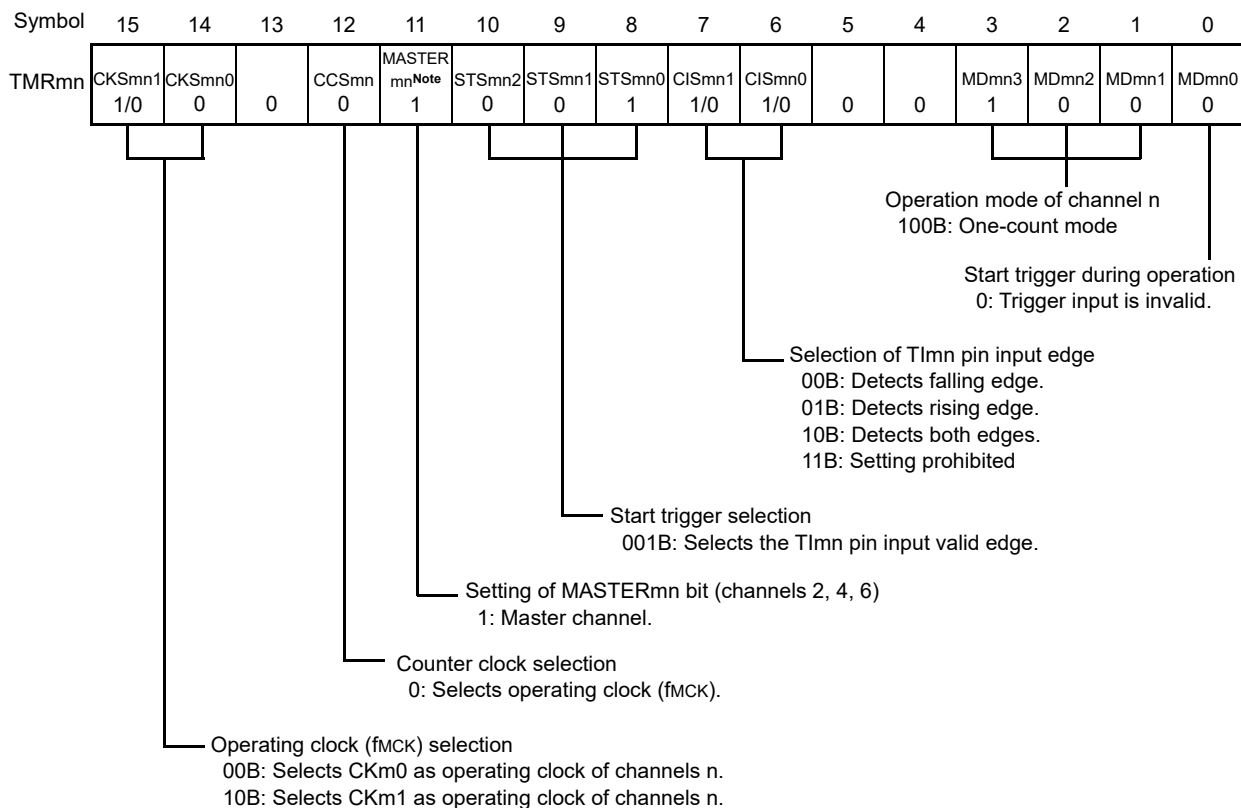
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

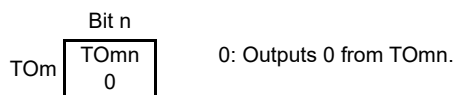
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 7 - 72 Example of Register Settings for the Master Channel When the One-shot Pulse Output Function Is to be Used

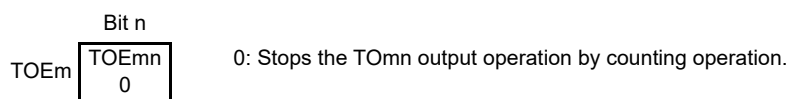
a) Timer mode register mn (TMRmn)



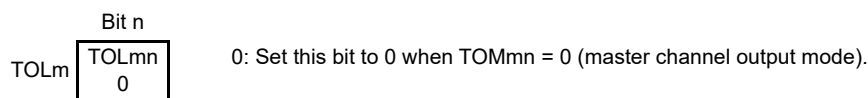
b) Timer output register m (TOM)



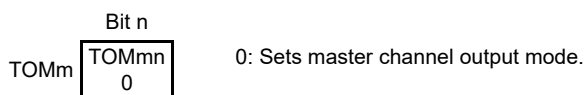
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)

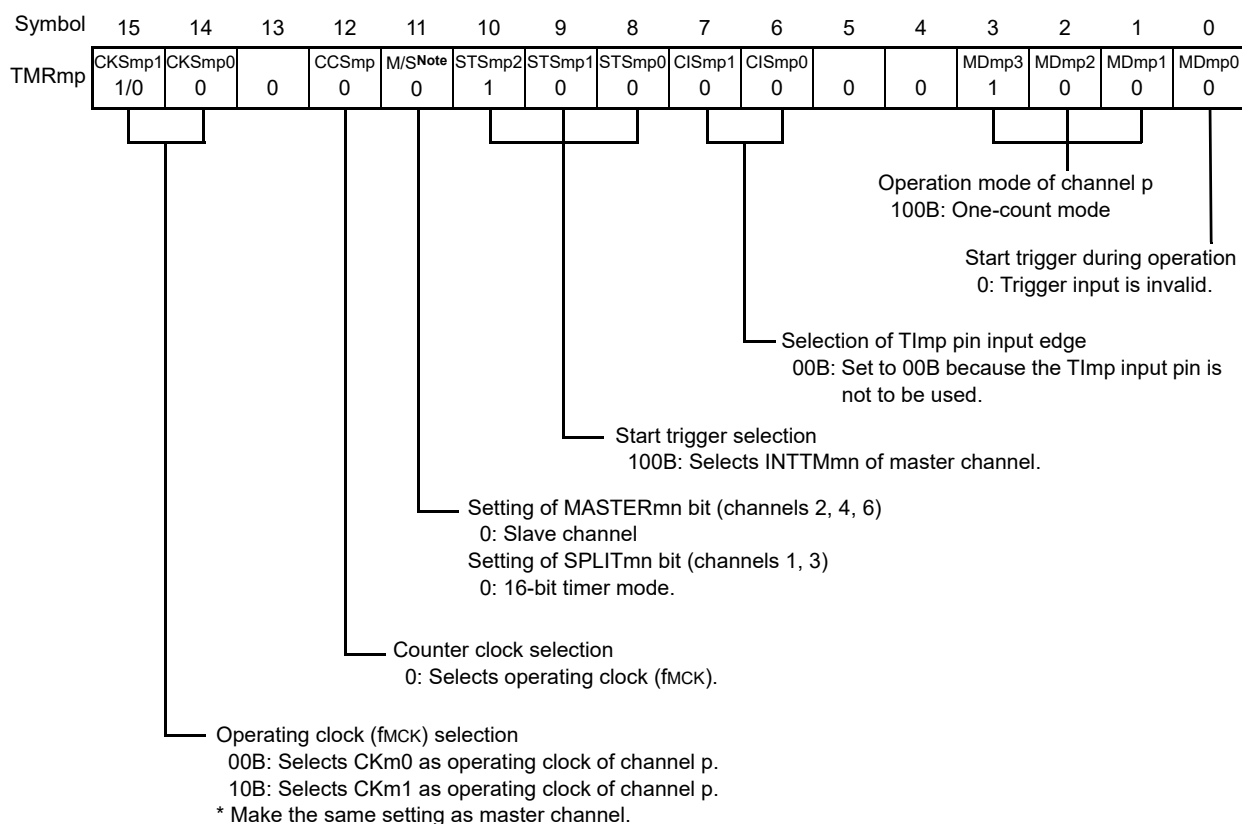


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

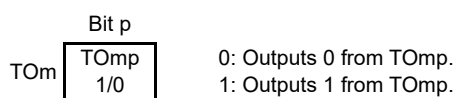
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 73 Example of Register Settings for the Slave Channel When the One-shot Pulse Output Function Is to be Used

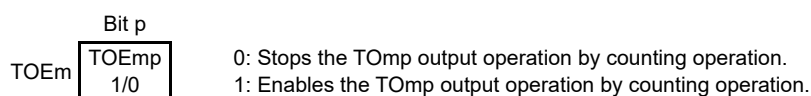
a) Timer mode register mp (TMRmp)



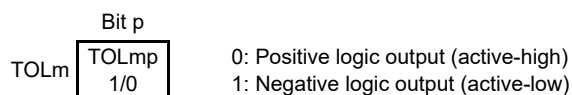
b) Timer output register m (TOM)



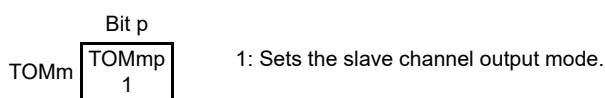
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
 TMRm1, TMRm3: 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 - 74 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 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 - 74 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.
	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 counter 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} × Counter 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}

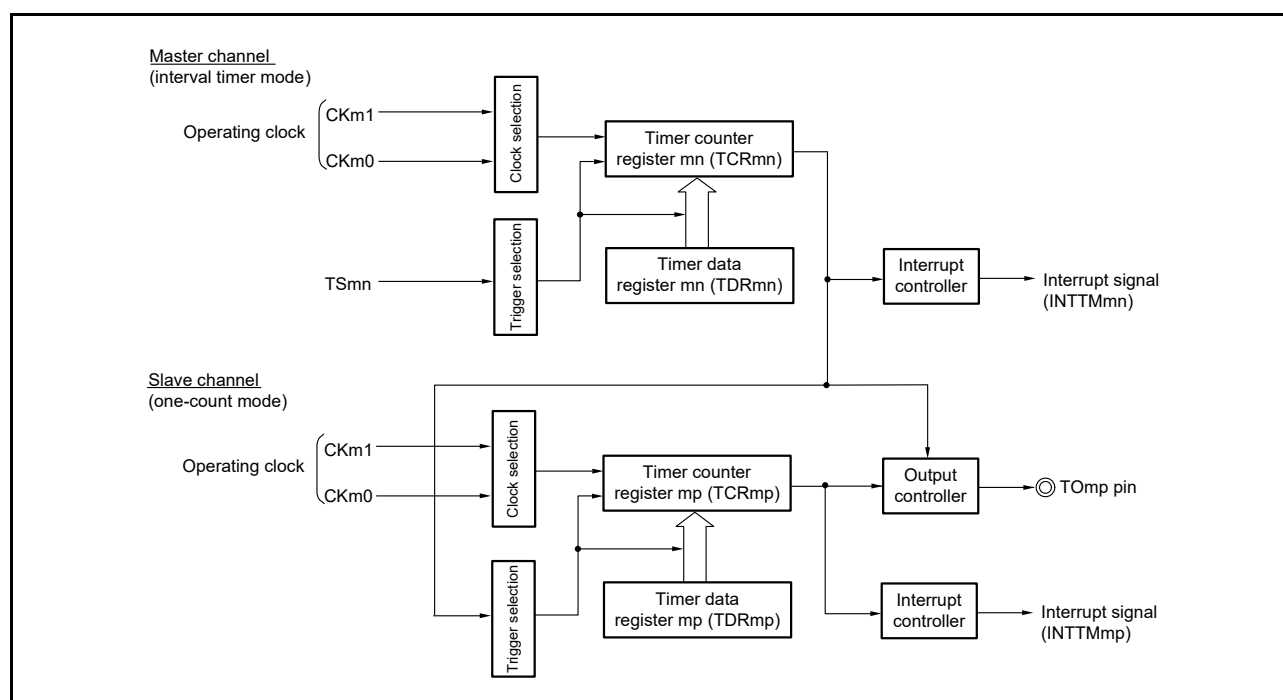
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the operation enable (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 counter 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 operation 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.

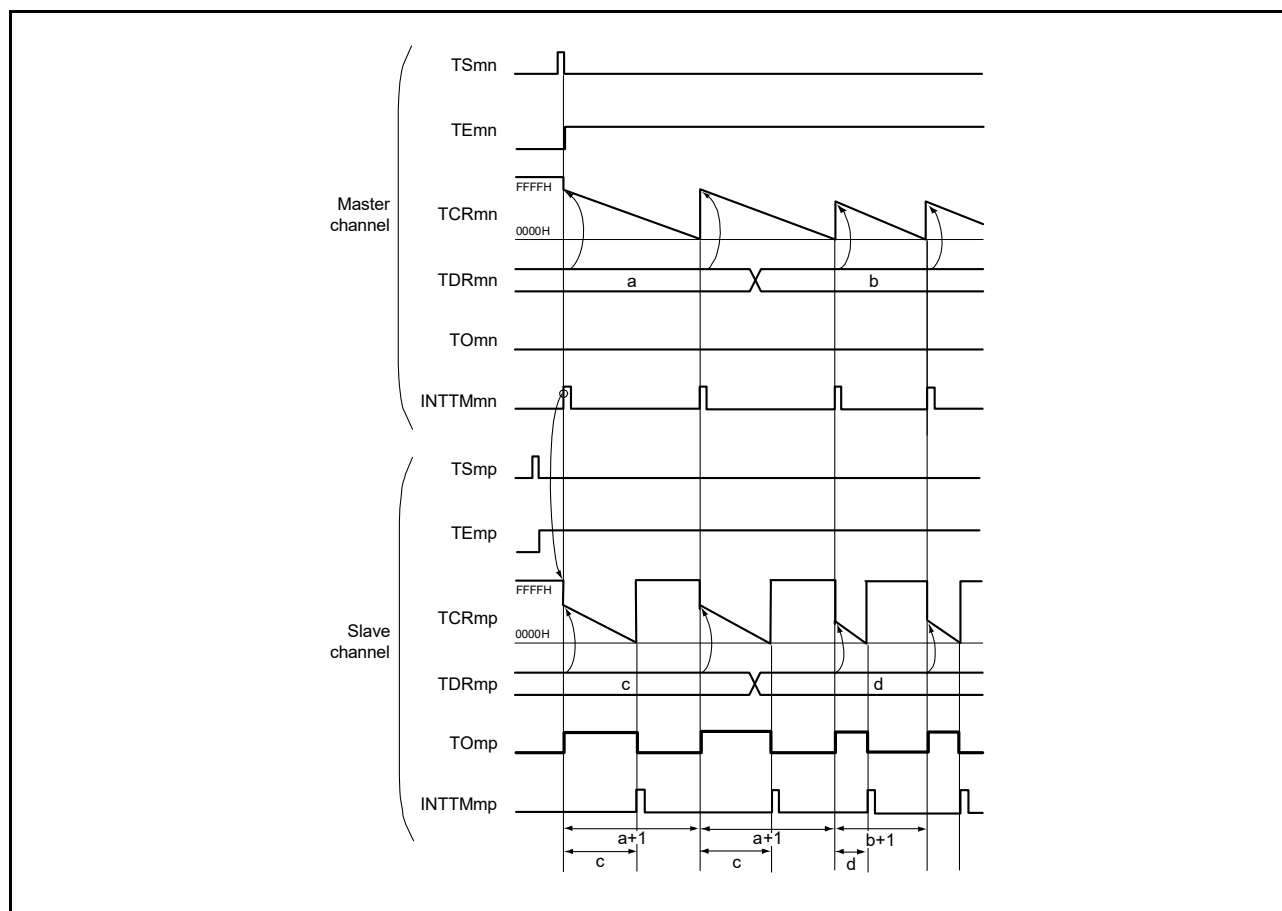
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
 p: Slave channel number (n < p ≤ 7)

Figure 7 - 75 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 - 76 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 ($n < p \leq 7$)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)

TEmn, TEmp: Bits n and p of timer channel enable status register m (TEm)

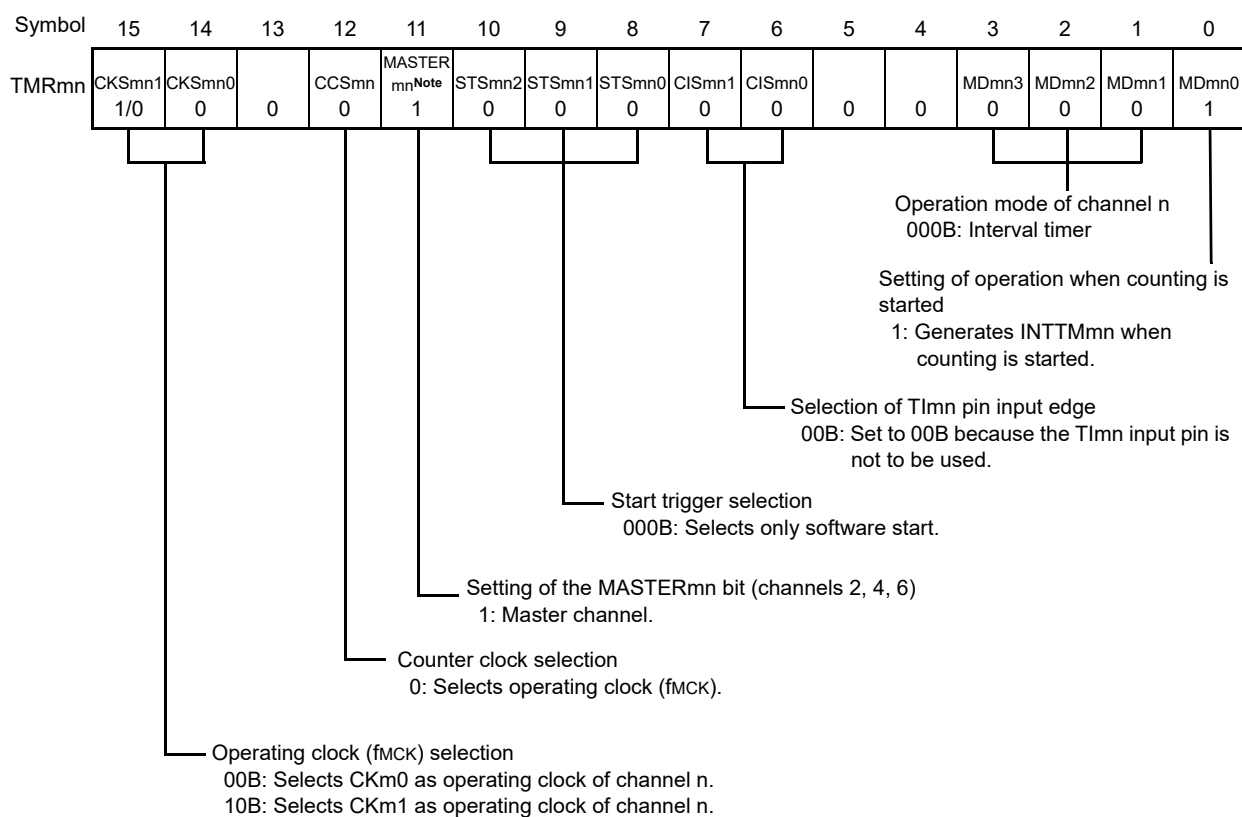
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

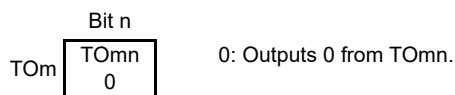
TOMn, TOMP: Signals on the TOMn and TOMP output pins

Figure 7 - 77 Example of Register Settings for the Master Channel When the PWM Function Is to be Used

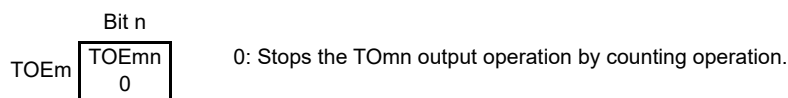
a) Timer mode register mn (TMRmn)



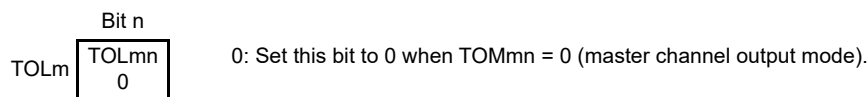
b) Timer output register m (TOM)



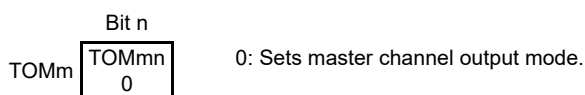
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)

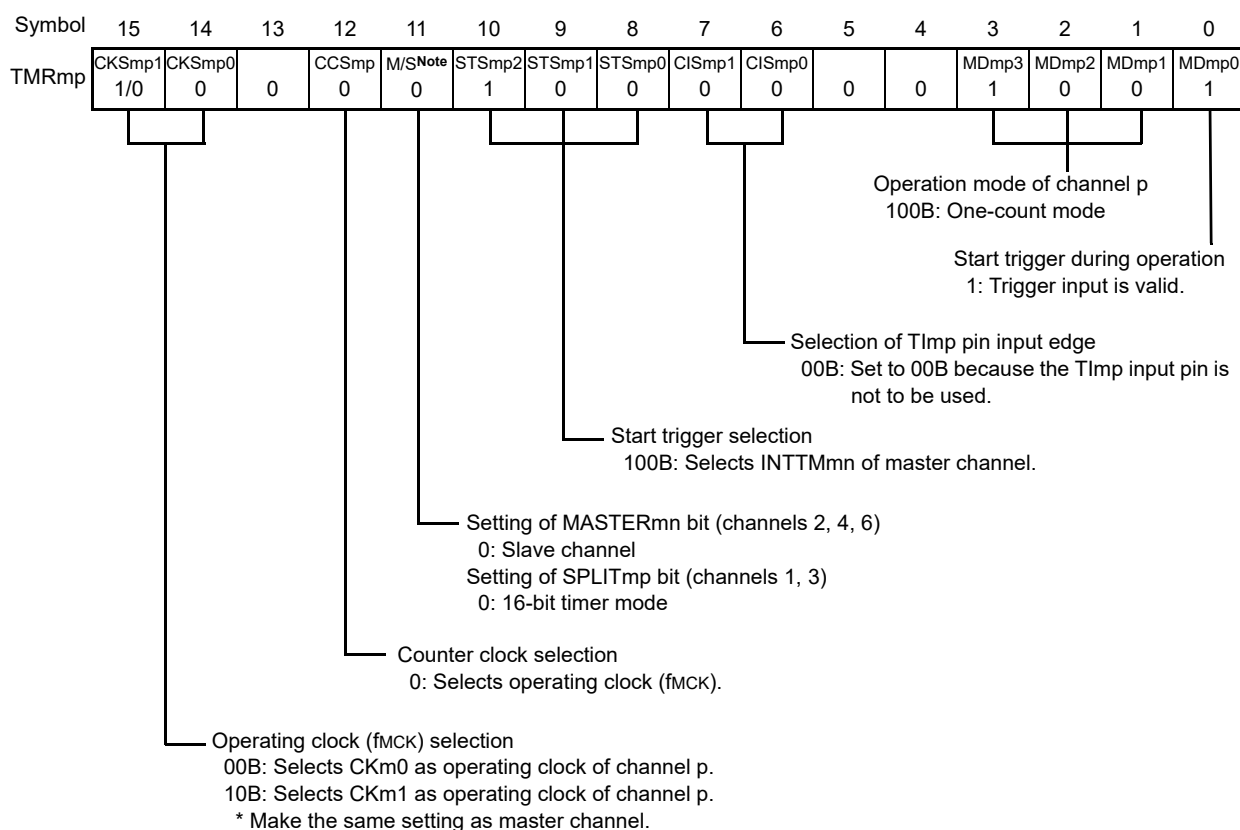


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
 TMRm0: Fixed to 0

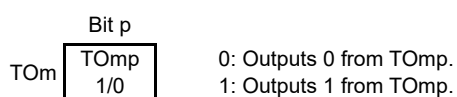
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 78 Example of Register Settings for the Slave Channel When the PWM Function Is to be Used

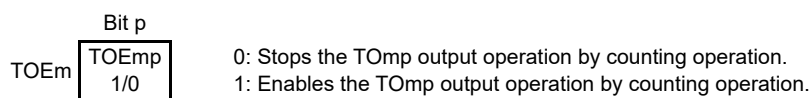
a) Timer mode register mp (TMRmp)



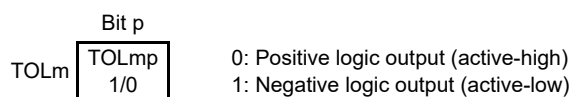
b) Timer output register m (TOM)



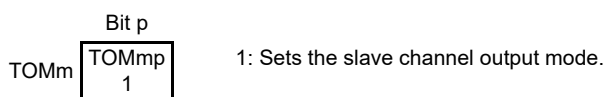
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
 TMRm1, TMRm3: 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 - 79 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 - 79 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, TEmP = 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 counter 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, TEmP = 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: Master 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.

$$\begin{aligned} \text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Counter clock period} \\ \text{Duty factor 1 [\%]} &= \{\text{Set value of TDRmp (slave 1)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{Duty factor 2 [\%]} &= \{\text{Set value of TDRmq (slave 2)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \end{aligned}$$

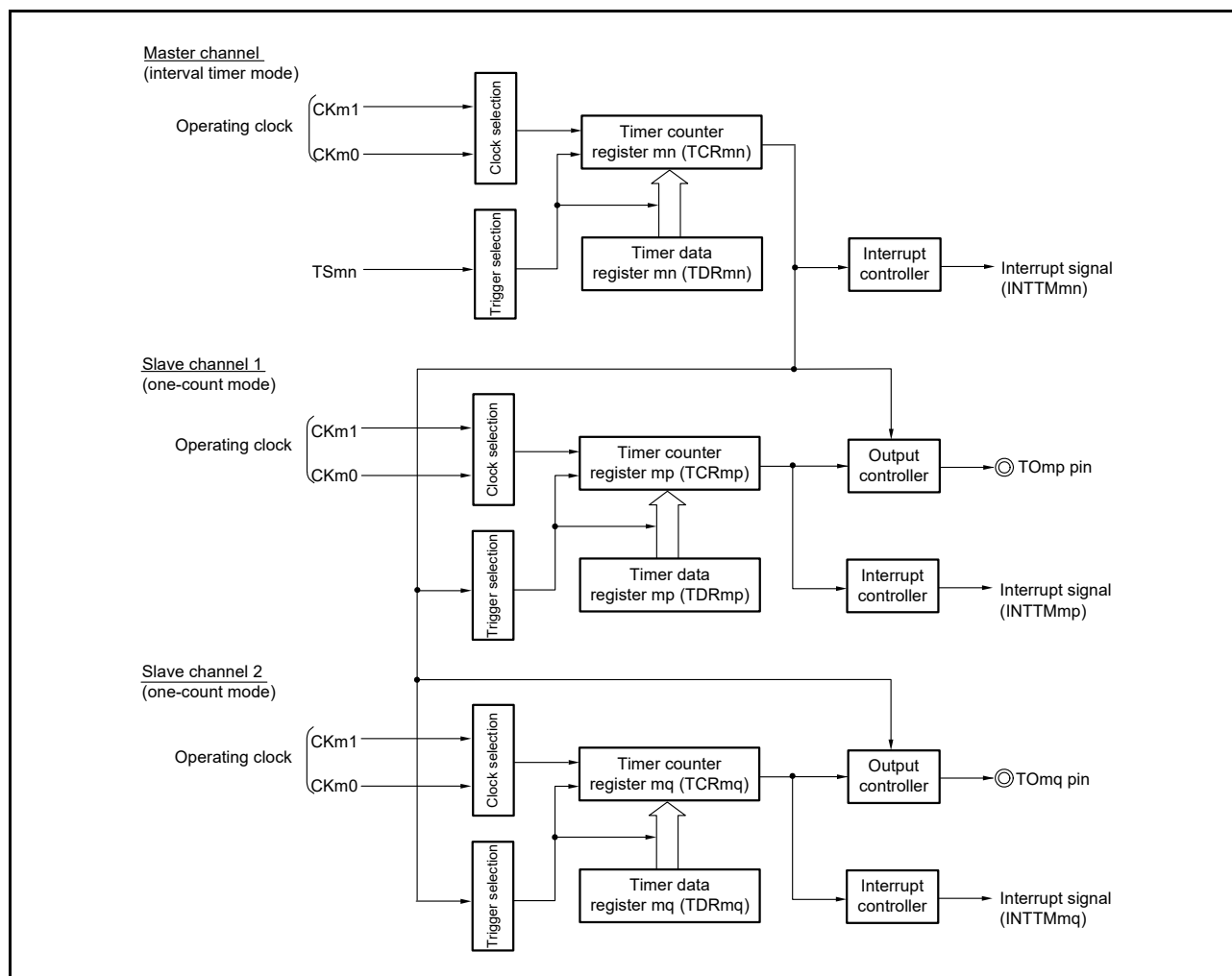
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

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 counter 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 counter 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).

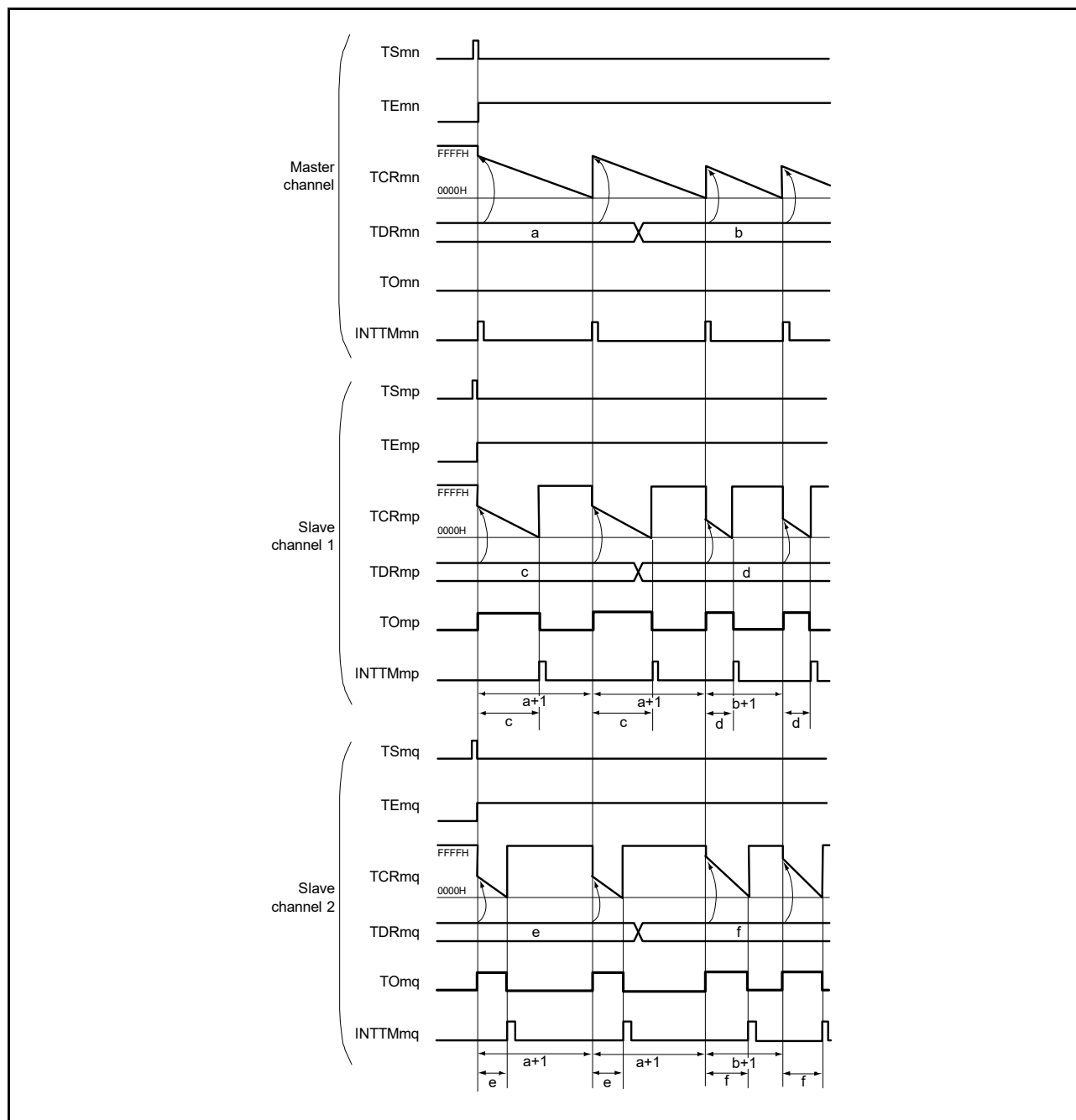
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
 p: Slave channel number, q: Slave channel number
 n < p < q ≤ 7 (Where p and q are integers)

Figure 7 - 80 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)

Figure 7 - 81 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)

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)

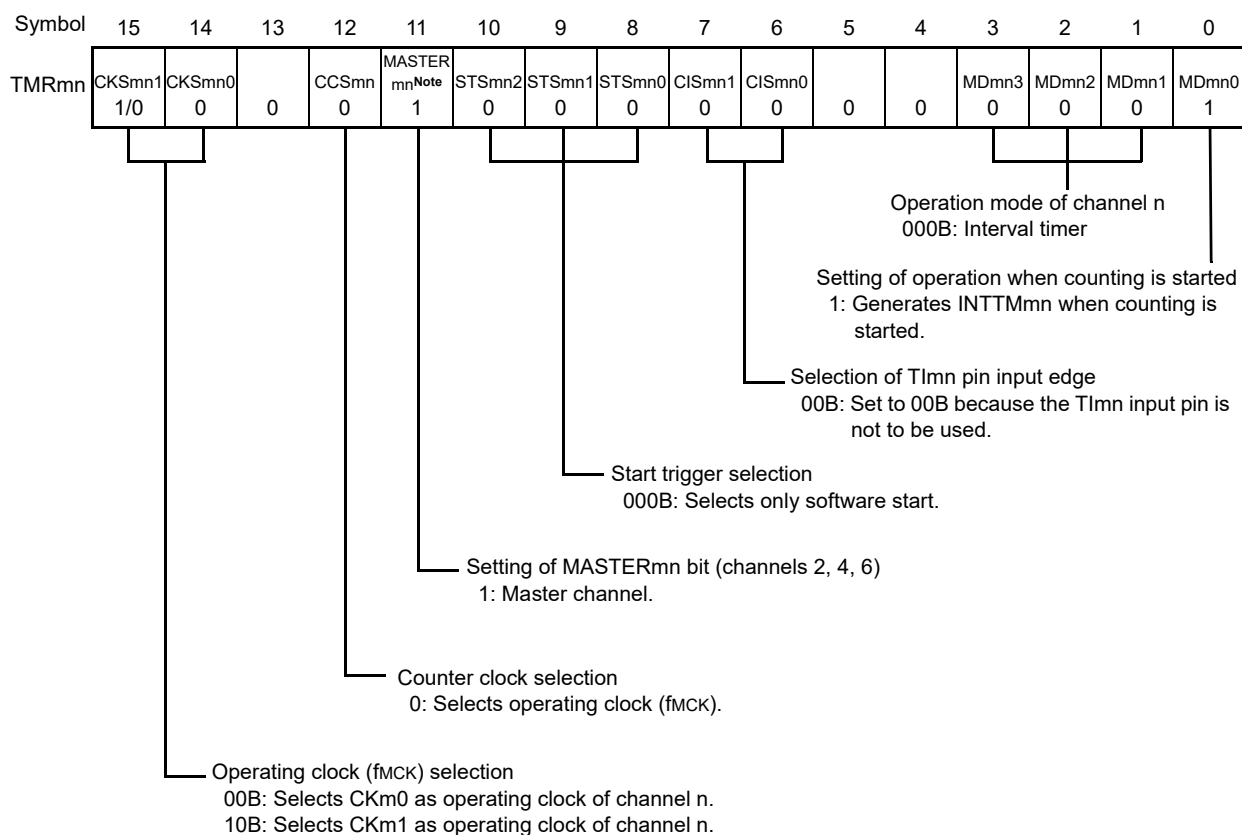
TCR_{mn} , TCR_{mp} , TCR_{mq} : Timer counter registers mn, mp, mq (TCR_{mn} , TCR_{mp} , TCR_{mq})

TDR_{mn} , TDR_{mp} , TDR_{mq} : Timer data registers mn, mp, mq (TDR_{mn} , TDR_{mp} , TDR_{mq})

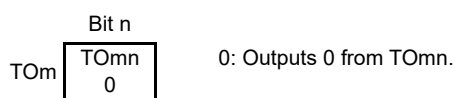
TOM_{mn} , TOM_{mp} , TOM_{mq} : Signals on the TOM_{mn} , TOM_{mp} , and TOM_{mq} output pins

Figure 7 - 82 Example of Register Settings for the Master Channel When the Multiple PWM Output Function Is to be Used

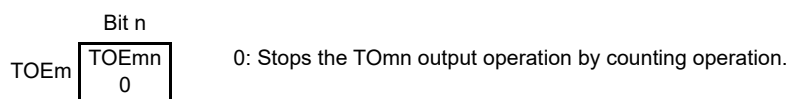
a) Timer mode register mn (TMRmn)



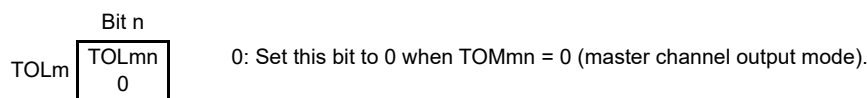
b) Timer output register m (TOM)



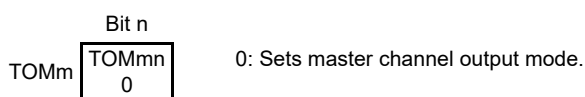
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)

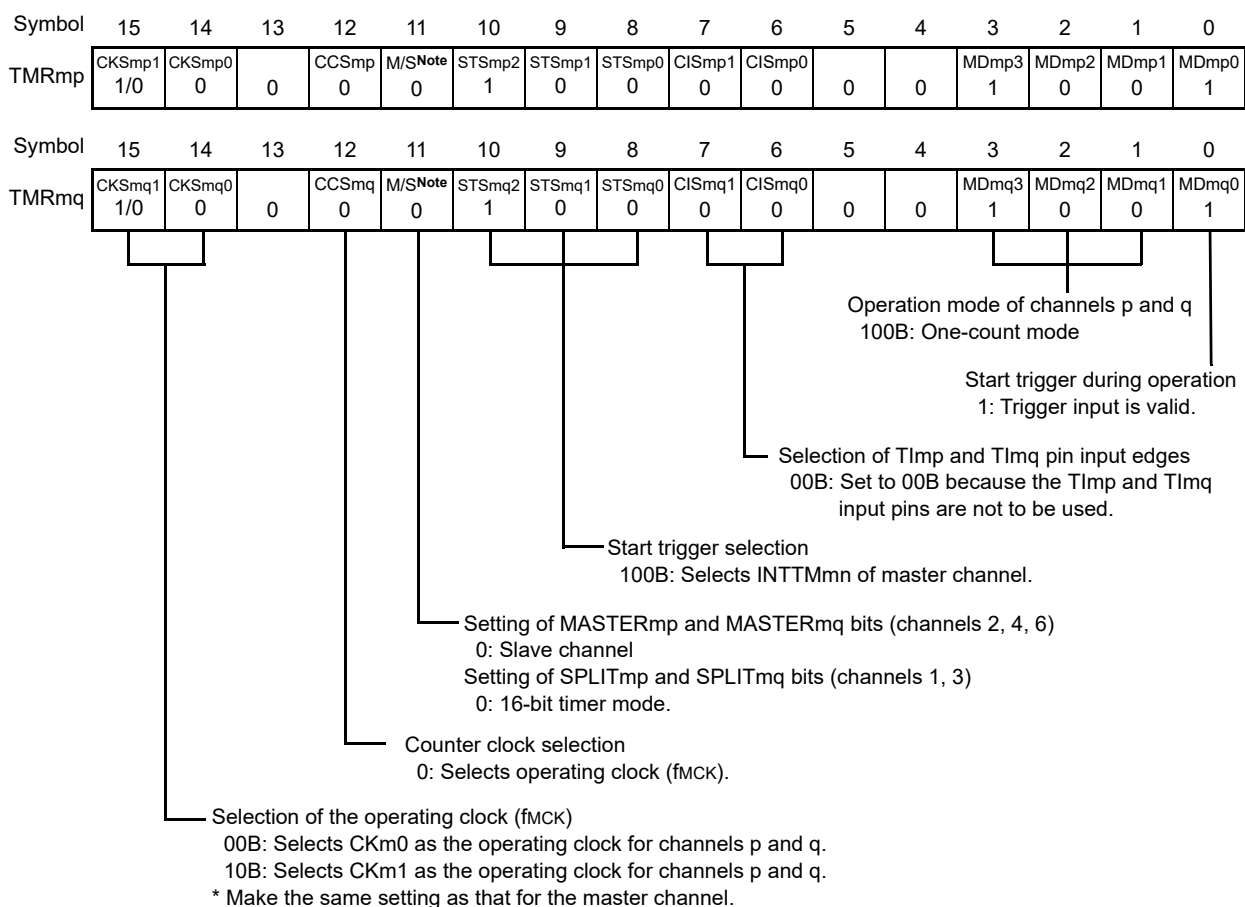


Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
 TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)

Figure 7 - 83 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 ≤ 7 (Where p and q are integers)

Figure 7 - 84 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.

Figure 7 - 84 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, 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 counter 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 counter 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, 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 integers)

7.9.4 Remote-control output function

The remote-control output function operates with the use of the PWM output function. The PWM output function is used with the set of channels 2 and 3 and channels 4 and 5. For the procedure to set each of the channels, see **7.9.2**

Operation for the PWM function. The PWM signal output by channel 3 is used as the masking waveform, the PWM signal output by channel 5 is used as the carrier waveform, and the logical AND of these signals is output as a remote-control signal. The portion of the remote-control output signal in which high-level width output is not masked consists of a carrier signal roughly in the range from 20 to 60 kHz.

Figure 7 - 85 Remote-control Output

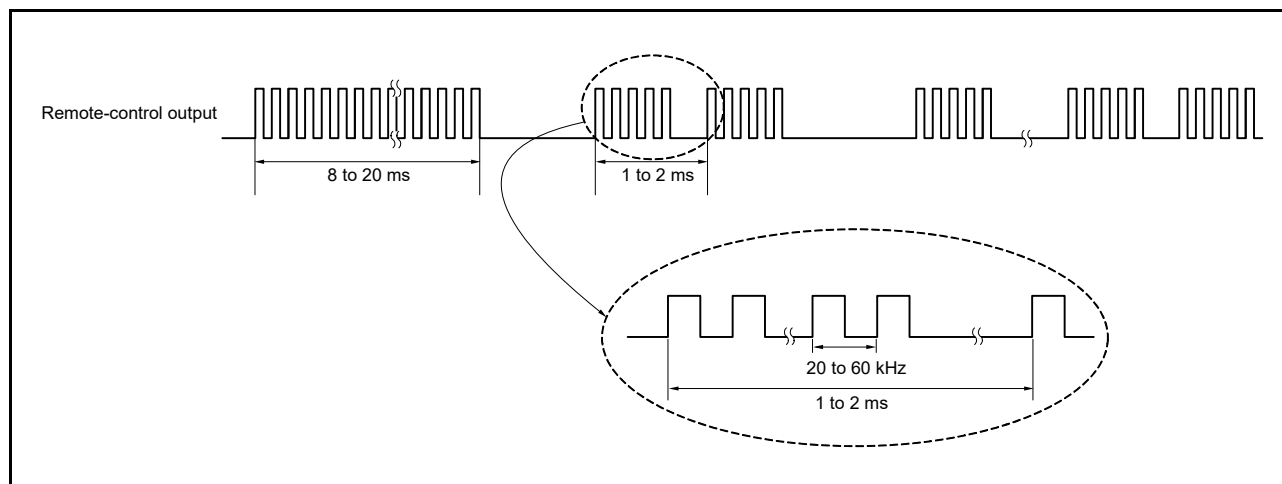
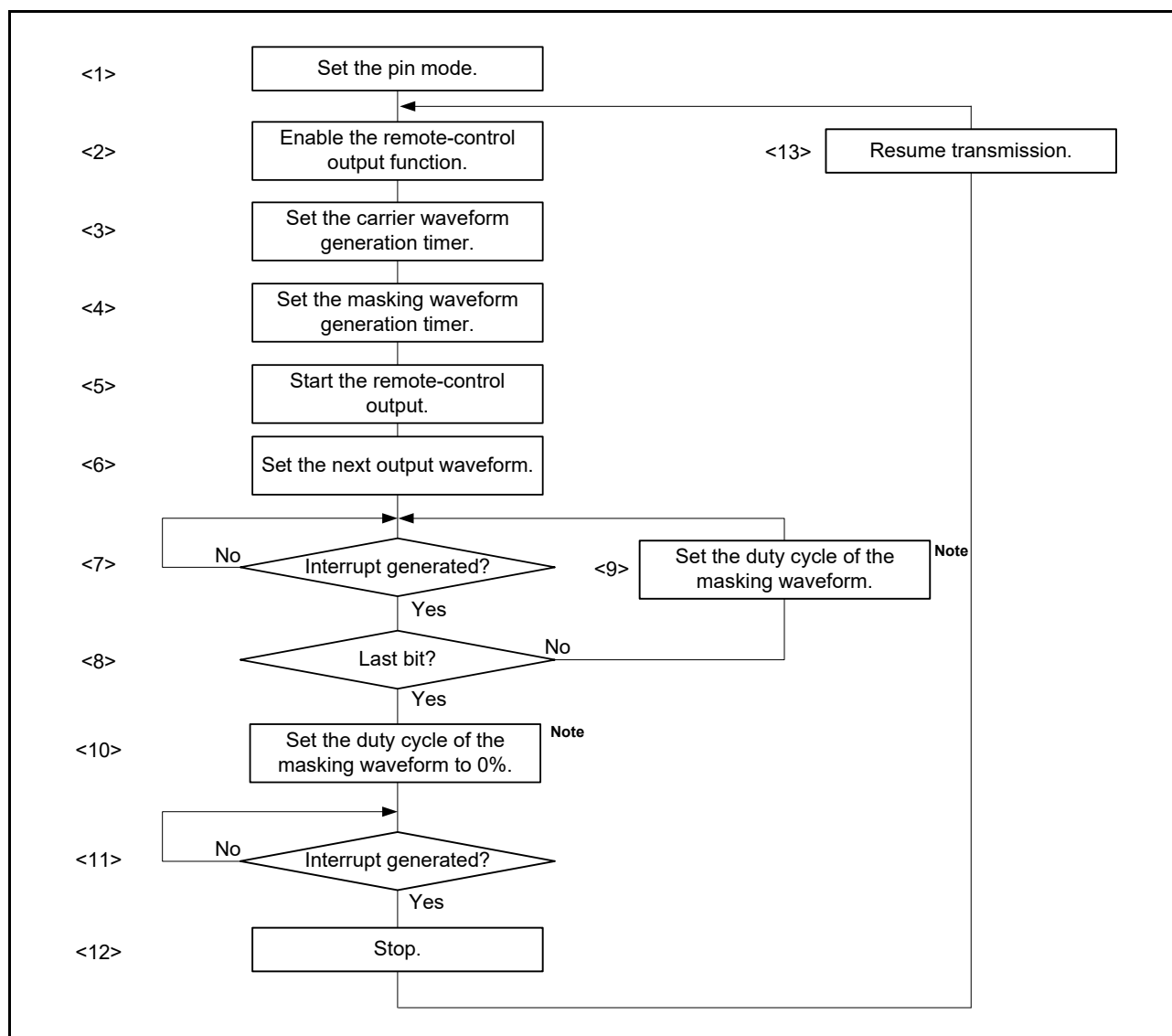


Figure 7 - 86 shows the procedure for setting the remote-control output.

Figure 7 - 86 Procedure for Setting the Remote-control Output Function



Note The value set is reflected from the rising edge of the masking waveform.
The period and width at high level of the masking waveform can only be set to integer multiples of the carrier period.

- Caution 1.** During the period from steps <3> to <11>, stopping supply of a clock signal to the timer array unit (by using the STOP command, etc.) and changes to the settings of registers other than TDR02 and TDR03 are prohibited.
- Caution 2.** During the period from steps <3> to <11>, use this module in normal operating mode or HALT mode.
- Caution 3.** Select the same operating clock for channels 2 to 5.
- Caution 4.** When stopping the remote-control output, be sure to do so in accord with steps <10> to <12>. Using a different procedure to stop it may lead to the errors listed below.
- A waveform other than one with the carrier frequency period is output.
 - After operation stops, the output is fixed to the high level.
- Caution 5.** For the remote-control output, channels 2 to 5 must be in the PWM output mode.

- <1> Set the pin mode.
Make the port-pin settings for the remote-control output pin (REMOOUT). See **7.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of timers.**
- <2> Set the TOS0 bit in the timer output select register (TOS) to 1.
- <3> Use the PWM functions of channel 4 (master) and channel 5 (slave) to set the carrier waveform.
The value of the TDR04 register = the period of the carrier waveform - 1
The value of the TDR05 register = the width at high level of the carrier waveform
- <4> Use the PWM functions of channel 2 (master) and channel 3 (slave) for generating the masking waveform.
Set the period and width at high level of the masking waveform (in header code).
The value of the TDR02 register = the period of the masking waveform - 1
The value of the TDR03 register = the width at high level of the masking waveform
- <5> Start output.
Set the TS02 to TS05 bits of timer channel start register 0 (TS0) to 1 at the same time.
- <6> Set the next output waveform.
The value of the TDR02 register = the period of the masking waveform - 1
The value of the TDR03 register = the width at high level of the masking waveform
- <7> Wait for the interrupt signal (INTTM02) generated on the rising edge of the masking waveform.
- <8> If this is to be the last code bit of the remote-control waveform for transmission, go to step <10>.
If it is not the last code bit, make the setting of the next masking waveform in step <9> and repeat steps <7> to <9> until the last code bit is reached.

<Continuous transmission>

- <9> Set the period and duty cycle (period: TDR02, width at high level: TDR03).

<Procedure for stopping>

- <10> Set the duty cycle of the masking waveform to 0% (set TDR03 to 0000H).
- <11> Wait for the interrupt signal (INTTM02) generated on the rising edge of the masking waveform.
- <12> Stop the timer operation.
Set the TT02 to TT05 bits of timer channel stop register 0 (TT0) to 1 at the same time, set the TOE02 to TOE05 bits of timer output enable register 0 (TOE0) to 0, and finally set the TO02 to TO05 bits of timer output register 0 (TO0) to 0.

<Procedure for resuming>

- <13> To resume transmission, set the TOE03 and TOE05 bits to 1 and repeat the steps from step <2>.
Overwriting of settings is not necessary.

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 ELCL

The timer outputs (TO00 to TO06) of channels 0 to 6 of timer array unit 0 can be used as event inputs for the logic and event link controller (ELCL). When the timer output signal is to be used for event input to the ELCL but not for output to the TO0m pin, set the corresponding bit in port function output enable register 0 (PFOE0) to 0. For details, see **4.3.14 Port function output enable registers (PFOEx)**.

Section 8 Timer RJ

RL78/L23 devices have one or two timer RJ channels. The number of channels mounted depends on the number of pins and capacity of the flash memory.

Number of Pins	44 to 64		80 and 100
Flash Memory	64 to 128 Kbytes	256 to 512 Kbytes	128 to 512 Kbytes
Number of channels	1	2	2

8.1 Functions of Timer RJn

Timer RJn is a 16-bit timer that can be used for pulse output, measuring the width or period of an externally input pulse signal, and counting external events.

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and they can be accessed by accessing the TRJn register.

Table 8 - 1 lists the timer RJn specifications.

Table 8 - 1 Timer RJn Specifications

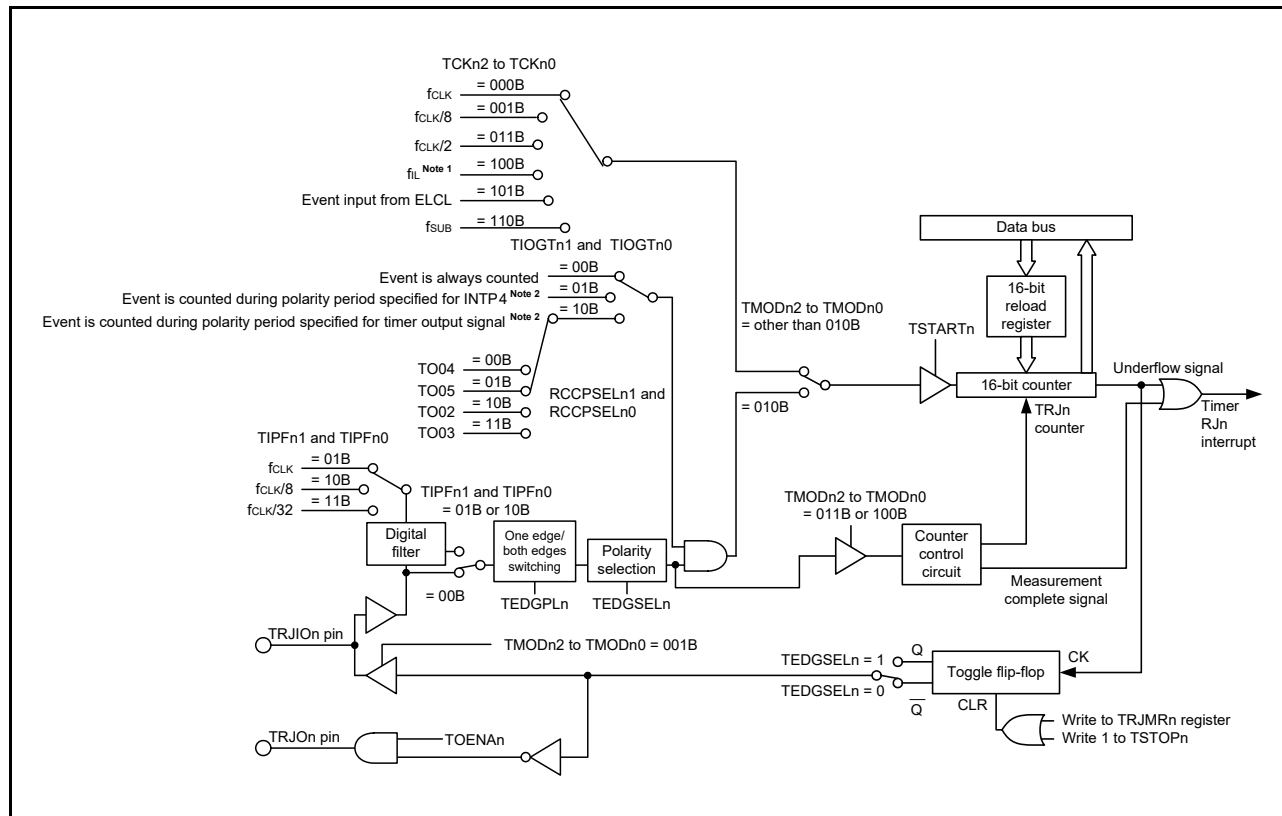
Item		Description
Operating modes	Timer mode	The count source is counted.
	Pulse output mode	The count source is counted and the output is inverted at each underflow of the timer.
	Event counter mode	An external event is counted. Operation is possible in STOP mode.
	Pulse width measurement mode	The width of an externally input pulse signal is measured.
	Pulse period measurement mode	The period of an externally input pulse signal is measured.
Count source (operating clock)		fCLK, fCLK/2, fCLK/8, fIL, fSUB, or event input from the logic and event link controller (ELCL) selectable
Interrupt		- When the counter underflows. - When the measurement of the active width of the external input (TRJIO_n) is completed in pulse width measurement mode. - When the set edge of the external input (TRJIO_n) is input in pulse period measurement mode.
Selectable functions		Coordination with the logic and event link controller (ELCL). Event input from the ELCL is selectable as a count source.

Remark n: Channel number (n = 0, 1)

8.2 Configuration of Timer RJn

Figure 8 - 1 shows the timer RJn block diagram and **Table 8 - 2** lists the timer RJn pin configuration.

Figure 8 - 1 Timer RJn Block Diagram



Note 1. Selecting fIL as the source for counting is prohibited while fIL is not oscillating.

Note 2. The polarity can be selected by the RCCPSELn2 bit in the TRJISRn register.

Remark TSTARTn, TSTOPn: Bits in TRJCRn register

TEDGSELn, TOENAn, TIPFn0, TIPFn1, TIOGTn0, TIOGTn1: Bits in TRJIOCn register

TMODn0 to TMODn2, TEDGPLn, TCKn0 to TCKn2: Bits in TRJMRn register

RCCPSELn0, RCCPSELn1: Bits in TRJISRn register

Table 8 - 2 Timer RJn Pin Configuration

Pin Name	I/O	Function
INTP4	Input	Event counter mode control for timer RJn
TRJIOOn	Input/output	External event input and pulse output for timer RJn
TRJOn	Output	Pulse output for timer RJn

Remark n: Channel number (n = 0, 1)

8.3 Registers for Controlling Timer RJ

The following registers are used to control timer RJ.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer RJ counter registers n (TRJn) (n = 0, 1)^{Note}
- Timer RJ control registers n (TRJCRn) (n = 0, 1)
- Timer RJ I/O control registers n (TRJIOCn) (n = 0, 1)
- Timer RJ mode registers n (TRJMRn) (n = 0, 1)
- Timer RJ event pin select registers n (TRJISRn) (n = 0, 1)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Note In the case of access to the TRJn register, the CPU does not automatically proceed to processing of the next instruction but enters a state of waiting for the completion access before further CPU processing. This wait state will increase the number of clock cycles for instruction execution by the number of clock cycles of waiting. The number of clock cycles of waiting for access to the TRJn register is one for both writing and reading.

Remark n: Channel number (n = 0, 1)

8.3.1 Peripheral enable register 2 (PER2)

The PER2 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 RJ0 is to be used, be sure to set bit 0 (TRJ0EN) to 1. If timer RJ1 is to be used, be sure to set bit 1 (TRJ1EN) to 1. The PER2 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 2 (PER2)

Address: F00FCH
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN	UTA2EN	UTA1EN	TRJ1EN	TRJ0EN
	Control of supply of an input clock to timer RJn							
	0	Stops supply of an input clock. • The SFRs used by timer RJn cannot be written. ^{Note}						
	1	Enables supply of an input clock. • The SFRs used by timer RJn can be read and written.						

Note When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution Be sure to set the TRJnEN bit to 1 before making the timer RJn settings. If TRJnEN = 0, writing to the registers for controlling timer RJn is ignored, and the value read is 00H. Note, however, that writing to the following registers is valid.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Remark n: Channel number (n = 0, 1)

8.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 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 0 (TRJ0RES) and 1 (TRJ1RES) to 1 to reset timer RJ0 and timer RJ1, respectively. The PRR2 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 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
PRR2	EXSDRES	OSDCRES	TKBRES	0	0	0	TRJ1RES	TRJ0RES
	TRJnRES	Control resetting of timer RJn						
	0	Timer RJn is released from the reset state.						
	1	Timer RJn is in the reset state. • The SFRs for use with timer RJn are initialized.						

Caution Be sure to clear bits 4 to 2 to 0.

Remark n: Channel number (n = 0, 1)

8.3.3 Timer RJ counter registers n (TRJn) (n = 0, 1)

Each TRJn is a 16-bit register. The write value is written to the reload register and the read value is read from the counter. The states of the reload register and the counter change depending on the TSTARTn bit in the TRJCRn register. For details, see **8.4.1 Reload register and counter rewrite operation**. The TRJn registers can be set by a 16-bit memory manipulation instruction. The value of each TRJn register following a reset is FFFFH.

Figure 8 - 4 Format of Timer RJ Counter Registers n (TRJn)

Address: F0598H (TRJ0), F059AH (TRJ1)

After reset: FFFFH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRJn																
—		Function													Setting Range	
Bits 15 to 0		16-bit counter ^{Notes 1, 2}													0000H to FFFFH	

Note 1. When 1 is written to the TSTOPn bit in the TRJCRn register, the 16-bit counter is forcibly stopped and set to FFFFH.

Note 2. When the setting of bits TCKn2 to TCKn0 in the TRJMRn register is other than 001B (fCLK/8) or 011B (fCLK/2), if the TRJn register is set to 0000H, a request signal to the DTC and the ELCL is generated only once immediately after counting starts. However, the TRJOn and TRJIOn output is toggled.

When the TRJn register is set to 0000H in event counter mode, regardless of the value of bits TCKn2 to TCKn0, a request signal to the DTC and the ELCL is generated only once immediately after counting starts. In addition, the TRJOn output is toggled even during a period other than the specified count period.

When the TRJn register is set to 0001H or a higher value, a request signal is generated each time TRJn underflows.

Caution In the case of access to the TRJn register, the CPU does not automatically proceed to processing of the next instruction but enters a state of waiting for the completion access before further CPU processing. This wait state will increase the number of clock cycles for instruction execution by the number of clock cycles of waiting. The number of clock cycles of waiting for access to the TRJn register is one for both writing and reading.

Remark n: Channel number (n = 0, 1)

8.3.4 Timer RJ control registers n (TRJCRn) (n = 0, 1)

The TRJCRn registers start or stop counting by timer RJn and indicate the state of timer RJn. The TRJCRn registers can be set by an 8-bit memory manipulation instruction. The value of each TRJCRn register following a reset is 00H.

Figure 8 - 5 Format of Timer RJ Control Registers n (TRJCRn) (1/2)

Address: F0240H (TRJCR0), F0244H (TRJCR1)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJCRn	0	0	TUNDFn	TEDGFn	0	TSTOPn	TCSTFn	TSTARTn

TUNDFn	Timer RJn underflow flag
0	No underflow
1	Underflow
[Condition for setting to 0] • When 0 is written to this bit by a program. [Condition for setting to 1] • When the counter underflows.	

TEDGFn	Active edge judgement flag
0	No active edge received
1	Active edge received
[Condition for setting to 0] • When 0 is written to this bit by a program. [Conditions for setting to 1] • When the measurement of the active width of the external input (TRJION) is completed in pulse width measurement mode. • The set edge of the external input (TRJION) is input in pulse period measurement mode.	

TSTOPn	Timer RJn counter forced stop ^{Note 1}
Writing 1 to this bit forcibly stops counting. The read value is 0.	

TCSTFn	Timer RJn counter status flag ^{Note 2}
0	Counting stops.
1	Counting is in progress.
[Conditions for setting to 0] • When 0 is written to the TSTARTn bit (the TCSTF bit is set to 0 in synchronization with the count source). • When 1 is written to the TSTOPn bit. [Condition for setting to 1] • When 1 is written to the TSTARTn bit (the TCSTF bit is set to 1 in synchronization with the count source).	

Figure 8 - 5 Format of Timer RJ Control Registers n (TRJCRn) (2/2)

TSTARTn	Timer RJn counter start ^{Note 2}
0	Counting stops.
1	Counting starts.
Writing 1 to the TSTARTn bit starts counting and writing 0 stops counting. When the TSTARTn bit is set to 1 (to start counting), the TCSTFn bit is set to 1 (to indicate that counting is in progress) in synchronization with the count source. Also, after 0 is written to the TSTARTn bit, the TCSTFn bit is set to 0 (counting is stopped) in synchronization with the count source. For details, see 8.5.1 Control over starting and stopping of the counter .	

Note 1. Writing 1 to the TSTOPn bit (to forcibly stop counting) initializes bits TSTARTn and TCSTFn at the same time. The pulse output level is also initialized.

Note 2. For notes on using bits TSTARTn and TCSTFn, see **8.5.1 Control over starting and stopping of the counter**.

Remark n: Channel number (n = 0, 1)

8.3.5 Timer RJ I/O control registers n (TRJIOCn) (n = 0, 1)

The TRJIOCn registers set the input/output of timer RJn. The TRJIOCn registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each TRJIOCn register following a reset is 00H.

Figure 8 - 6 Format of Timer RJ I/O Control Registers n (TRJIOCn)

Address: F0241H (TRJIOC0), F0245H (TRJIOC1)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJIOCn	TIOGTn1	TIOGTn0	TIPFn1	TIPFn0	0	TOENAn	0	TEDGSELn

TIOGTn1	TIOGTn0	TRJOn counting control ^{Notes 1, 2}
0	0	Event is always counted
0	1	Event is counted during polarity period specified for INTP4
1	0	Event is counted during polarity period specified for timer output signal
Other than above		Setting prohibited

TIPFn1	TIPFn0	TRJOn input filter select
0	0	No filter
0	1	Filter sampled at fCLK
1	0	Filter sampled at fCLK/8
1	1	Filter sampled at fCLK/32
These bits are used to specify the sampling frequency of the filter for the TRJOn input. If the input to the TRJOn pin is sampled and the value matches three successive times, that value is taken as the input value.		

TOENAn	TRJOn output enable
0	TRJOn output disabled (port)
1	TRJOn output enabled

TEDGSELn	I/O polarity switch
Function varies depending on the operating mode (see Tables 8 - 3 and 8 - 4).	

Note 1. When INTP4 or the timer output signal is used, the polarity to count an event can be selected by the RCCPSELn2 bit in the TRJISRn register.

Note 2. Bits TIOGTn0 and TIOGTn1 are enabled only in event counter mode.

Remark n: Channel number (n = 0, 1)

Table 8 - 3 TRJOn I/O Edge and Polarity Switching

Operating Mode	Function
Timer mode	Not used (I/O port)
Pulse output mode	0: Output starts at the high level. (Initialization level: high) 1: Output starts at the low level. (Initialization level: low)
Event counter mode	0: Count at rising edge. 1: Count at falling edge.
Pulse width measurement mode	0: Low-level width is measured. 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge. 1: Measure from one falling edge to the next falling edge.

Remark n: Channel number (n = 0, 1)

Table 8 - 4 TRJOn Output Polarity Switching

Operating Mode	Function
All modes	0: Output starts at the low level. (Initialization level: low) 1: Output starts at the high level. (Initialization level: high)

Remark n: Channel number (n = 0, 1)

8.3.6 Timer RJ mode registers n (TRJMRn) (n = 0, 1)

The TRJMRn registers set the operating mode of timer RJn. The TRJMRn registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each TRJMRn register following a reset is 00H.

Figure 8 - 7 Format of Timer RJ Mode Registers n (TRJMRn)

Address: F0242H (TRJMR0), F0246H (TRJMR1)

After reset: 0000H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJMRn	0	TCKn2	TCKn1	TCKn0	TEDGPLn	TMODn2	TMODn1	TMODn0

TCKn2	TCKn1	TCKn0	Timer RJn count source select ^{Notes 1, 2}
0	0	0	fCLK
0	0	1	fCLK/8
0	1	1	fCLK/2
1	0	0	fIL ^{Note 3}
1	0	1	Event input from the ELCL
1	1	0	fSUB
Other than above			Setting prohibited

TEDGPLn	TRJOn edge polarity select ^{Note 4}
0	One edge
1	Both edges

TMODn2	TMODn1	TMODn0	Timer RJn operating mode select ^{Note 5}
0	0	0	Timer mode
0	0	1	Pulse output mode
0	1	0	Event counter mode
0	1	1	Pulse width measurement mode
1	0	0	Pulse period measurement mode
Other than above			Setting prohibited

Note 1. When event counter mode is selected, the external input (TRJOn) is selected as the count source regardless of the setting of bits TCKn0 to TCKn2.

Note 2. Do not switch count sources during counting. Count sources should be switched when both the TSTARTn and TCSTFn bits in the TRJCRn register are set to 0 (counting is stopped).

Note 3. Selecting fIL as the source for counting is prohibited while fIL is not oscillating.

Note 4. The TEDGPLn bit is enabled only in event counter mode.

Note 5. The operating mode can be changed only when counting is stopped while both the bits TSTARTn and TCSTFn in the TRJCRn register are set to 0 (counting is stopped). Do not change the operating mode during counting.

Caution Write access to the TRJMRn register initializes the output from pins TRJOn and TRJOn of timer RJn. For details on the output level at initialization, refer to Table 8 - 3 TRJOn I/O Edge and Polarity Switching and Table 8 - 4 TRJOn Output Polarity Switching.

Remark n: Channel number (n = 0, 1)

8.3.7 Timer RJ event pin select registers n (TRJISRn) (n = 0, 1)

The TRJISRn registers select the timer for controlling the event count period and set the polarity in event counter mode. The TRJISRn registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each TRJISRn register following a reset is 00H.

Figure 8 - 8 Format of Timer RJ Event Pin Select Registers n (TRJISRn)

Address: F0243H (TRJISR0), F0247H (TRJISR1)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TRJISRn	0	0	0	0	0	RCCPSELn2	RCCPSELn1	RCCPSELn0

RCCPSELn2	Timer output signal and INTP4 polarity selection	
Note		
0	An event is counted during the low-level period	
1	An event is counted during the high-level period	

RCCPSELn1	RCCPSELn0	Timer output signal selection
Note	Note	
0	0	TO04
0	1	TO05
1	0	TO02
1	1	TO03

Note Bits RCCPSELn0 to RCCPSELn2 are enabled only in event counter mode.

Remark n: Channel number (n = 0, 1)

8.3.8 Registers for controlling the port functions multiplexed with the inputs and outputs of timer RJ

Set the following registers for controlling the port functions multiplexed with inputs and outputs of timer RJ.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

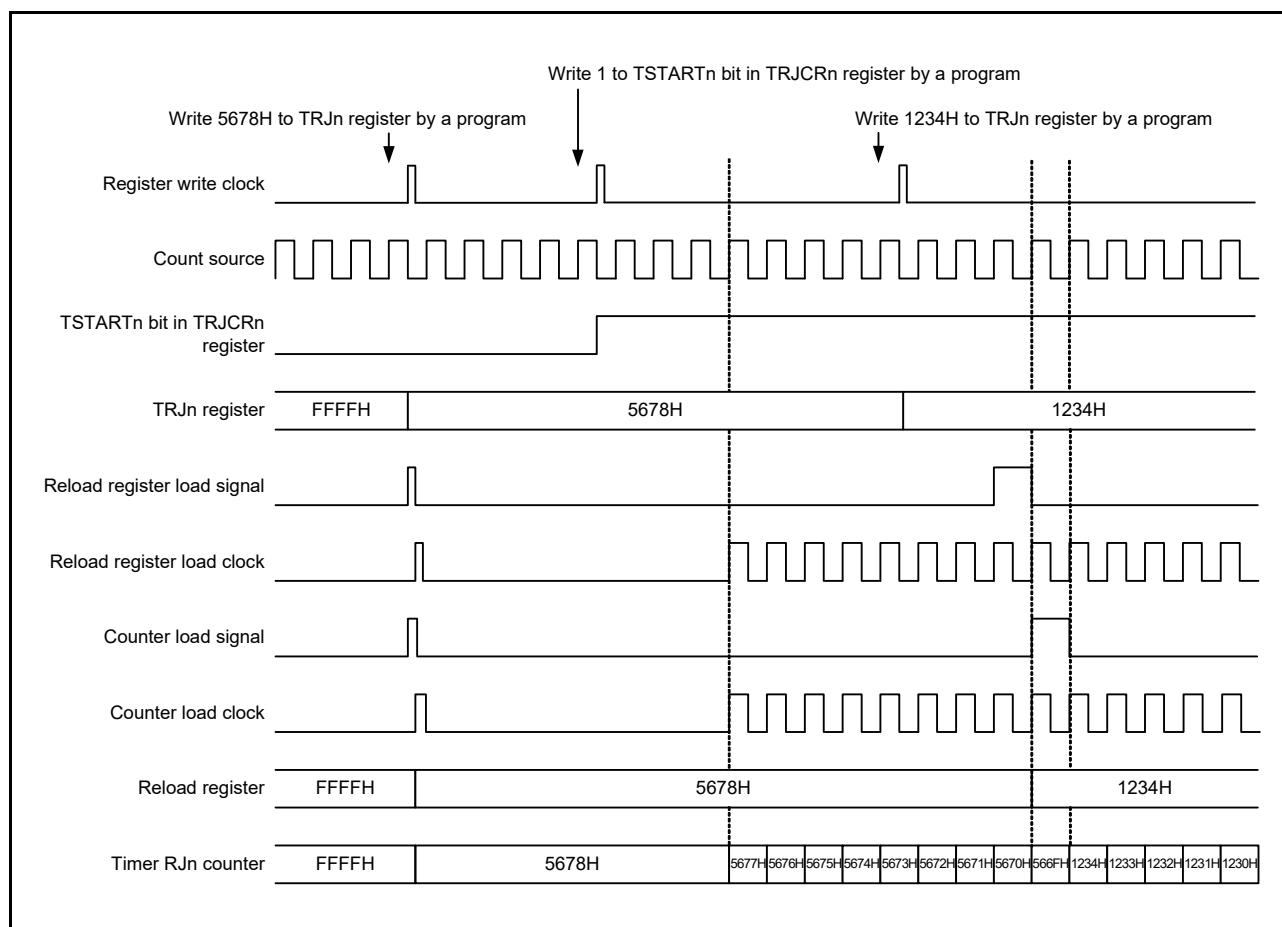
8.4 Timer RJn Operation

8.4.1 Reload register and counter rewrite operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value in the TSTARTn bit in the TRJCRn register. When the TSTARTn bit is 0 (counting is stopped), the count value is directly written to the reload register and the counter. When the TSTARTn bit is 1 (to start counting), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source.

Figure 8 - 9 shows the timing of rewrite operation with TSTARTn bit value.

Figure 8 - 9 Timing of Rewrite Operation with TSTARTn Bit Value



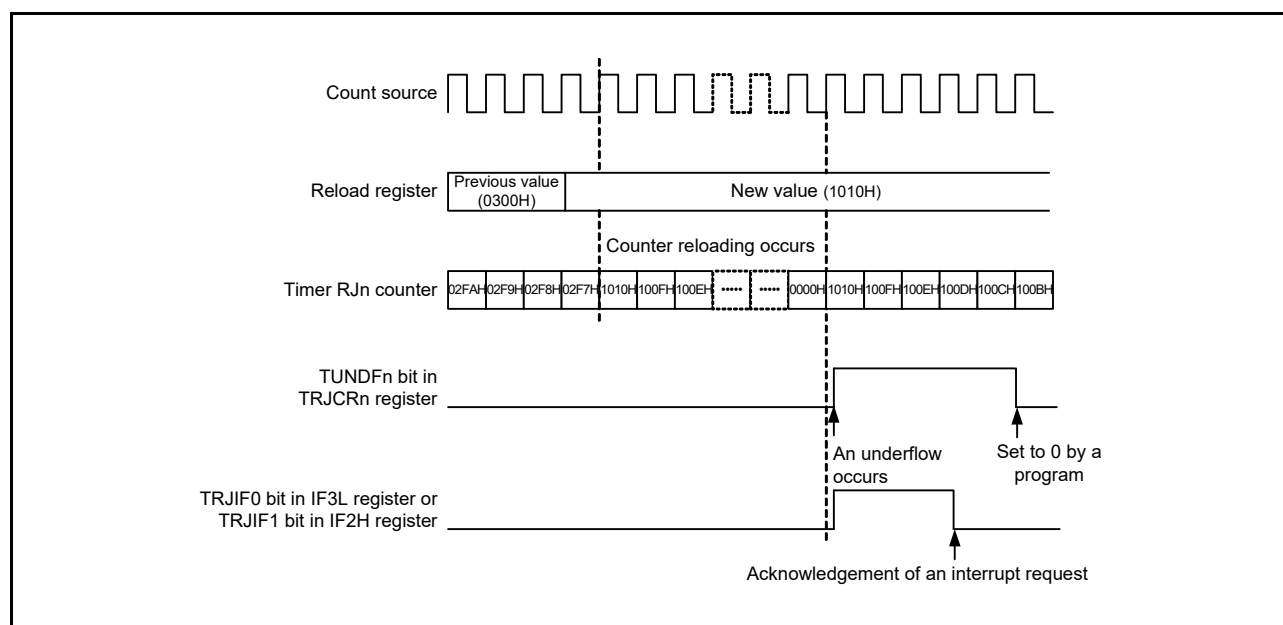
Remark n: Channel number (n = 0, 1)

8.4.2 Timer mode

In this mode, the counter is decremented by the count source selected by bits TCKn0 to TCKn2 in the TRJMRn register. In timer mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 8 - 10 shows the operation example in timer mode.

Figure 8 - 10 Operation Example in Timer Mode



Remark n: Channel number (n = 0, 1)

8.4.3 Pulse output mode

In this mode, the counter is decremented by the count source selected by bits TCKn0 to TCKn2 in the TRJMRn register, and the output level of pins TRJOn and TRJOn is inverted each time an underflow occurs.

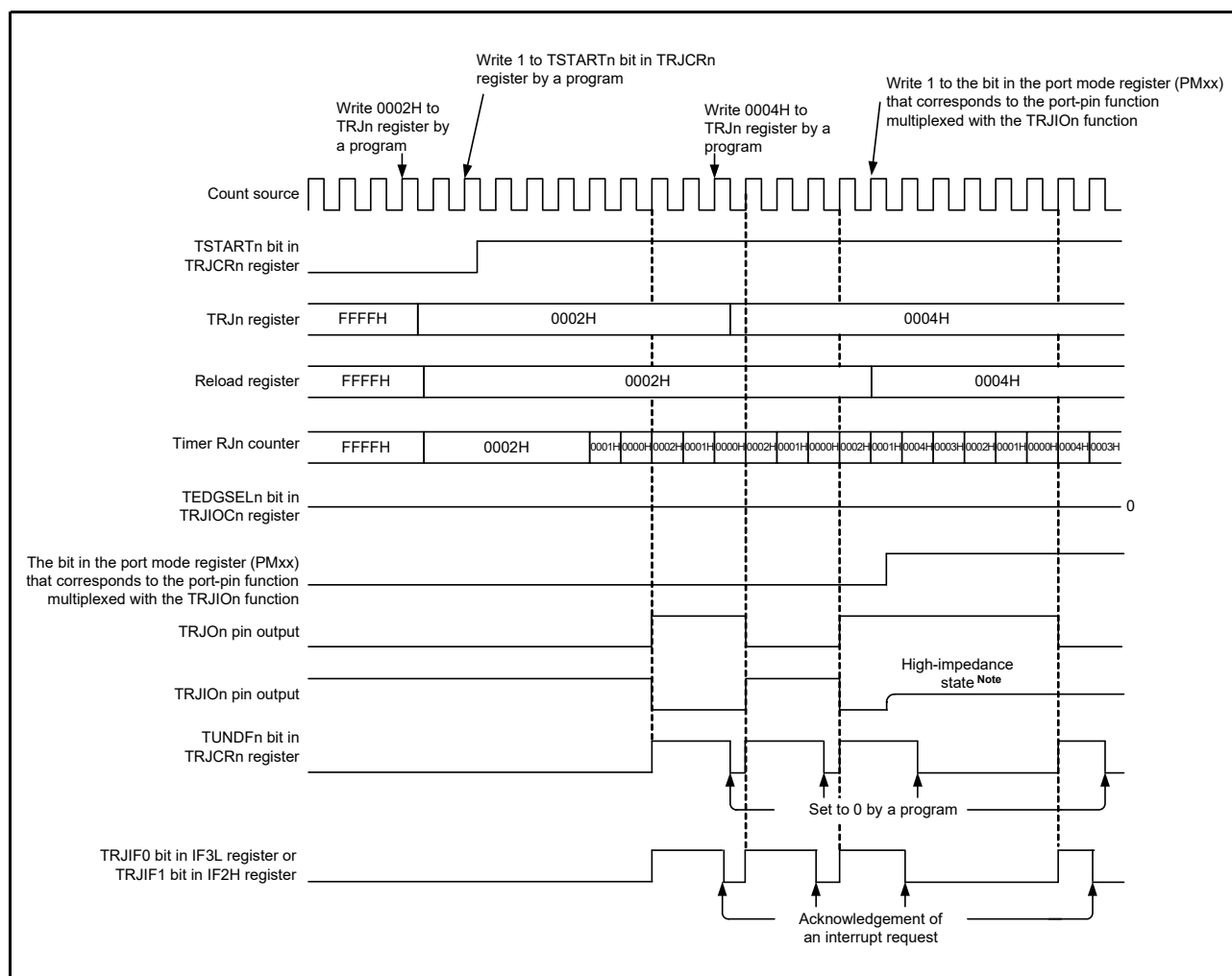
In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

In addition, a pulse can be output from pins TRJOn and TRJOn. The output level is inverted each time an underflow occurs. The pulse output from the TRJOn pin can be stopped by the TOENAn bit in the TRJIOCn register.

Also, the output level can be selected by the TEDGSELn bit in the TRJIOCn register.

Figure 8 - 11 shows the operation example in pulse output mode.

Figure 8 - 11 Operation Example in Pulse Output Mode



Note The TRJOn pin becomes high impedance by output enable control on the port selected as the TRJOn function.

Remark n: Channel number (n = 0, 1)

8.4.4 Event counter mode

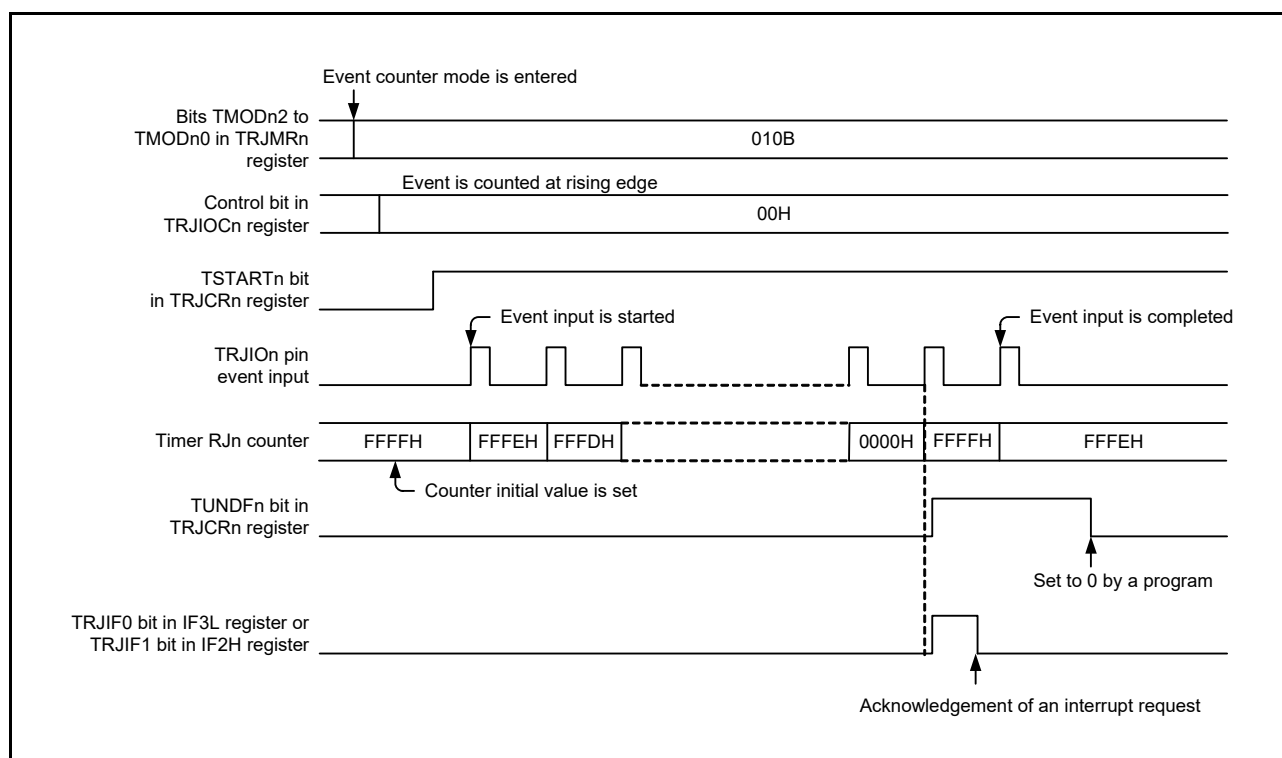
In this mode, the counter is decremented by an external event signal (count source) input to the TRJOn pin. Various periods for counting events can be set by bits TIOGTn0 and TIOGTn1 in the TRJIOcN register and the TRJISRn register. In addition, the filter function for the TRJOn input can be specified by bits TIPFn0 and TIPFn1 in the TRJIOcN register.

Also, the output from the TRJOn pin can be toggled even in event counter mode.

When event counter mode is used, see **8.5.5 Procedure for setting pins TRJOn and TRJIOcN**.

Figure 8 - 12 shows the operation example 1 in event counter mode.

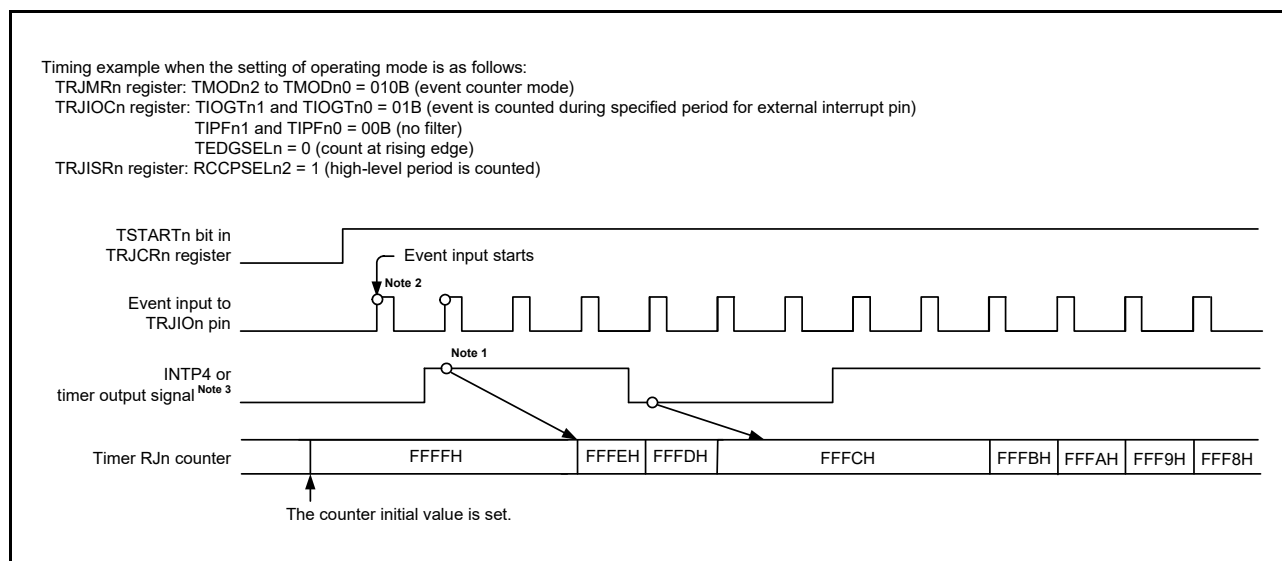
Figure 8 - 12 Operation Example 1 in Event Counter Mode



Remark n: Channel number (n = 0, 1)

Figure 8 - 13 shows an operation example for counting during the specified period in event counter mode (bits TIOGTn1 and TIOGTn0 in the TRJIOCn register are set to 01B or 10B).

Figure 8 - 13 Operation Example 2 in Event Counter Mode



The following notes apply only when bits TIOGTn1 and TIOGTn0 in the TRJIOCn register are 01B or 10B for the setting of operating mode in event counter mode.

Note 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.

Note 2. Counting of two cycles of the source for counting may proceed immediately after counting is started, depending on the state before counting was previously stopped. To disable this counting of two cycles immediately after counting is started, write 1 to the TSTOPn bit in the TRJCRn register to initialize the internal circuit, and then make the settings for operation before starting the counter.

Note 3. For the timer output signal selected by the RCCPSELn1 and RCCPSELn0 bits in the TRJISRn register, the pin assigned to the timer output function cannot be used as the output of any multiplexed function other than the timer.

Remark n: Channel number (n = 0, 1)

8.4.5 Pulse width measurement mode

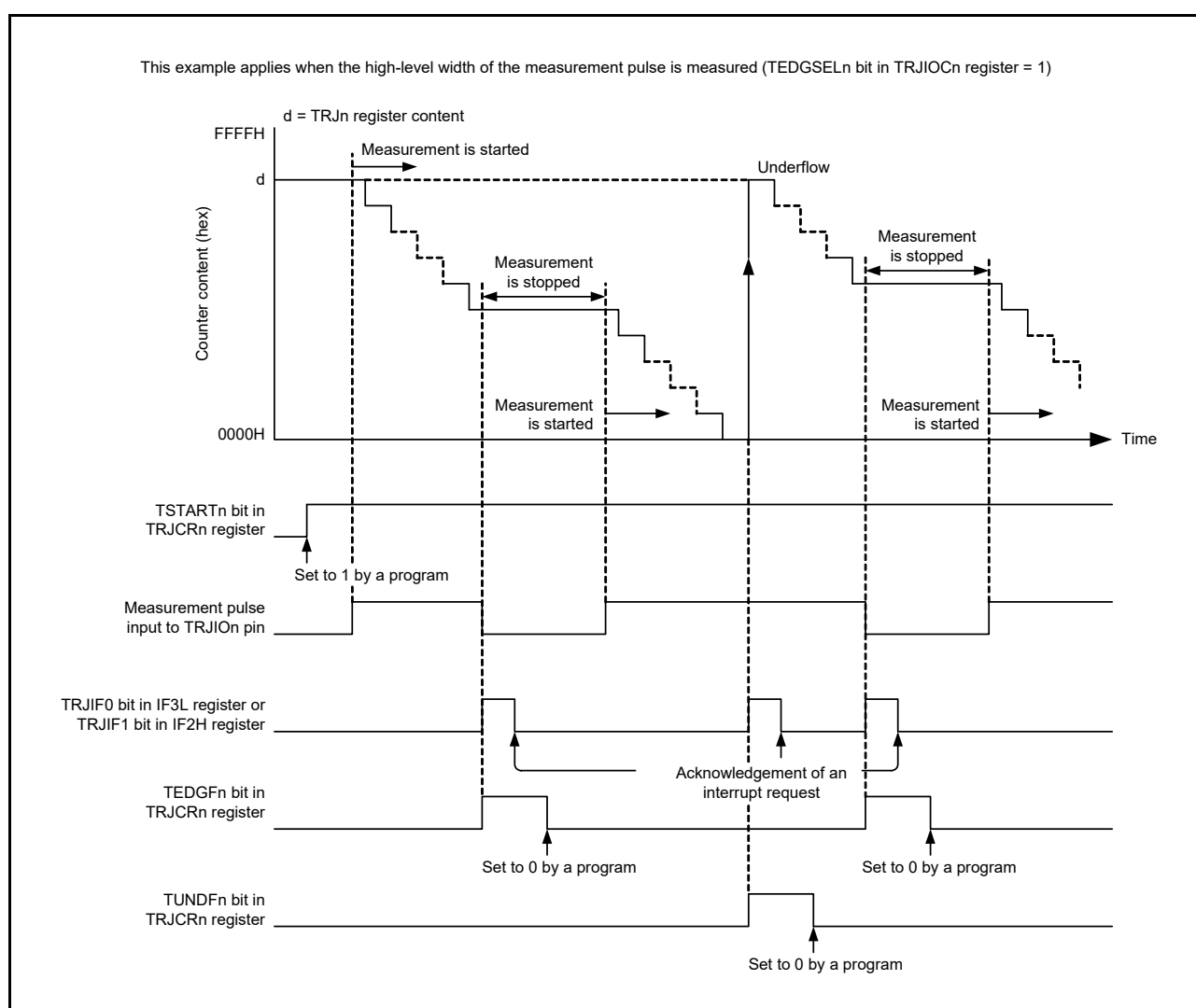
In this mode, the pulse width of an external signal input to the TRJION pin is measured.

When the level specified by the TEDGSELn bit in the TRJIOCn register is input to the TRJION pin, the decrement is started with the selected count source. When the specified level on the TRJION pin ends, the counter is stopped, the TEDGFn bit in the TRJCRn register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDFn bit in the TRJCRn register is set to 1 (underflow) and an interrupt request is generated.

Figure 8 - 14 shows the operation example in pulse width measurement mode.

When accessing bits TEDGFn and TUNDFn in the TRJCRn register, see **8.5.2 Access to flags (bits TEDGFn and TUNDFn in TRJCRn register)**.

Figure 8 - 14 Operation Example in Pulse Width Measurement Mode



Remark n: Channel number (n = 0, 1)

8.4.6 Pulse period measurement mode

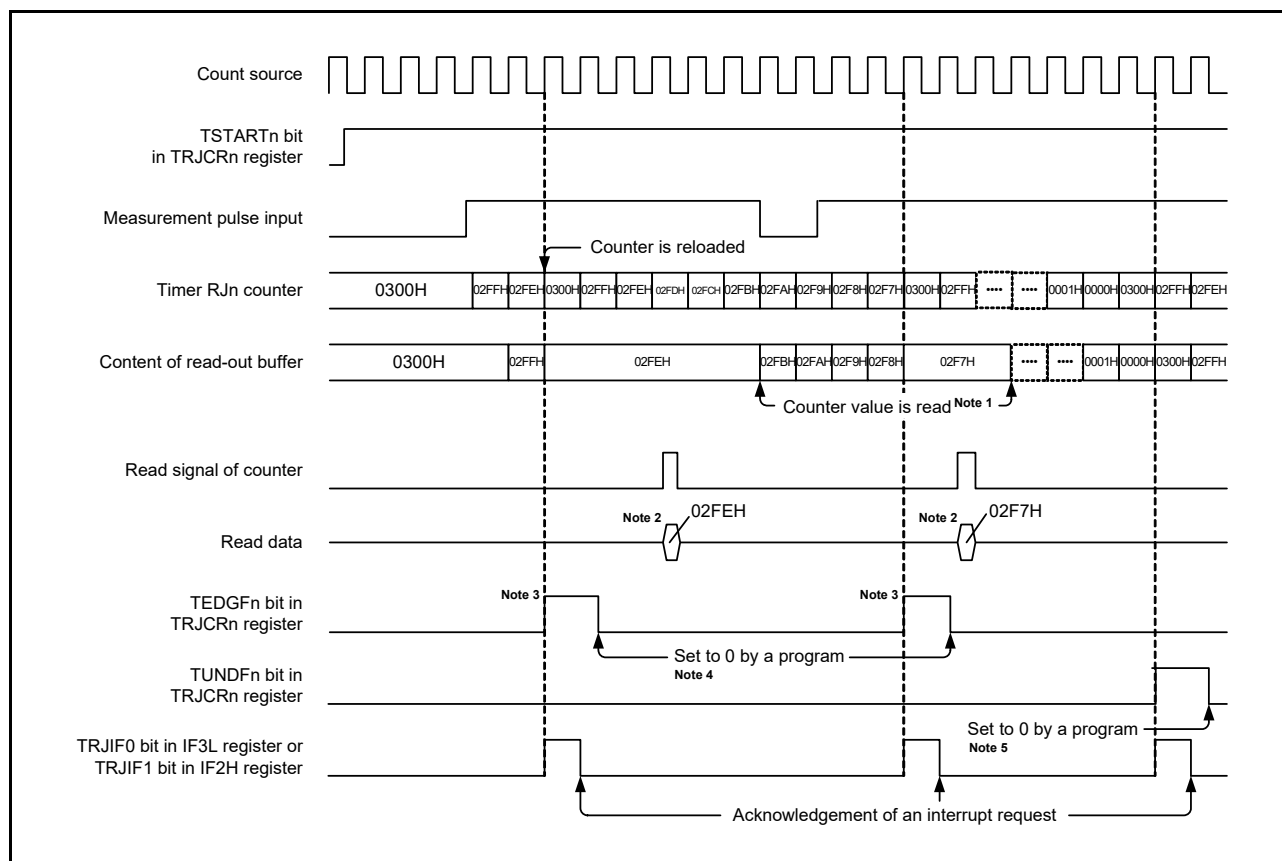
In this mode, the pulse period of an external signal input to the TRJIO pin is measured.

The counter is decremented by the count source selected by bits TCKn0 to TCKn2 in the TRJMRn register. When a pulse with the period specified by the TEDGSELn bit in the TRJIOCn register is input to the TRJIO pin, the count value is transferred to the read-out buffer at the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGFn bit in the TRJCRn register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (TRJn register) is read at this time and the difference from the reload value is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDFn bit in the TRJCRn register is set to 1 (underflow) and an interrupt request is generated.

Figure 8 - 15 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source. Also, the low-level and high-level widths must be both longer than the period of the count source. If a pulse period shorter than these conditions is input, the input may be ignored.

Figure 8 - 15 Operation Example in Pulse Period Measurement Mode



This example applies when the initial value of the TRJn register is set to 0300H, the TEDGSELn bit in the TRJIOCn register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the TRJn register must be performed during the period from when the TEDGFn bit is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the TRJn register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.

Note 2. When the TRJn register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGFn bit in the TRJCRn register is set to 1 (active edge received).

Note 4. To set to 0 by a program, write 0 to the TEDGFn bit in the TRJCRn register using an 8-bit memory manipulation instruction.

Note 5. To set to 0 by a program, write 0 to the TUNDFn bit in the TRJCRn register using an 8-bit memory manipulation instruction.

Remark n: Channel number (n = 0, 1)

8.4.7 Coordination with the logic and event link controller (ELCL)

Through interlinking with the ELCL, event input from the ELCL can be set as the source for counting. Counting is of rising edges of the event input from the ELCL when this is selected as the source for counting by the TCKn0 to TCKn2 bits of the TRJMRn register. However, ELCL input does not function in event counter mode.

The ELCL setting procedure is shown below:

- Procedure for starting operation
 - <1> Make the settings of the ELCL.**Note**
 - <2> Set the operating mode for the event generation source.
 - <3> Set the mode for timer RJn.
 - <4> Start counting by timer RJn.
 - <5> Start the operation of the event generation source.
- Procedure for stopping operation
 - <1> Stop the operation of the event generation source.
 - <2> Stop counting by timer RJn.
 - <3> Make the settings of the ELCL.**Note**

Note See **24.5 Procedure of Setting the ELCL**.

Remark n: Channel number (n = 0, 1)

8.4.8 Output settings for each mode

Table 8 - 5 and **Table 8 - 6** list the states of pins TRJOn and TRJOn in each mode.

Table 8 - 5 TRJOn Pin Setting

Operating Mode	TRJIOCn Register		TRJOn Pin Output
	TOENAn Bit	TEDGSELn Bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 8 - 6 TRJOn Pin Setting

Operating Mode	TRJIOCn Register		TRJOn Pin I/O
	PMxx Bit ^{Note}	TEDGSELn Bit	
Timer mode	0 or 1	0 or 1	Input (Not used)
Pulse output mode	1	0 or 1	Output disabled (Hi-Z output)
	0	1	Normal output
		0	Inverted output
Event counter mode	1	0 or 1	Input
Pulse width measurement mode			
Pulse period measurement mode			

Note This refers to the bits in port mode registers (PMxx) that correspond to port-pin functions multiplexed with the TRJOn function.

Remark n: Channel number (n = 0, 1)

8.5 Points for Caution when Timer RJn is to be Used

8.5.1 Control over starting and stopping of the counter

- When event counter mode is set or the count source is set to other than the ELCL

After 1 is written to the TSTARTn bit in the TRJCRn register (to start counting) while counting is stopped, the TCSTFn bit in the TRJCRn register remains 0 (counting is stopped) for three cycles of the count source. Do not access the registers associated with timer RJn^{Note} other than the TCSTFn bit until this bit is set to 1 (counting is in progress).

After 0 is written to the TSTARTn bit (to stop counting) during counting, the TCSTFn bit remains 1 for three cycles of the count source. When the setting of the TCSTFn bit becomes 0, counting has stopped. Do not access the registers associated with timer RJn^{Note} other than the TCSTFn bit until this bit is set to 0.

Clear the interrupt register before changing the TSTARTn bit from 0 to 1. See **Section 25 Interrupt Functions** for details.

- When event counter mode is set or the count source is set to the ELCL

After 1 is written to the TSTARTn bit in the TRJCRn register (to start counting) while counting is stopped, the TCSTFn bit in the TRJCRn register remains 0 (counting is stopped) for two cycles of the CPU clock. Do not access the registers associated with timer RJn^{Note} other than the TCSTFn bit until this bit is set to 1 (counting is in progress).

After 0 is written to the TSTARTn bit (to stop counting) during counting, the TCSTFn bit remains 1 for two cycles of the CPU clock. When the setting of the TCSTFn bit becomes 0, counting has stopped. Do not access the registers associated with timer RJn^{Note} other than the TCSTFn bit until this bit is set to 0.

Clear the interrupt register before changing the TSTARTn bit from 0 to 1. See **Section 25 Interrupt Functions** for details.

Note Registers associated with timer RJn: TRJn, TRJCRn, TRJIOCn, TRJMRn, and TRJISRn

8.5.2 Access to flags (bits TEDGFn and TUNDFn in TRJCRn register)

Bits TEDGFn and TUNDFn in the TRJCRn register are set to 0 by writing 0 by a program, but writing 1 to these bits has no effect. If a read-modify-write instruction is used to set the TRJCRn register, bits TEDGFn and TUNDFn may be erroneously set to 0 depending on the timing, even when the TEDGFn bit is set to 1 (active edge received) and the TUNDFn bit is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCRn register.

8.5.3 Access to counter register

When the TSTARTn bit in the TRJCRn register is 1 (to have started counting) or the TCSTFn bit in the TRJCRn register is 1 (counting is in progress), allow at least three cycles of the source clock for counting between write operations in the case of successive writing to the TRJn register.

Remark n: Channel number (n = 0, 1)

8.5.4 When changing mode

The registers associated with timer RJn operating mode (TRJIOCn, TRJMRn, and TRJISRn) can be changed only when counting is stopped with both the TSTARTn and TCSTFn bits set to 0 (counting is stopped). Do not change these registers during counting.

When the registers associated with timer RJn operating mode are changed, the values of bits TEDGFn and TUNDFn are undefined. Write 0 (no active edge received) to the TEDGFn bit and 0 (no underflow) to the TUNDFn bit before starting the counter.

8.5.5 Procedure for setting pins TRJOn and TRJIOOn

After a reset, the I/O ports multiplexed with pins TRJOn and TRJIOOn function as input ports.

To output from pins TRJOn and TRJIOOn, use the following setting procedure:

- <1> Set the mode.
- <2> Set the initial value/output enabled.
- <3> Set the bits in port registers that correspond to the TRJOn and TRJIOOn pin functions to 0.
- <4> Set the bits in port mode registers that correspond to the TRJOn and TRJIOOn pin functions to output mode.
(Output is started from pins TRJOn and TRJIOOn)
- <5> Start the counter (TSTARTn in TRJCRn register = 1).

To input from the TRJIOOn pin, use the following setting procedure:

- <1> Set the mode.
- <2> Set the initial value/edge selected.
- <3> Set the bit in the port mode register that corresponds to the TRJIOOn pin function to input mode.
(Input is started from the TRJIOOn pin)
- <4> Start the counter (TSTARTn in TRJMRn register = 1).
- <5> Wait until the TCSTFn bit in the TRJCRn register is set to 1 (counting is in progress).
(In event counter mode only)
- <6> Input an external event from the TRJIOOn pin.
- <7> The processing on completion of the first measurement is invalid (the measured value is valid for the second and subsequent times). (In pulse width measurement mode and pulse period measurement mode only)

8.5.6 When timer RJn is not used

When timer RJn is not used, set bits TMODn2 to TMODn0 in the TRJMRn register to 000B (timer mode) and set the TOENAn bit in the TRJIOCn register to 0 (TRJOn output disabled).

8.5.7 When timer RJn operating clock is stopped

Supplying or stopping the timer RJn clock can be controlled by the TRJnEN bit in the PER2 register. Note that the following SFRs cannot be accessed while the timer RJn clock is stopped. Make sure the timer RJn clock is supplied before accessing any of these registers.

Registers TRJn, TRJCRn, TRJMRn, TRJIOCn, and TRJISRn.

Remark n: Channel number (n = 0, 1)

8.5.8 Procedures for enabling and disabling operation in the event counter mode while the CPU is in the STOP mode

To enable operation in the event counter mode while in the STOP mode, follow the procedure below before placing the CPU in the STOP mode.

- <1> Start supply of the operating clock to the timer RJ.
- <2> Set the operating mode of the timer RJ.
- <3> Start the counter (TSTARTn = 1, TCSTFn = 1).
- <4> Stop supply of the operating clock to the timer RJ, and then place the CPU in the STOP mode.

To disable operation in the event counter mode while in the STOP mode, follow the procedure below.

- <1> Start supply of the operating clock to the timer RJ.
- <2> Stop the counter (TSTARTn = 0, TCSTFn = 0)
- <3> Stop supply of the operating clock to the timer RJ.

8.5.9 Functional restriction in STOP mode (event counter mode only)

When operation in event counter mode is proceeding while in STOP mode, the digital filter function cannot be used.

8.5.10 When counting is forcibly stopped by the TSTOPn bit

After the counter is forcibly stopped by the TSTOPn bit in the TRJCRn register, wait for at least one cycle of the source for counting before access to the following SFRs.

Registers TRJn, TRJCRn, and TRJMRn

8.5.11 Digital filter

When the digital filter is to be used, wait for at least five cycles of the digital filter clock after having set the TIPFn1 and TIPFn0 bits in the TRJIOCn register. Also, wait for at least five cycles of the digital filter clock after having changed the setting of the TEDGSELn bit in the TRJIOCn register while the digital filter is being used.

8.5.12 When selecting fIL as the source for counting

Selecting fIL as the source for counting is prohibited while fIL is not oscillating.

Remark n: Channel number (n = 0, 1)

Section 9 16-bit Timers KB40, KB41, and KB42

16-bit timers KB40, KB41, and KB42 can generate PWM signals suitable for controlling power supplies, lighting, and inductive heating (IH) cookers. The number of 16-bit timers depends on the product.

Number of Pins	44 to 64		80 and 100
Flash Memory	64 to 128 Kbytes	256 to 512 Kbytes	128 to 512 Kbytes
16-bit timer KB40	✓	✓	✓
16-bit timer KB41	—	✓	✓
16-bit timer KB42	—	✓	✓

Caution Most of the subsequent descriptions in this section take the 100-pin products as an example.

9.1 Functions of 16-bit Timers KB40, KB41, and KB42

The 16-bit timers KB40, KB41, and KB42 are dedicated PWM output timers with two output signals each, enabling the generation of up to six PWM signals. Complementary PWM signals can also be generated to control a half-bridge circuit with two output signals, full-bridge circuit with four output signals, or 3-phase inverter circuit with six output signals. Furthermore, through interlocking with a comparator output signal or INTP0 to INTP7, the output of PWM signals can be forcibly stopped and the timers restarted. Using these functions enables the configuration of power factor correction (PFC) circuits and DC/DC converters. The 16-bit timers KB40, KB41, and KB42 have the following functions.

1. PWM output

Each of the 16-bit timers KB4n can generate two PWM signals. The duty cycles and periods of the PWM signals can be changed during timer operation. The default level while the timer is stopped and the active level while the timer is operating can be set to the high level or low level. Using these functions enables control over PFC circuits in the continuous current mode (CCM-PFC) and half-bridge circuits.

2. A/D conversion start timing signal output

Output of the A/D conversion start timing signal can be controlled by using the 16-bit timer KB trigger compare register n (TKBTGCRn) of a 16-bit timer KB4n. Using this function enables synchronization of the timing of the start of A/D conversion with a 16-bit timer KB4n.

3. Simultaneous start/stop mode

Once the 16-bit timer KB40 is set as the master and the 16-bit timers KB41 and KB42 are set as slaves, the slave 16-bit timers KB41 and KB42 can be started and stopped at the same time in synchronization with the timing at which counting by the 16-bit timer KB40 starts or stops.

4. Simultaneous start/clear mode

Once the 16-bit timer KB40 is set as the master and the 16-bit timers KB41 and KB42 are set as slaves, the counting cycles of the master and slave timers can be synchronized. This mode allows the generation of up to six complementary PWM signals in three phases. This enables control over full-bridge and 3-phase inverter circuits.

5. Timer restart (by interlocked operation with a comparator or INTP)

Generation of a trigger source (one among the outputs from comparators 0 and 1 and INTP0 to INTP7) enables the restarting of timer output without CPU intervention. Using this function enables the implementation of, for example, PFC control in the critical conduction mode.

6. PWM output for IH control

Using timer restart enables this function to keep output from the TKBOn1 pin at the active level for a desired duration regardless of the period. This function can disable restart control by an external source during the period from the beginning of a cycle to the completion of the period at the active level of the TKBOn1 pin, thus guaranteeing the width of the period for output at the active level.

7. Forced output stop 1 (by interlocked operation with a comparator or INTP)

Generation of a trigger source (one among the outputs from comparators 0 and 1 and INTP0 to INTP7) enables fixing of timer output to the Hi-Z state or to the low or high level without CPU intervention, and asynchronously with respect to the fKBKC operating clock for the 16-bit timer KB4n. Once the cancel trigger of forced output stop 1 is set, the forced output stop state can be canceled in synchronization with the fKBKC operating clock for the 16-bit timer KB4n. Using these functions enables forcibly stopping the output of PWM signals for over-current protection (OCP) and over-voltage protection (OVP).

8. Forced output stop 2 (by interlocked operation with a comparator or INTP)

Generation of a trigger source (one among the outputs from comparators 0 and 1 and INTP0 to INTP7) enables fixing of timer output to the low or high level without CPU intervention, and asynchronously with respect to the fKBKC operating clock for the 16-bit timer KB4n. Release from the forced output stop state proceeds at the beginning of the counter cycle immediately after generation of the trigger source or a transition of the trigger source signal to the inactive level. Furthermore, using the fixed off function enables positioning of the period over which the active level of the trigger source signal is detected in the forced output stop state. Using these functions enables the implementation of multiple operations for a DC/DC converter, including control over peak current, fixing the output to the off state, and protection.

9. PWM output dithering

A waveform with a set duty cycle + one cycle of the counter clock per 16 clock cycles can be output over the range from 0 to 15 PWM cycles. Using this function enables a PWM waveform that is 16 times the counter-clock frequency to be output at an average resolution raised to that equivalent to 16 cycles of the 16-bit timer KB4n.

10. PWM output smooth start

Using a smooth start, in which the duty cycle is automatically increased after the start of the output of a PWM signal until the duty cycle reaches the value set for the duty cycle allows the output of PWM waveforms suitable for the control of inrush current and for over-voltage prevention. The initial duty cycle and the number of cycles in which incrementation of the duty cycle by one clock cycle is repeated can be set.

11. Masking of external trigger inputs while the TKBOnp output is active

Using timer restart enables this function to disable the control of restarting by an external source during the active-level period of the TKBOnp output and to suppress the switching of external sources due to changes in the level of the TKBOnp output.

12. Maximum frequency limiting

When a trigger is generated earlier than would be the case with the set maximum frequency, using timer restart enables the suspension of restarting until this is within the set maximum frequency for restarting. During PFC control in the critical conduction mode, using this function enables the prevention of switching at an excessively high frequency.

13. Interleaved PFC output mode

An external source (INTP0 or INTP1) can be used to automatically alternate the restarting of output between two output signals. This allows the implementation of interleaved PFC control in the critical conduction mode.

Remark The critical conduction mode is a method of PFC control that involves activating a switching FET in response to detecting an inductor current of zero.

9.2 Configuration of 16-bit Timers KB40, KB41, and KB42

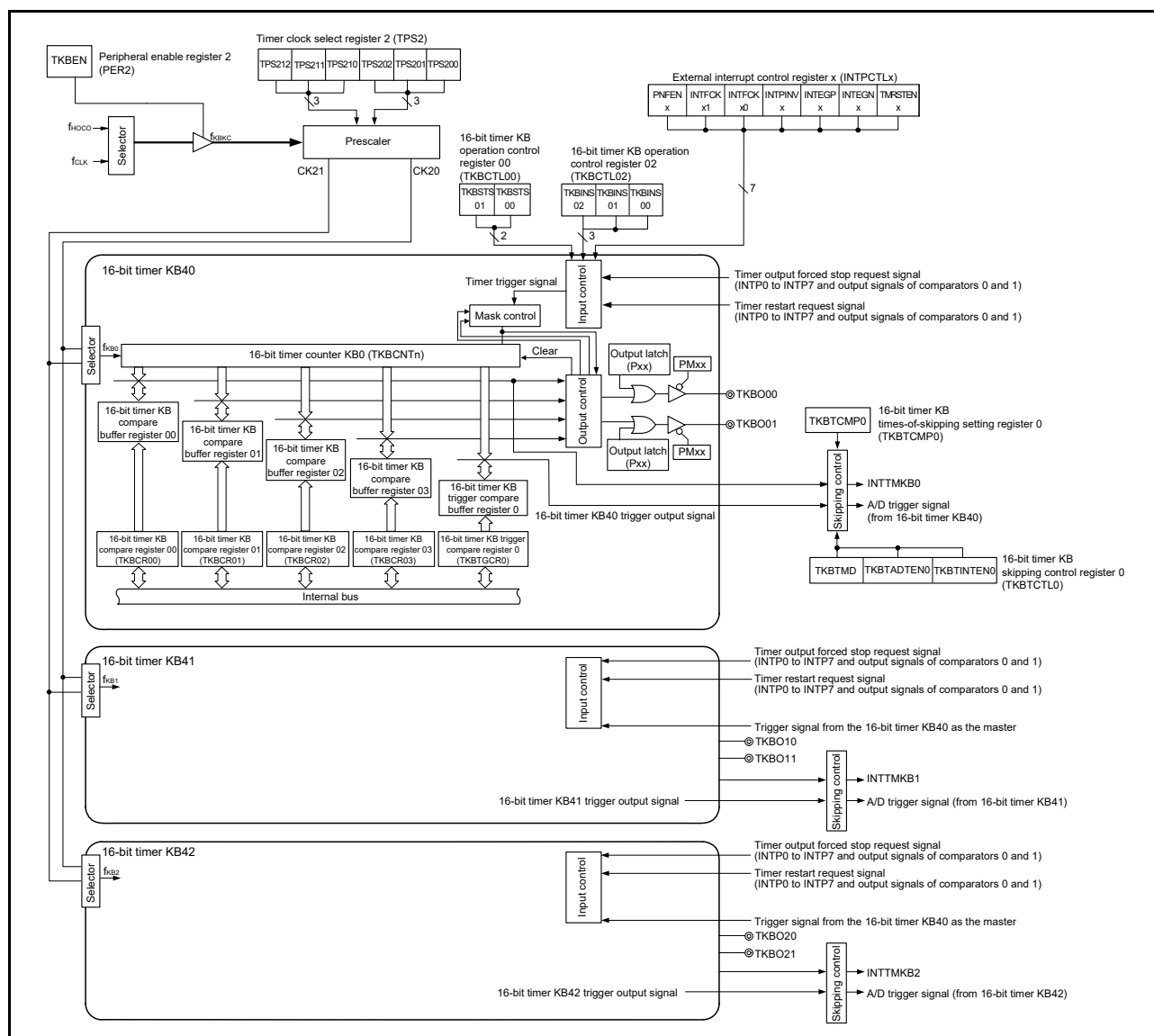
16-bit timers KB40, KB41, and KB42 include the following hardware.

Table 9 - 1 Configuration of 16-bit Timers KB40, KB41, and KB42

Item	Configuration
Timer/counter	16-bit timer counter KBn (TKBCNTn)
Timer output	TKBOn0 and TKBOn1 pins
Control registers	• Peripheral enable register 2 (PER2) • Peripheral reset control register 2 (PRR2) • Timer clock select register 2 (TPS2) • 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3) • 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2) • 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2) • 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2) • 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2) • 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2) • 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2) • 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2) • 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2) • 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2) • 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2) • 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2) • 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1) • 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2) • 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2) • 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2) • External interrupt control register x (INTPCTLx) (x = 0 to 7) • Port mode registers (PMxx) • Port registers (Pxx) • Port output mode registers (POMxx) • Port mode control A registers (PMCAxx) • Port mode control T registers (PMCTxx) • Port mode control E registers (PMCExx) • LCD port function registers (PFSEGx)

Figure 9 - 1 shows a block diagram of the 16-bit timer KB4n.

Figure 9 - 1 Block Diagram of the 16-bit Timer KB4n



Remark 1. f_{KBKC}: Operating clock of the 16-bit timer KB4n

fKBn: Counter clock of the 16-bit timer KB4n

Remark 2. $n = 0$ to 2

9.3 Registers for Controlling 16-bit Timers KB40, KB41, and KB42

The following registers are used to control 16-bit timers KB40, KB41, and KB42.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer clock select register 2 (TPS2)
- 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3)
- 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2)
- 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2)
- 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2)
- 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2)
- 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2)
- 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2)
- 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2)
- 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2)
- 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2)
- 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2)
- 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2)
- 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1)
- 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2)
- 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2)
- 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2)
- External interrupt control register x (INTPCTLx) (x = 0 to 7)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

9.3.1 Peripheral enable register 2 (PER2)

The PER2 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 16-bit timer KB4n is to be used, be sure to set bit 5 (TKBEN) of this register to 1. The PER2 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 9 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN	UTA2EN	UTA1EN	TRJ1EN	TRJ0EN
	TKBEN							
	Control of supply of an input clock to the 16-bit timer KB4n							
	0							
	Stops supply of an input clock. • The SFRs used by the 16-bit timer KB4n cannot be written.							
	1							
	Enables supply of an input clock. • The SFRs used by the 16-bit timer KB4n can be read and written.							

Caution 1. Be sure to set the TKBEN bit to 1 before making the 16-bit timer KB4n settings. If TKBEN = 0, writing to the registers for controlling the 16-bit timer KB4n is ignored, and the value read is 00H or 0000H.

Note, however, that writing to the following registers is valid.

- Timer clock select register 2 (TPS2)
- External interrupt control registers 0 to 7 (INTPCTL0 to INTPCTL7)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Caution 2. Selecting fHOCO as the operating clock for a 16-bit timer KB4n (CK20, CK21) requires selecting fIH as fCLK before setting the PER2.TKBEN bit. To change the clock selected for fCLK from fIH to another clock, clear the PER2.TKBEN bit to 0 beforehand.

9.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 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 5 (TKBRES) to 1 to reset the 16-bit timer KB4n. The PRR2 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 9 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
PRR2	EXSDRES	OSDCRES	TKBRES	0	0	0	TRJ1RES	TRJ0RES
	Control resetting of the 16-bit timer KB4n							
	0	The 16-bit timer KB4n is released from the reset state.						
	1	The 16-bit timer KB4n is in the reset state. • The SFRs for use with the 16-bit timer KB4n are initialized.						

Caution Be sure to clear bits 4 to 2 to 0.

9.3.3 Timer clock select register 2 (TPS2)

This is an 8-bit register used to select two operating clock signals (CK20, CK21) that are supplied in common to the KB40, KB41, and KB42 16-bit timers from the prescaler. CK21 is selected by using bits 6 to 4 of the TPS2 register, and CK20 is selected by using bits 2 to 0. Overwriting of the TPS2 register during timer operation is possible only in the following cases.

- The TPS20[2:0] bits can be overwritten while all timers for which CK20 is selected as the operating clock (TKBCTLn1.TKBCKSn = 0) are stopped. That is, it requires that the applicable 16-bit timers KB40, KB41, and KB42 be stopped (TKBCTLn1.TKBCEn = 0) (n = 0 to 2).
- The TPS21[2:0] bits can be overwritten while all timers for which CK21 is selected as the operating clock (TKBCTLn1.TKBCKSn = 1) are stopped. That is, it requires that the applicable 16-bit timers KB40, KB41, and KB42 be stopped (TKBCTLn1.TKBCEn = 0) (n = 0 to 2).

The TPS2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 4 Format of Timer Clock Select Register 2 (TPS2)

Address: F037FH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TPS2	0	TPS212	TPS211	TPS210	0	TPS202	TPS201	TPS200

TPS2k2	TPS2k1	TPS2k0	Selection of the operating clock (CK2k) ^{Notes 1, 2} (k = 0, 1)				
				fCLK = 20 MHz	fCLK = 32 MHz	fHOCO = 48 MHz	fHOCO = 64 MHz
0	0	0	fCLK, fHOCO	20 MHz	32 MHz	48 MHz	64 MHz
0	0	1	fCLK/2	10 MHz	16 MHz	Setting prohibited	Setting prohibited
0	1	0	fCLK/2 ²	5 MHz	8 MHz		
0	1	1	fCLK/2 ³	2.5 MHz	4 MHz		
1	0	0	fCLK/2 ⁴	1.25 MHz	2 MHz		
1	0	1	fCLK/2 ⁵	625 kHz	1 MHz		
Other than above			Setting prohibited				

Note 1. Changing the clock selected for fCLK (by changing the value in the system clock control register (CKC)) requires that the applicable 16-bit timers KB40, KB41, and KB42 be stopped (TKBCTLn1.TKBCEn = 0).

Note 2. When the FRQSEL4 bit of the user option byte (000C2H) is cleared to 0 or set to 1, fCLK and fHOCO are respectively selected. Selecting fHOCO as the operating clock for a 16-bit timer KB4n (CK20, CK21) requires selecting fH as fCLK before setting bit 5 (TKBEN) of the peripheral enable register 2 (PER2). To change the clock selected for fCLK from fH to another clock, clear bit 5 (TKBEN) of the peripheral enable register 2 (PER2) to 0 beforehand.

Caution 1. Be sure to clear bits 7 and 3 to 0.

Caution 2. Setting the FRQSEL4 bit of the user option byte (000C2H) to 1 requires that the timer be used with VDD no lower than 2.2 V.

Remark fCLK: CPU/peripheral hardware clock frequency
fHOCO: High-speed on-chip oscillator clock frequency (64 MHz max.)

9.3.4 16-bit timer KB compare register nm (TKBCRnm) (n = 0 to 2, m = 0 to 3)

The TKBCRnm register can be refreshed (by writing the existing value to it) or its value can be overwritten while the timer is counting (TKBCTLn1.TKBCEn = 1). On overwriting the value of the TKBCRnm register while the timer is operating, the value that was written is latched and then transferred to the internal buffer register at either of the following times, and the value of the buffer register is changed accordingly.

- Counting by the counter is started by changing the setting of the TKBCTLn1.TKBCEn bit from 0 to 1.
- A batch-overwriting trigger is generated while the value of the TKBFLGn.TKBRSFn flag bit is 1.

This register can be read or written in 16-bit units. The value of this register following a reset is 0000H.

Figure 9 - 5 Format of 16-bit Timer KB Compare Register nm (TKBCRnm)

Address: F0740H (TKBCR00), F0742H (TKBCR01), F0744H (TKBCR02), F0746H (TKBCR03),
F0780H (TKBCR10), F0782H (TKBCR11), F0784H (TKBCR12), F0786H (TKBCR13),
F04A0H (TKBCR20), F04A2H (TKBCR21), F04A4H (TKBCR22), F04A6H (TKBCR23)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRnm																

Remark n = 0 to 2, m = 0 to 3

9.3.5 16-bit timer KB trigger compare register n (TKBTGCRn) (n = 0 to 2)

The TKBTGCRn register can be refreshed (by writing the existing value to it) or its value can be overwritten while the timer is counting (TKBCTLn1.TKBCEn = 1). On overwriting the value of the TKBTGCRn register while the timer is operating, the value that was written is latched and then transferred to the internal buffer register at either of the following times, and the value of the buffer register is changed accordingly.

- Counting by the counter is started by changing the setting of the TKBCTLn1.TKBCEn bit from 0 to 1.
- A batch-overwriting trigger is generated while the value of the TKBFLGn.TKBRSFn flag bit is 1.

A periodic signal generated under the control of this register can be used as a hardware trigger for A/D conversion. This register can be read or written in 16-bit units. The value of this register following a reset is 0000H.

Figure 9 - 6 Format of 16-bit Timer KB Trigger Compare Register n (TKBTGCRn)

Address: F0748H (TKBTGCR0), F0788H (TKBTGCR1), F04A8H (TKBTGCR2)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBTGCRn																

Remark n = 0 to 2

9.3.6 16-bit timer KB operation control register n0 (TKBCTLn0) (n = 0 to 2)

The TKBCTLn0 register is used to set the following.

- Masking of external trigger inputs while the TKBOnp output is active
- PWM output smooth start
- PWM output dithering
- Maximum frequency limiting
- Interleaved PFC output mode
- PWM output for IH control
- Batch overwriting of the compare registers in response to an external trigger
- Selecting the timer restart trigger

The TKBCTLn0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0040H.

Figure 9 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (1/2)

Address: F0762H (TKBCTL00), F07A2H (TKBCTL10), F04C2H (TKBCTL20)

After reset: 0040H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn0	TKBTGMn1	0	TKBSSEn1	TKBDIEn1	TKBTGMn0	0	TKBSSEn0	TKBDIEn0
	7	6	5	4	3	2	1	0
	TKBMFEn	TKBIRSn2	TKBIRSn1	TKBIRSn0	TKBIHEn	TKBTSEn	TKBSTSn1	TKBSTSn0
TKBTGMn p	Control of masking of external trigger inputs while the TKBOnp output is active							
0	Detection of the external trigger input selected by the TKBSTSn[1:0] and TKBCTLn2.TKBINSn[2:0] bits is always accepted.							
1	Detection of the external trigger input selected by the TKBSTSn[1:0] and TKBCTLn2.TKBINSn[2:0] bits is not accepted while the PWM output of TKBOnp is at the active level.							
TKBSSEn	Control of PWM output smooth start for TKBOnp							
0	PWM output smooth start is not used.							
1	PWM output smooth start is used.							
TKBDIEn	Control of PWM output dithering for TKBOnp							
0	PWM output dithering is not used.							
1	PWM output dithering is used.							
TKBMFEn	Control of maximum frequency limiting for TKBOnp							
0	Maximum frequency limiting is not used.							
1	Maximum frequency limiting is used.							

Figure 9 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (2/2)

TKBIRSn2	TKBIRSn1	TKBIRSn0	Configuration of the acceptable judgment range of an INTP1 input that results in the immediate output of TKBOn1 in the interleaved PFC output mode
0	0	0	$T/2$ to $T/2 + T/64$
0	0	1	$T/2$ to $T/2 + T/32$
0	1	0	$T/2$ to $T/2 + T/16$
0	1	1	$T/2$ to $T/2 + T/8$
1	0	0	No range for judgment applies. When INTP1 is detected within the range of $T/2$, TKBOn1 is output with the timing of $T/2$ (this setting is the initial value following a reset).
1	0	1	No range for judgment applies. When INTP1 is detected within the range of $T/2$, TKBOn1 is output at the time of detection.
Other than above			Setting prohibited

TKBIHEn	Control of PWM output for IH control
0	PWM output for IH control is not used.
1	PWM output for IH control is used.

TKBTSEn	Control of batch overwriting of the compare registers in response to an external trigger
0	Batch overwriting of the compare registers in response to an external trigger is not used.
1	Batch overwriting of the compare registers in response to an external trigger is used.

TKBSTSn1	TKBSTSn0	Selection of the restart trigger for the 16-bit timer KB4n
0	0	Trigger input is not used.
0	1	External interrupt signal (INTP0)
1	0	External interrupt signal (INTP1)
1	1	External interrupt signal (INTP2)

Caution 1. Overwriting the TKBCTLn0 register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 14 and 10 to 0.

Caution 3. Regarding the settings for INTP0 to INTP2, see Section 25 Interrupt Functions and 9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7).

Caution 4. When the setting of the TKBIHEn bit is 1, be sure to clear bits 15 to 7 to 0.

Caution 5. In the interleaved PFC output mode, the setting of the TKBSTSn[1:0] bits is ignored and an external interrupt signal (INTP0) is selected as the restart trigger.

Remark 1. n = 0 to 2; p = 0, 1

Remark 2. T is the period for the immediately preceding pulse.

9.3.7 16-bit timer KB operation control register n1 (TKBCTLn1) (n = 0 to 2)

The TKBCTLn1 register is used to control the counting operation and to set the counter clock for the 16-bit timer. The TKBCTLn1 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 9 - 8 Format of 16-bit Timer KB Operation Control Register n1 (TKBCTLn1)

Address: F0769H (TKBCTL01), F07A9H (TKBCTL11), F04C9H (TKBCTL21)

After reset: 00H

R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	0	0	TKBCKSn	TKBSCMn	0	TKBMDn1	TKBMDn0
TKBCEn								
Control over the 16-bit timer KB4n operation								
0								
Stops timer counting (the counter is set to FFFFH).								
1								
Enables timer counting.								
TKBCKSn								
Clock selection for the 16-bit timer KB4n								
0								
CK20 clock selected by the TPS20[2:0] bits								
1								
CK21 clock selected by the TPS21[2:0] bits								
TKBSCMn								
Control to start operation of the 16-bit timer KB4n								
0								
The timer operates with the clock selected by the TKBCKSn bit.								
1								
Counting begins when the CK20 and CK21 clock signals match. After the operation has started, the clock selected by the TKBCKSn bit is used to drive operation.								
TKBMDn1 TKBMDn0								
Operating mode selection for the 16-bit timer KB4n								
0 0								
Standalone mode (for the master)								
0 1								
Simultaneous start/stop mode (for the slaves)								
1 0								
Simultaneous start/clear mode (for the slaves)								
1 1								
Interleaved PFC output mode								

Caution 1. Overwriting the TKBCTLn1 register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 6, 5, and 2 to 0.

Caution 3. The TKBSCMn bit can only be set to 1 when at least one slave is operating in the simultaneous start/stop mode. For details, see 9.4.8 Simultaneous start/stop mode.

Remark n = 0 to 2

9.3.8 16-bit timer KB operation control register n2 (TKBCTLn2) (n = 0 to 2)

The TKBCTLn2 register is used for an extended setting of the timer restart trigger. Using this register enables the following items.

- Settings of the trigger that are additional to the settings selectable by the TKBSTSn[1:0] bits
- Masking of the source of the external trigger set by an extended setting is also selectable.

The TKBCTLn2 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 9 Format of 16-bit Timer KB Operation Control Register n2 (TKBCTLn2)

Address: F076AH (TKBCTL02), F07AAH (TKBCTL12), F04CAH (TKBCTL22)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn2	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	TKBINSn2	TKBINSn1	TKBINSn0
	TKBINSn2	TKBINSn1	TKBINSn0	Selection of the restart trigger for the 16-bit timer KB4n				
	0	0	0	Trigger input is not used.				
	0	0	1	External interrupt signal (INTP3)				
	0	1	0	External interrupt signal (INTP4)				
	0	1	1	External interrupt signal (INTP5)				
	1	0	0	External interrupt signal (INTP6)				
	1	0	1	External interrupt signal (INTP7)				
	1	1	0	Output signal of comparator 0				
	1	1	1	Output signal of comparator 1				

Caution 1. Overwriting the TKBCTLn2 register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 15 to 3 to 0.

Caution 3. Regarding the settings for INTP3 to INTP7, see Section 25 Interrupt Functions and 9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7).

Caution 4. Regarding the settings for the output signals from comparators 0 and 1, see Section 18 Comparator (CMP).

Caution 5. In the interleaved PFC output mode, the setting of the TKBINSn[2:0] bits is ignored and an external interrupt signal (INTP0) is selected as the restart trigger.

Remark n = 0 to 2

9.3.9 16-bit timer KB output control register n0 (TKBIOCn0) (n = 0 to 2)

The TKBIOCn0 register is used to set the default and active levels for the output (TKBOnp) from a 16-bit timer KB4n. The TKBIOCn0 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 9 - 10 Format of 16-bit Timer KB Output Control Register n0 (TKBIOCn0)

Address: F0766H (TKBIOC00), F07A6H (TKBIOC10), F04C6H (TKBIOC20)
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TKBIOCn0	0	0	0	0	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0

TKBTOLnp	Active level setting for the timer output (TKBOnp)	
0	High level	
1	Low level	

TKBTODnp	Default level setting for the timer output (TKBOnp)	
0	Low level	
1	High level	

Caution 1. Overwriting the TKBIOCn0 register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 7 to 4 to 0.

Caution 3. The actual TKBOnp pin output is not only set by these TKBOnp output settings, but also by the port mode register (PMxx) and port register (Pxx) settings for the multiplexed port pins.

Remark n = 0 to 2; p = 0, 1

9.3.10 16-bit timer KB output control register n1 (TKBIOCn1) (n = 0 to 2)

The TKBIOCn1 register is used to disable or enable timer control of the output (TKBOnp) from the corresponding 16-bit timer KB4n. The TKBIOCn1 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 9 - 11 Format of 16-bit Timer KB Output Control Register n1 (TKBIOCn1)

Address: F0768H (TKBIOC01), F07A8H (TKBIOC11), F04C8H (TKBIOC21)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBIOCn1	0	0	0	0	0	0	TKBTOEn1	TKBTOEn0

TKBTOEnp	Enabling or disabling the timer output (TKBOnp)
0	Disables the timer output. The low or high level is output when TKBTOEnp = 0 or 1, respectively.
1	Enables the timer output.

Caution 1. The TKBIOCn1 register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to 0.

Caution 3. The actual TKBOnp pin output is not only set by these TKBOnp output settings, but also by the port mode register (PMxx) and port register (Pxx) settings for the multiplexed port pins.

Remark n = 0 to 2; p = 0, 1

9.3.11 16-bit timer KB flag register n (TKBFLGn) (n = 0 to 2)

The TKBFLGn register indicates the state of the 16-bit timer KB4n. The TKBFLGn 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 9 - 12 Format of 16-bit Timer KB Flag Register n (TKBFLGn)

Address: F0753H (TKBFLG0), F0793H (TKBFLG1), F04B3H (TKBFLG2)
 After reset: 00H
 R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBFLGn	TKBSSF _{n1}	TKBSSF _{n0}	TKBSEF _{n1}	TKBSEF _{n0}	TKBIRF _n	TKBIEF _n	TKBMFF _n	TKBRSF _n
TKBSSF _n	Status flag for PWM output smooth start on the TKBO _n pin							
0	PWM output smooth start is not in use.							
1	PWM output smooth start is in use.							
TKBSEF _n	Error flag for PWM output smooth start on the TKBO _n pin							
0	No error has occurred, or clearing triggered by the TKBCL _{Rn} .TKBCLSE _n bit has been completed.							
1	An error has occurred; that is, the TKBTRG _n .TKBRD _{Tn} bit was set to 1 during usage of PWM output smooth start (TKBSSF _n = 1).							
TKBIRF _n	Error flag for non-detection of an INTP1 trigger in the interleaved PFC output mode							
0	No error has occurred, or clearing triggered by the TKBCL _{Rn} .TKBCLIR _n bit has been completed.							
1	Either of the kinds of error described below has occurred. - When TKBCTL_{n0}.TKBIRS_n[2] = 0: No INTP1 trigger was detected within the range from 0 to T/2 and within the range for judgment specified by the TKBCTL_{n0}.TKBIRS_n[2:0] bits. - When TKBCTL_{n0}.TKBIRS_n[2:0] = 100B or 101B: No INTP1 trigger was detected within the cycle.							
TKBIEF _n	Error flag for multiple detection of INTP1 triggers in the interleaved PFC output mode							
0	No error has occurred, or clearing triggered by the TKBCL _{Rn} .TKBCLIE _n bit has been completed.							
1	An error has occurred; that is, another INTP1 input was redundantly detected while the TKBO _{n1} output was at the active level.							
TKBMFF _n	Status flag for maximum frequency limiting							
0	Maximum frequency limiting has not started, or clearing triggered by the TKBCL _{Rn} .TKBCLMF _n bit has been completed.							
1	Maximum frequency limiting has started.							
TKBRSF _n	Status flag to indicate a pending batch-overwriting trigger							
0	Batch overwriting is enabled, or batch overwriting in response to the generation of a batch-overwriting trigger is completed.							
1	Batch overwriting is held pending, to wait for the completion of batch overwriting in response to writing to the batch-overwriting trigger bit (TKBTRG _n .TKBRD _{Tn}).							

Remark 1. $n = 0$ to 2 ; $p = 0, 1$

Remark 2. T is the period for the immediately preceding pulse.

9.3.12 16-bit timer KB trigger register n (TKBTRGn) (n = 0 to 2)

This is a trigger register used for batch overwriting of the compare registers for the 16-bit timer KB4n. The TKBTRGn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 13 Format of 16-bit Timer KB Trigger Register n (TKBTRGn)

Address: F0752H (TKBTRG0), F0792H (TKBTRG1), F04B2H (TKBTRG2)
After reset: 00H
R/W: W

Symbol	7	6	5	4	3	2	1	<0>
TKBTRGn	0	0	0	0	0	0	0	TKBRDTn
TKBRDTn	Trigger to request batch overwriting of the compare registers							
0	Invalid setting							
1	Triggers a request for batch overwriting of the compare registers.							

Remark n = 0 to 2

9.3.13 16-bit timer KB flag clear trigger register n (TKBCLRn) (n = 0 to 2)

The TKBCLRn register is used to clear flags in the 16-bit timer KB flag register n (TKBFLGn). The TKBCLRn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 14 Format of 16-bit Timer KB Flag Clear Trigger Register n (TKBCLRn)

Address: F0767H (TKBCLR0), F07A7H (TKBCLR1), F04C7H (TKBCLR2)

After reset: 00H

R/W: W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
TKBCLRn	0	0	TKBCLSEn1	TKBCLSEn0	TKBCLIRn	TKBCLIEEn	TKBCLMFEn	0

TKBCLSEn p	Trigger for clearing the error flag for PWM output smooth start on the TKBOnp pin							
0	Invalid setting							
1	Triggers clearing of the TKBFLGn.TKBSEFnp flag to 0.							

TKBCLIRn	Trigger for clearing the error flag for non-detection of an INTP1 trigger in the interleaved PFC output mode							
0	Invalid setting							
1	Triggers clearing of the TKBFLGn.TKBIRFEn flag to 0.							

TKBCLIEEn	Trigger for clearing the error flag for multiple detection of INTP1 triggers in the interleaved PFC output mode							
0	Invalid setting							
1	Triggers clearing of the TKBFLGn.TKBIEFEn flag to 0.							

TKBCLMFEn	Trigger for clearing the status flag for maximum frequency limiting							
0	Invalid setting							
1	Triggers clearing of the TKBFLGn.TKBMFFEn flag to 0.							

Remark n = 0 to 2; p = 0, 1

9.3.14 16-bit timer KB dithering count register np (TKBDNRnp) (n = 0 to 2; p = 0, 1)

The TKBDNRnp register is used for PWM dithering of the TKBOnp output. When the value of the four higher-order bits of this register is N (N = 0H to FH), output of the active period of the PWM signal is extended by one cycle of the counter clock N times in every 16 cycles of PWM output. **Table 9 - 2** shows the relation between the setting of the TKBDNRnp register and the number of times, N, in which the active period is extended by one cycle of the counter clock. The TKBDNRnp register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 15 Format of 16-bit Timer KB Dithering Count Register np (TKBDNRnp)

Address: F074EH (TKBDNR00), F078EH (TKBDNR10), F04AEH (TKBDNR20), F0750H (TKBDNR01),
F0790H (TKBDNR11), F04B0H (TKBDNR21)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBDNRnp	TKBDNR3n	TKBDNR2n	TKBDNR1n	TKBDNR0n	0	0	0	0

Caution Be sure to clear bits 3 to 0 to 0.

Remark n = 0 to 2; p = 0, 1

Table 9 - 2 Relationship between the Setting of the TKBDNRnp Bit and the Cycles with the Active Period Extended by One Cycle of the Counter Clock

Number of repetitions (N) \ Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

Remark 1.  Cell cycle: Inactive output via TKBCRn1 and TKBCRn3 register settings

 Cell cycle: Inactive output via TKBCRn1 and TKBCRn3 register settings + 1

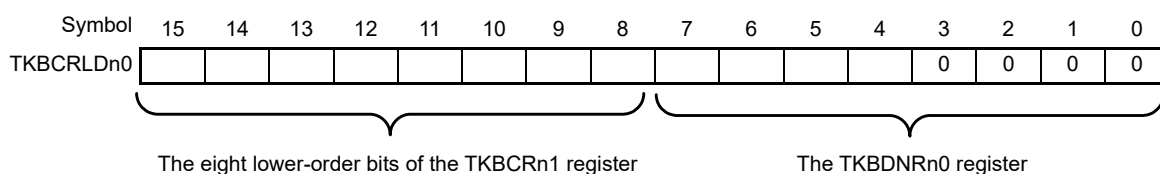
Remark 2. n = 0 to 2; p = 0, 1

9.3.15 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) (n = 0 to 2)

The TKBCRLDn0 register holds the value of the eight lower-order bits of the TKBCRn1 register in the eight higher-order bits and the value of the TKBDNRn0 register in the eight lower-order bits. The TKBCRLDn0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 16 Format of 16-bit Timer KB Compare 1L & Dithering Count Register n0 (TKBCRLDn0)

Address: F0754H (TKBCRLD00), F0794H (TKBCRLD10), F04B4H (TKBCRLD20)
 After reset: 0000H
 R/W: R/W



Caution Be sure to clear bits 3 to 0 to 0.

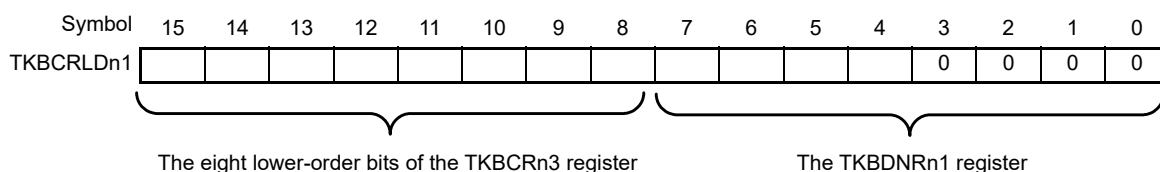
Remark n = 0 to 2

9.3.16 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) (n = 0 to 2)

The TKBCRLDn1 register holds the value of the eight lower-order bits of the TKBCRn3 register in the eight higher-order bits and the value of the TKBDNRn1 register in the eight lower-order bits. The TKBCRLDn1 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 17 Format of 16-bit Timer KB Compare 3L & Dithering Count Register n1 (TKBCRLDn1)

Address: F0756H (TKBCRLD01), F0796H (TKBCRLD11), F04B6H (TKBCRLD21)
 After reset: 0000H
 R/W: R/W



Caution Be sure to clear bits 3 to 0 to 0.

Remark n = 0 to 2

9.3.17 16-bit timer KB smooth start initial duty register np (TKBSIRnp) (n = 0 to 2; p = 0, 1)

The TKBSIRnp register is used to set the initial duty cycle (active period) in smooth starting of PWM output on the TKBOnp pin. The TKBSIRnp register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 18 Format of 16-bit Timer KB Smooth Start Initial Duty Register np (TKBSIRnp)

Address: F074AH (TKBSIR00), F078AH (TKBSIR10), F04AAH (TKBSIR20), F074CH (TKBSIR01),
F078CH (TKBSIR11), F04ACH (TKBSIR21)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBSIRnp																

Remark n = 0 to 2; p = 0, 1

9.3.18 16-bit timer KB smooth start step width register np (TKBSSRnp) (n = 0 to 2; p = 0, 1)

The TKBSSRnp register is used for smooth starting of PWM output on the TKBOnp pin. When the value of the four lower-order bits of this register is N (N = 0H to FH), output of a PWM waveform with the active period specified by the TKBSIRnp register continues for N + 1 cycles. Afterward, output of a waveform with that set active period + 1 clock cycle continues for N + 1 cycles, followed by output at the active period + 2 clock cycles for N + 1 cycles, and so on. Finally, when the duty cycle (active period) reaches the value set in the TKBCRn1 or TKBCRn3 register, operations for smooth starting of PWM output are stopped and normal PWM output starts. The TKBSSRnp register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 19 Format of 16-bit Timer KB Smooth Start Step Width Register np (TKBSSRnp)

Address: F074FH (TKBSSR00), F078FH (TKBSSR10), F04AFH (TKBSSR20), F0751H (TKBSSR01),
F0791H (TKBSSR11), F04B1H (TKBSSR21)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBSSRnp	0	0	0	0				

Caution Be sure to clear bits 7 to 4 to 0.

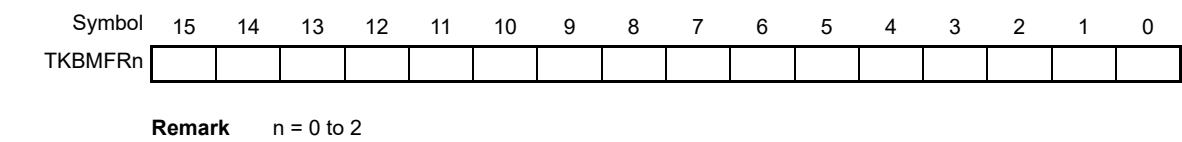
Remark n = 0 to 2; p = 0, 1

9.3.19 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) (n = 0 to 2)

The TKBMFRn register is used to set the minimum period for restarting the timer in response to an external trigger. If a trigger input is detected while the value of a 16-bit timer counter KBn register (TKBCNTn) is smaller than that of the TKBMFRn register, the trigger is held pending, and the TKBCNTn counter is cleared and restarted after counting has reached the value set in the TKBMFRn register. The TKBMFRn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 20 Format of 16-bit Timer KB Maximum Frequency Limit Setting Register n (TKBMFRn)

Address: F0764H (TKBMFR0), F07A4H (TKBMFR1), F04C4H (TKBMFR2)
After reset: 0000H
R/W: R/W



9.3.20 16-bit timer KB skipping control register n (TKBTCTLn) (n = 0 to 2)

The TKBTCTLn register is used to control the skipping of interrupt signals and the A/D trigger signals. The TKBTCTLn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 21 Format of 16-bit Timer KB Skipping Control Register n (TKBTCTLn)

Address: F0490H (TKBTCTL0), F0492H (TKBTCTL1), F0494H (TKBTCTL2)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBTCTLn	0	0	0	TKBTMDn	0	TKBTADTE Nn	0	TKBTINTEN n
TKBTMDn								
Enabling or disabling output in the first cycle after a 16-bit timer KB4n starts to operate								
0 Output is disabled.								
1 Output is enabled.								
TKBTADTE Nn								
Enabling or disabling skipping of the A/D trigger signal from the 16-bit timer KB4n								
0 Skipping is disabled.								
1 Skipping is enabled.								
TKBTINTE Nn								
Enabling or disabling skipping of the count end interrupt signal from the 16-bit timer KB4n								
0 Skipping is disabled.								
1 Skipping is enabled.								

Caution 1. Overwriting the TKBTCTLn register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 7 to 5, 3, and 1 to 0.

Remark n = 0 to 2

9.3.21 16-bit timer KB times-of-skipping setting register n (TKBTCMPn) (n = 0 to 2)

The TKBTCMPn register is used to control the number of times the interrupt signals and the A/D trigger signals are skipped. The TKBTCMPn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 22 Format of 16-bit Timer KB Times-of-Skipping Setting Register n (TKBTCMPn)

Address: F0491H (TKBTCMP0), F0493H (TKBTCMP1), F0495H (TKBTCMP2)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TKBTCMPn	0	0	0	TKBTCMPn[4:0]				
TKBTCMPn[4:0]					Number of times for skipping			
0	0	0	0	0	No skipping			
0	0	0	0	1	Skipping once After an interrupt signal or an A/D trigger signal is output, the next signal is not output for one cycle.			
⋮					⋮			
1	1	1	1	0	Skipping 30 times After an interrupt signal or an A/D trigger signal is output, the next signal is not output for 30 cycles.			
1	1	1	1	1	Skipping 31 times After an interrupt signal or an A/D trigger signal is output, the next signal is not output for 31 cycles.			

Caution Be sure to clear bits 7 to 5 to 0.

Remark n = 0 to 2

9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7)

The INTPCTLx register is used to control the INTPx external interrupt. The INTPCTLx register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The INTPCTL0 to INTPCTL7 registers correspond to the INTP0 to INTP7 external interrupts, respectively.

Figure 9 - 23 Format of External Interrupt Control Register x (INTPCTLx)

Address: F0370H (INTPCTL0), F0371H (INTPCTL1), F0372H (INTPCTL2), F0373H (INTPCTL3),
F0374H (INTPCTL4), F0375H (INTPCTL5), F0376H (INTPCTL6), F0377H (INTPCTL7)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
INTPCTLx	PNFENx	INTFCKx1	INTFCKx0	INTPINVx	INTEGPx	INTEGNx	0	TMRSTENx
PNFENx								
Enabling or disabling the use of the noise filter for the INTPx external interrupt								
0 The noise filter is used.								
1 The noise filter is not used.								
INTFCKx1 INTFCKx0								
Selection of external interrupt INTPx filter								
0 0 No INTPx filter								
0 1 INTPx filter enabled with sampling at fCLK or fHOCO								
1 0 INTPx filter enabled with sampling at fCLK/8 or fHOCO/8								
1 1 INTPx filter enabled with sampling at the frequency of TAU output (TO01)								
INTPINVx								
Setting of the output inversion of the INTPx external interrupt signal								
0 The INTPx signal is not inverted.								
1 The INTPx signal is inverted.								
INTEGPx INTEGNx								
Selection of the valid edge or edges of the INTPx external interrupt								
0 0 Edge detection disabled (Output of the timer restart signal for the 16-bit timers KB40, KB41, and KB42 is disabled and the output signal is fixed to the low level.)								
0 1 Falling edge (Output of the timer restart signal for the 16-bit timers KB40, KB41, and KB42 is enabled.)								
1 0 Rising edge (Output of the timer restart signal for the 16-bit timers KB40, KB41, and KB42 is enabled.)								
1 1 Both rising and falling edges (Output of the timer restart signal for the 16-bit timers KB40, KB41, and KB42 is enabled.)								
TMRSTENx								
Enabling or disabling of the functions of timers KB40, KB41, and KB42 interlocked for triggering in response to the INTPx external interrupt								
0 Triggering of timer restart and forced stopping of output from the 16-bit timers KB40, KB41, and KB42 in response to the INTPx is disabled.								
1 Triggering of timer restart and forced stopping of output from the 16-bit timers KB40, KB41, and KB42 in response to the INTPx is enabled. Note								

(Note, Caution, and Remark are listed on the next page.)

Note For using the INTP0 pin as an external trigger for timer restart and forced stopping of output from the 16-bit timers KB40, KB41, and KB42, clear the ISC.ISC0 bit to 0 to use the signal input through the INTP0 pin as an external interrupt signal. For details, see **7.3.16 Input switch control register (ISC)**.

Caution Be sure to clear bit 1 to 0.

Remark x = 0 to 7

9.3.23 Registers for controlling the port functions multiplexed with the outputs of 16-bit timers KB40, KB41, and KB42

Set the following registers for controlling the port functions multiplexed with the outputs of 16-bit timers KB40, KB41, and KB42.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

9.4 Operations of 16-bit Timers KB40, KB41, and KB42

The operational specifications of 16-bit timers KB40, KB41, and KB42 are described below.

- Basic counter operation (see **9.4.1.**)
- Default level and active level (see **9.4.2.**)
- Stopping and restarting operation (see **9.4.3.**)
- Batch overwriting (see **9.4.4.**)

There are the following six operating modes for 16-bit timers KB40, KB41, and KB42.

- Standalone mode (with the period controlled by the TKBCRn0 register) (see **9.4.6.**)
- Standalone mode (with the period controlled by external trigger input) (see **9.4.7.**)
- Simultaneous start/stop mode (with the period controlled by the TKBCRn0 register) (see **9.4.8.**)
- Simultaneous start/stop mode (with the period controlled by external trigger input) (see **9.4.8.**)
- Simultaneous start/clear mode (with the period controlled by the master) (see **9.4.9.**)
- Interleaved power factor correction (PFC) output mode (see **9.4.10.**)

Figure 9 - 24 Example of Settings for 16-bit Timer KB4n Operation (Flow of Starting Operation)

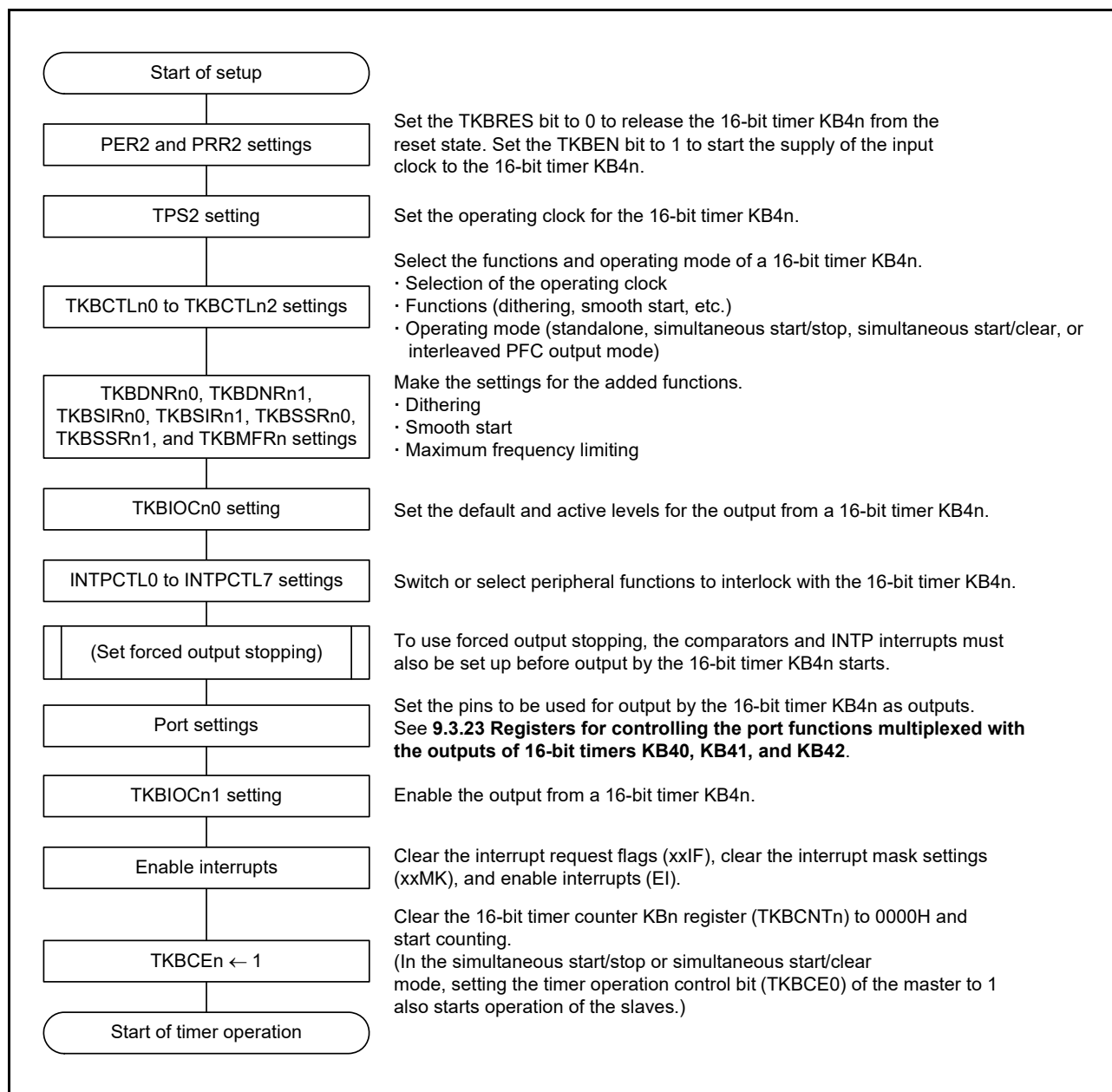


Figure 9 - 25 Example of Settings for 16-bit Timer KB4n Operation (Flow of Stopping Operation)

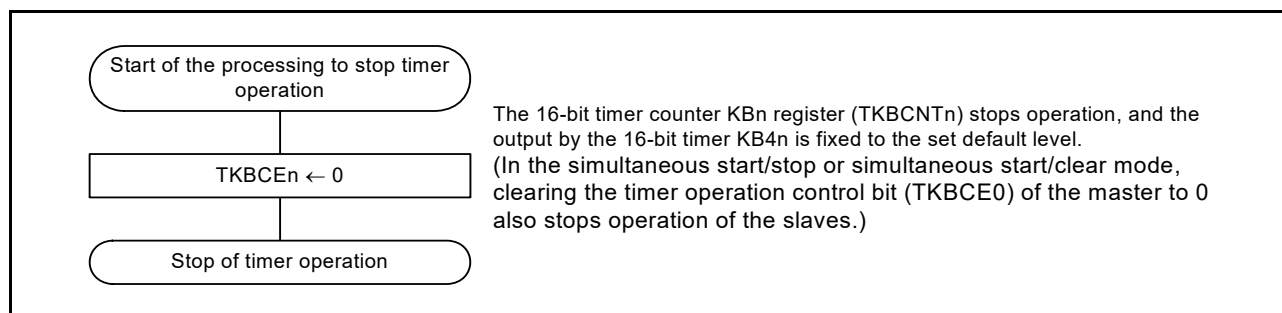
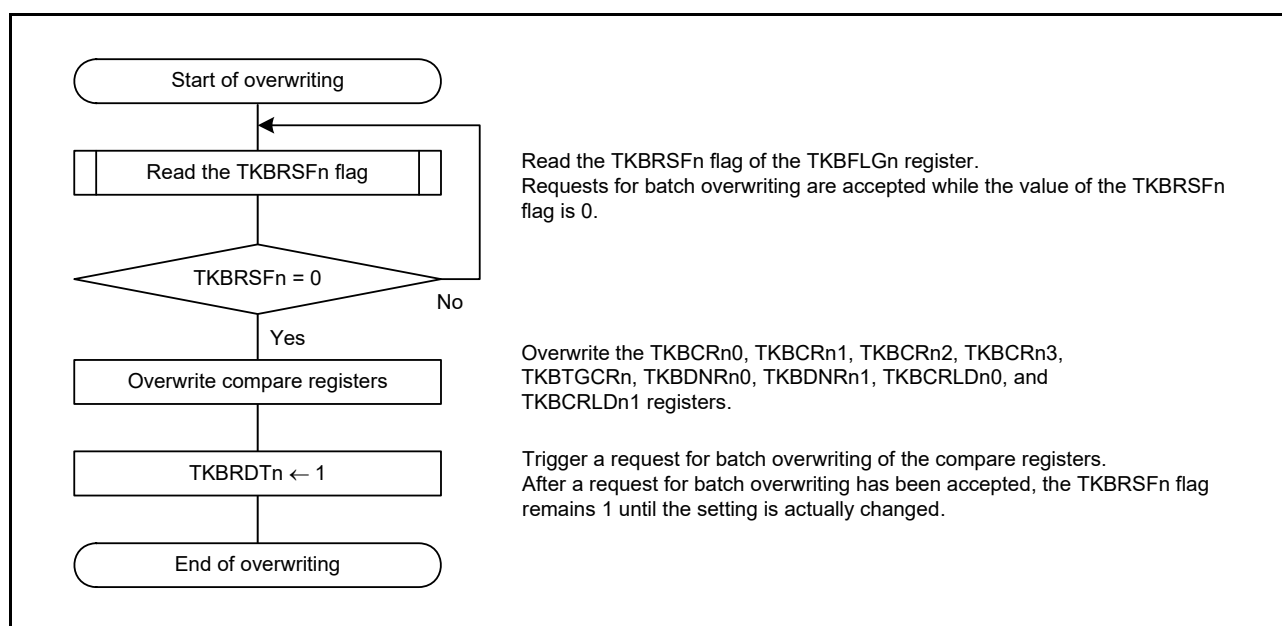


Figure 9 - 26 Example of Settings for 16-bit Timer KB4n Operation (Flow for Batch Overwriting of the Compare Registers)



Remark The batch overwriting is used to change the timer counter operation settings while the 16-bit timer KB4n is operating. The set values are reflected to the operation from the next restart.

9.4.1 Basic counter operation

1. Operation to start counting

In any mode, changing the TKBCEn bit from 0 to 1 starts the 16-bit counter for the 16-bit timer KB4n counting from the initial value FFFFH. The value of the counter is incremented from FFFFH to 0000H, 0001H, 0002H, 0003H and so on.

2. Operation for clearing

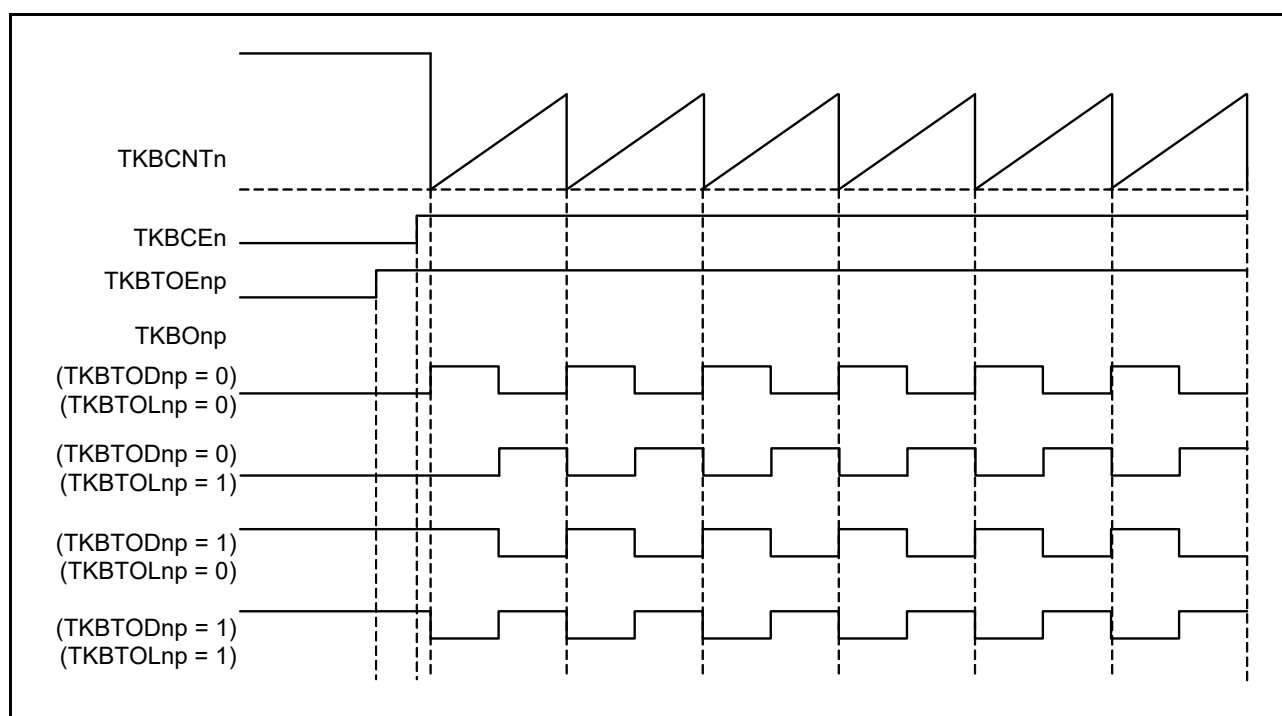
The 16-bit counter is reset to 0000H when the 16-bit counter value matches that of the TKBCRn0 register or when an external trigger input is detected if the period is determined by external triggers, and an INTTMKBn interrupt occurs.

9.4.2 Default level and active level

1. Basic operation

The 16-bit timer KB output control register n0 (TKBIOcn0) is used to set the default and active levels for the output from a 16-bit timer KB4n.

Figure 9 - 27 Basic Timing of the Default and Active Levels



Changing the TKBTOEn bit from 0 to 1 enables TKBOnp output and a PWM waveform is output according to the setting of the TKBTOlnp bit.

Changing the TKBTOEn bit from 1 to 0 disables TKBOnp output and the TKBOnp output changes to the default level specified in the TKBTOlnp bit.

2. Operation when the value of the TKBTOEnp bit is changed from 0 to 1

When the TKBTOEnp bit is changed from 0 to 1 before the value of the 16-bit timer counter KBn register (TKBCNTn) has matched the value of the 16-bit timer KB compare register np (TKBCRnp) during timer counter operation, the TKBOnp pin outputs a PWM waveform with the active level set by the TKBTOLnp bit on detection of the match.

When the TKBTOEnp bit is changed from 0 to 1 after a match between the TKBCNTn counter and TKBCRnp register, the timer output is kept at the default level until the next match.

Figure 9 - 28 Timing of the Default and Active Levels When the TKBTOEnp Bit Is Changed from 0 to 1 before a Match between the TKBCNTn Counter and Any of the TKBCRn1 to TKBCRn3 Registers

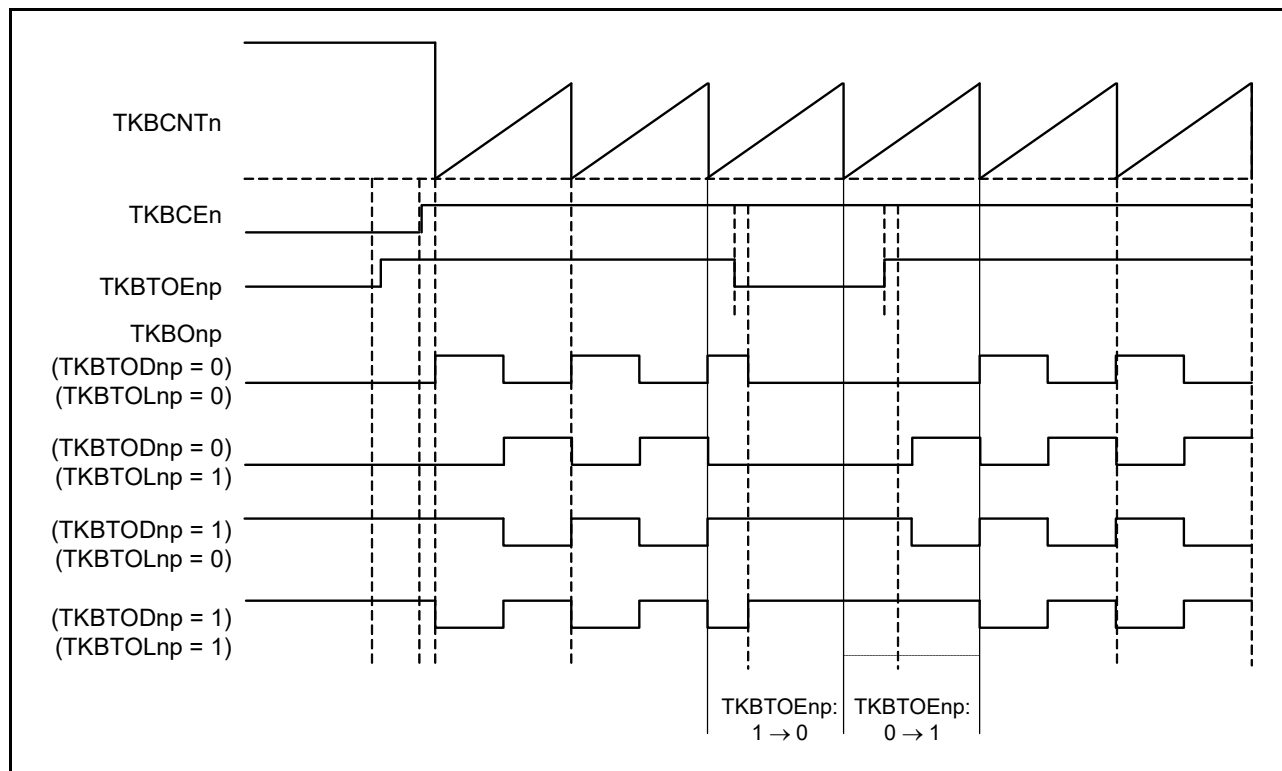
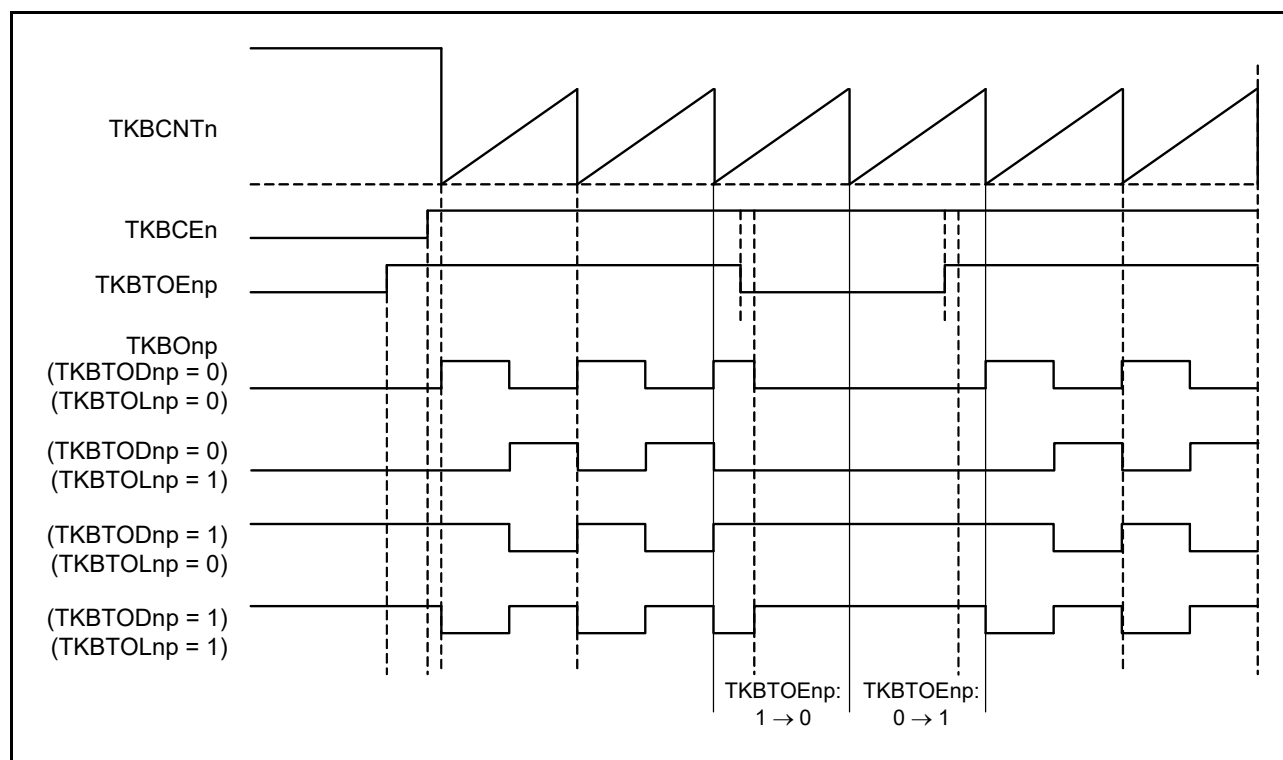


Figure 9 - 29 Timing of the Default and Active Levels When the TKBTOEnp Bit Is Changed from 0 to 1 after a Match between the TKBCNTn Counter and Any of the TKBCRn1 to TKBCRn3 Registers

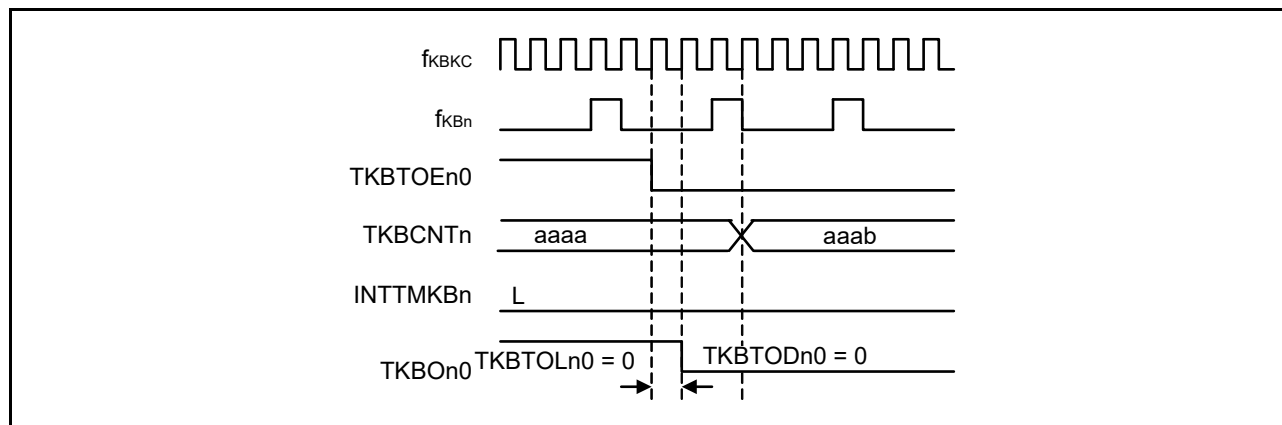


3. Operation when the value of the TKBTOEnp bit is changed from 1 to 0

a) Basic timing

Changing the TKBTOEnp bit from 1 to 0 makes the TKBOnp pin output the default level set by the TKBTOEnp bit after a cycle of the f_{KBKC} clock.

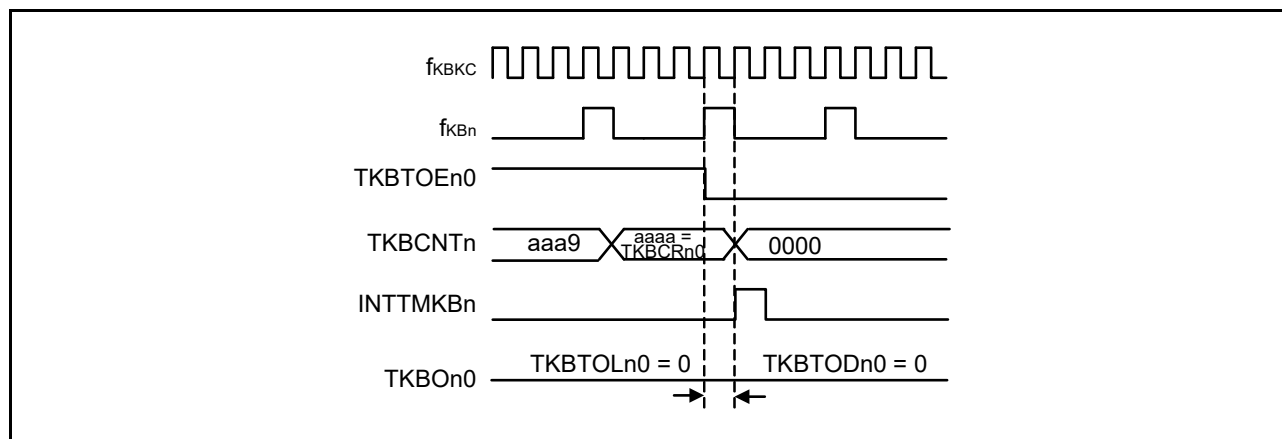
Figure 9 - 30 Timing of the Default and Active Levels When the TKBTOEn0 Bit Is Changed from 1 to 0



b) When a match with the TKBCRn0 register is generated at the same time as clearing of the TKBTOEnp bit

When the TKBTOEnp bit is changed from 1 to 0 at the same time as a match between the TKBCNTn counter and TKBCRnm register, changing of the TKBTOEnp setting is given priority over the processing in response to the match, so the TKBOnp pin outputs the default level set by the TKBTOEnp bit.

Figure 9 - 31 Timing of the Default and Active Levels When the TKBTOEnp Bit Is Changed from 1 to 0 at the Same Time as a Match between the TKBCNTn Counter and TKBCRnm Register

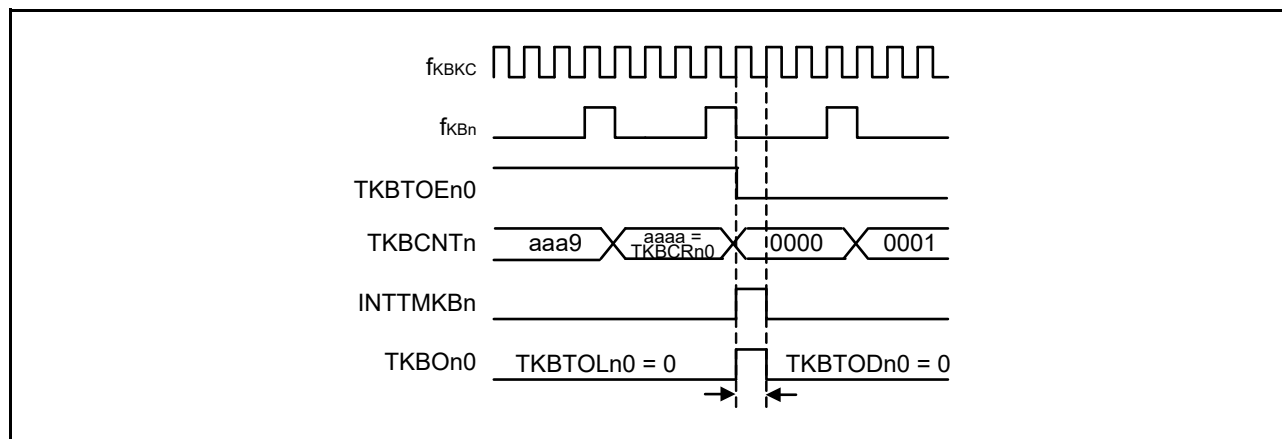


- c) When the value of the $TKBTOEn$ bit is changed at the same time as a pulse of the timer counter clock is generated

When the value of the $TKBTOEn$ bit is changed at the same time as a pulse of the f_{KBn} timer counter clock is generated, the $TKBOn$ goes to the active level on detection of a match between the $TKBCNTn$ counter and $TKBCRnm$ register.

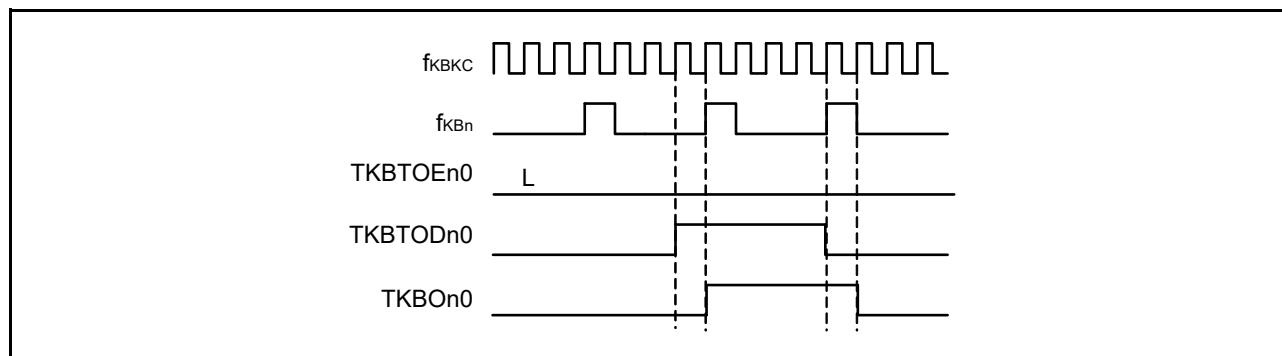
The $TKBOn$ pin outputs the default level set by the $TKBTODn$ bit after a cycle of the f_{KBK} clock.

Figure 9 - 32 Timing of the Default and Active Levels When the Value of the $TKBTOEn0$ Bit Is Changed at the Same Time as a Pulse of the Timer Counter Clock Is Generated



4. Operation when the value of the TKB TODn bit is changed while the TKB TOEn bit is set to 0
 Changing the value of the TKB TODn bit while the TKB TOEn bit is set to 0 makes the TKB On pin output the default level set by the TKB TODn bit after a cycle of the f_{KBK} clock.

Figure 9 - 33 Timing of the Default and Active Levels When the Value of the TKB TODn0 Bit Is Changed While the TKB TOEn0 Bit Is Set to 0



9.4.3 Stopping and restarting operation

Stopping and restarting operation of the 16-bit timer KB4n are controlled by the setting of the TKBCEn bit.

The 16-bit timer KB4n is reset and stops operation by changing the TKBCEn bit from 1 to 0.

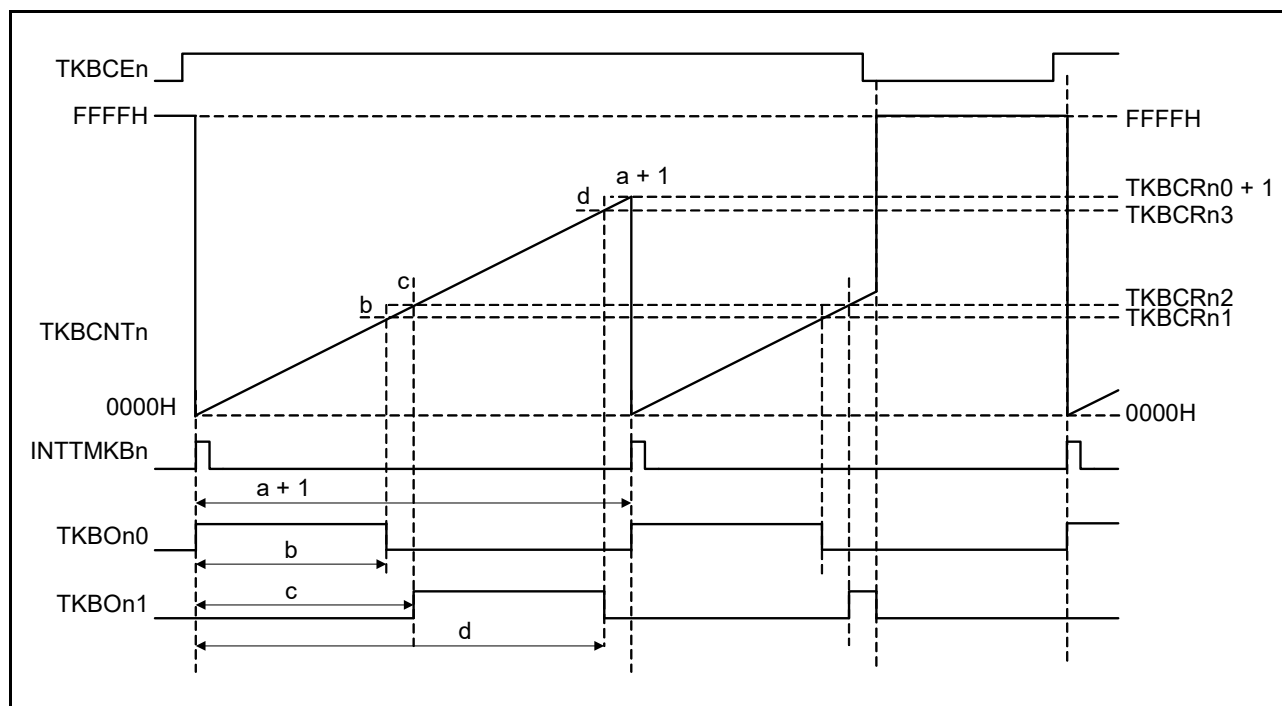
The TKBCNTn counter is reset to FFFFH and stops operation at this time.

The TKBOnp pin outputs the default level set by the TKBTODnp bit.

Changing the TKBCEn bit from 0 to 1 restarts the 16-bit timer KB4n operation.

The TKBCNTn counter register retains the value FFFFH while the TKBCEn bit is set to 0 and starts counting up by changing the TKBCEn bit from 0 to 1.

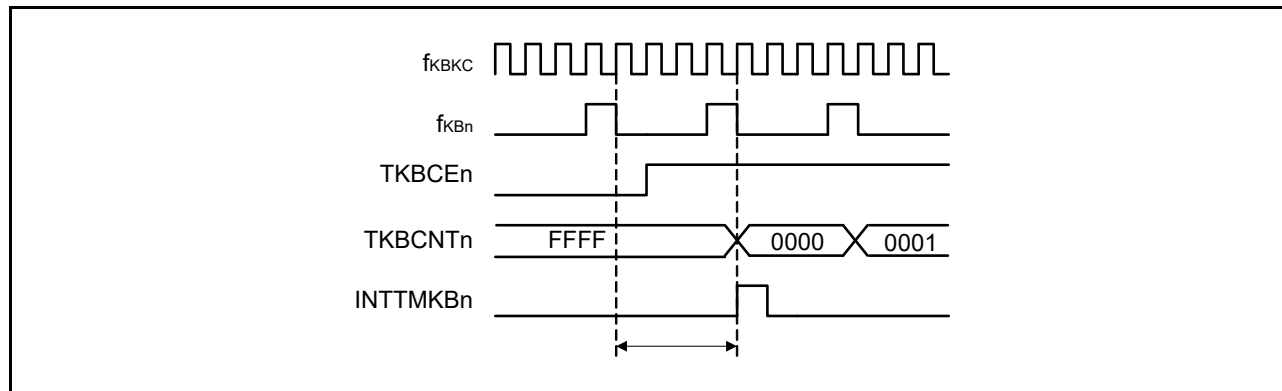
Figure 9 - 34 Timing of Operation to Stop When Both the TKBTOLnp and TKBTODnp Bits Are Set to 0



1. Timing of the start of counting

Changing the TKBCEn bit from 0 to 1 starts counting following the generation of a pulse of f_{KBn}. At the same time, an INTTMKBn interrupt is output.

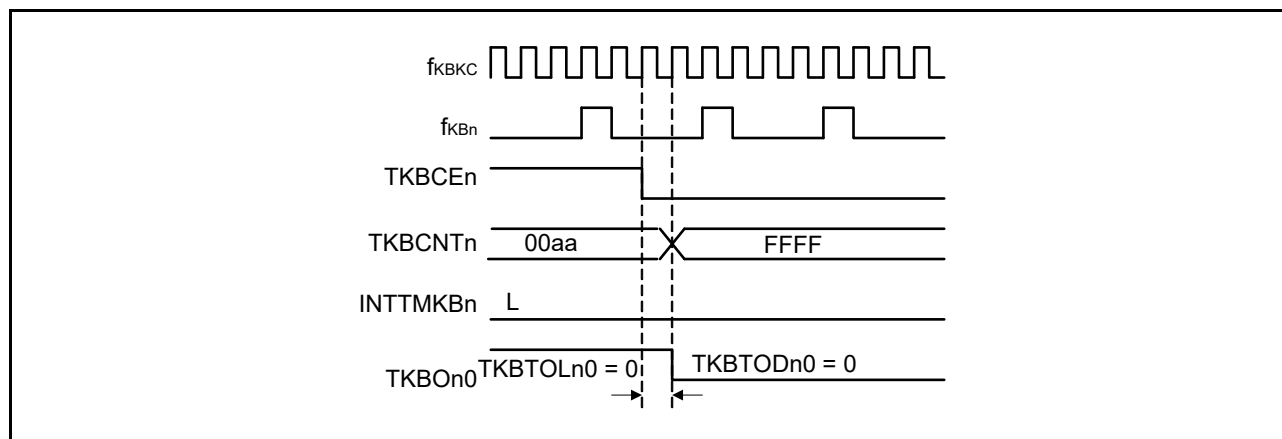
Figure 9 - 35 Timing of Operation to Start When the TKBCEn Bit Is Changed from 0 to 1



2. Timing of the stop of counting

Changing the TKBCEn bit from 1 to 0 stops counting after a cycle of the f_{KBKC} clock. It also resets the TKBCNTn counter to FFFFH and makes the TKBOnp pin output the default level set by the TKBTODnp bit.

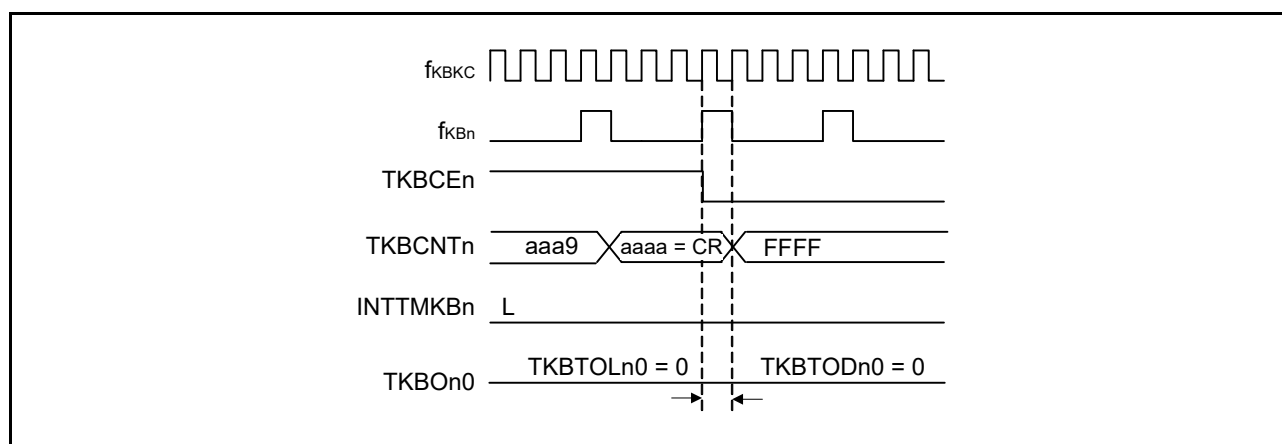
Figure 9 - 36 Timing of Operation to Stop When the TKBCE Bit Is Changed from 1 to 0



When the TKBCEn bit is changed prior to generation of the next pulse of the f_{KBn} timer counter clock, the INTTMKBn interrupt is not output and the level of the TKBOnp output does not change, even if the values of the TKBCNTn counter and TKBCRn0 register are matching at the time.

The TKBCNTn counter is reset to FFFFH and the TKBOnp pin outputs the default level set by the TKBTODnp bit after a cycle of the f_{KBKC} clock.

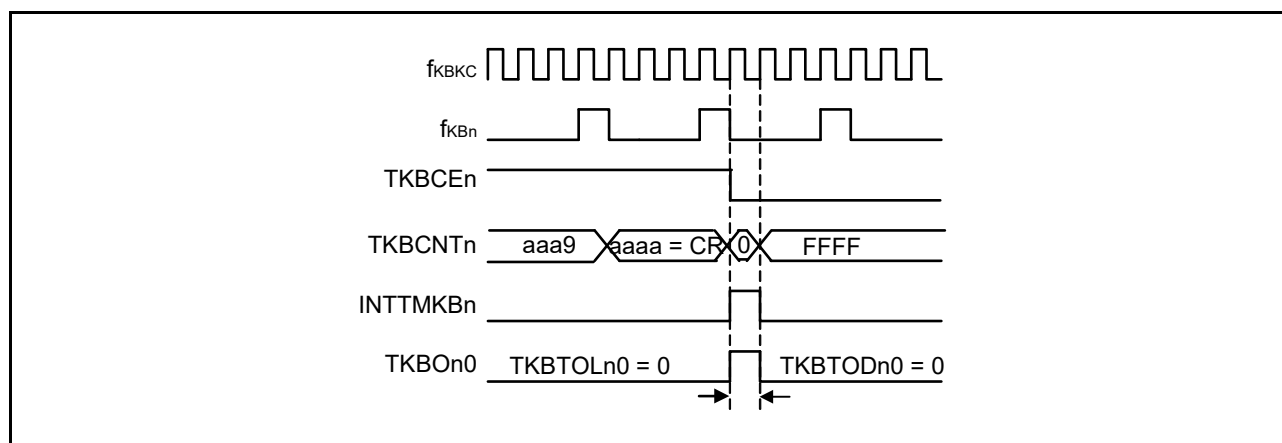
Figure 9 - 37 Timing of Operation to Stop When the Value of the TKBCEn Bit Is Changed Prior to Generation of the Next Pulse of the Timer Counter Clock



When the the TKBCEn bit is changed at the same time as a pulse of the f_{KBn} timer counter clock is generated, an INTTMKBn interrupt is output and the TKBOnp goes to the active level following a match between the TKBCNTn counter and TKBCRn0 register.

The TKBCNTn counter is reset to FFFFH and the TKBOnp pin outputs the default level set by the TKBTODnp bit after a cycle of the f_{KBKC} clock.

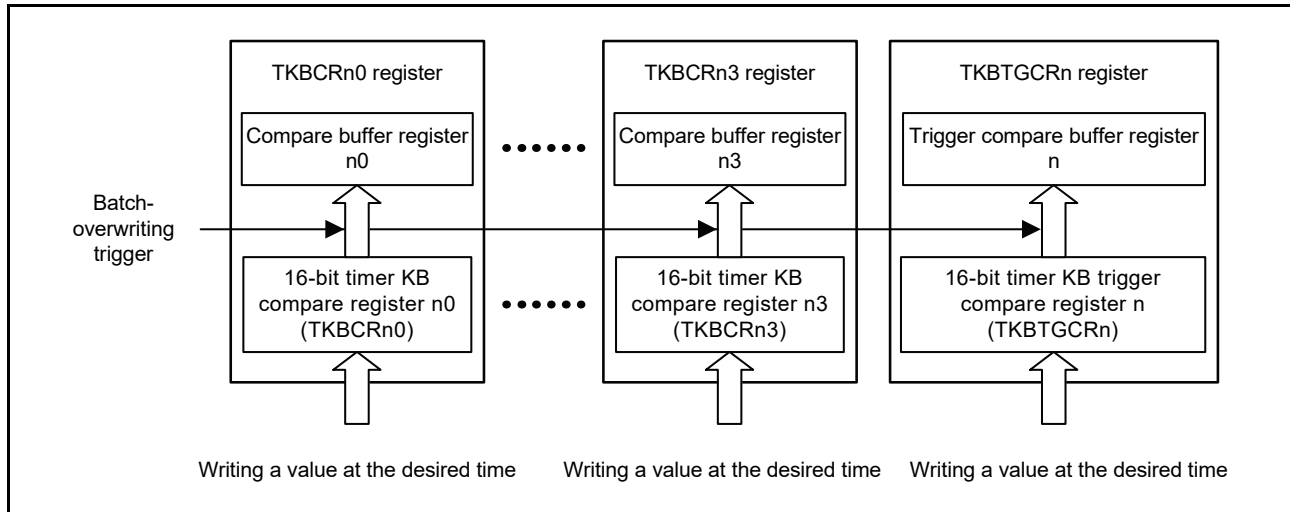
Figure 9 - 38 Timing of Operation to Stop When the Value of the TKBCEn Bit Is Changed at the Same Time as a Pulse of the Timer Counter Clock Is Generated



9.4.4 Batch overwriting

The 16-bit timer KB compare registers n p (TKBCR n p) of a 16-bit timer KB4 n each have two stages as shown in **Figure 9 - 39**. Therefore, values written to the TKBCR n p registers by software do not become effective immediately. The values written to the TKBCR n p registers with the desired timing are transferred to the buffer registers at the same time as the start of counting or on detection of a transfer trigger and then become usable in comparison. This enables writing to multiple compare registers at different times.

Figure 9 - 39 Batch Overwriting of the Compare Registers



Remark The 16-bit timer KB compare registers n p (TKBCR n p) are only handled as two-stage registers as stated above in the case of writing values; that is, they are handled as one-stage registers in any other processing. The values of the buffer registers are not directly readable.

1. Procedure for batch overwriting

- Check that the TKBRSF n flag in the 16-bit timer KB flag register n (TKBFLG n) is 0.
- Update the 16-bit timer KB compare register n p (TKBCR n p).
- Set the TKBRDT n bit in the 16-bit timer KB trigger register n (TKBTRG n) to 1.
- A batch-overwriting trigger is generated (any of the three conditions shown in **2. Timing of triggering batch overwriting** is met).

2. Timing of triggering batch overwriting

Batch overwriting can be applied to the compare registers with any of the following three timings. Of the three, timing c can be controlled by a register setting.

- The start of counting by the 16-bit timer KB4 n
- A match between the 16-bit timer counter KB n register and 16-bit timer KB compare register n 0 (TKBCR n 0)
- The occurrence of an external trigger while batch overwriting in response to an external trigger is enabled

9.4.5 Skipping of interrupt requests and A/D conversion triggers

16-bit timers KB40, KB41, KB42 can be used to control skipping of the output of interrupt request and A/D conversion trigger signals. Whether to enable or disable the output of interrupt request and A/D conversion trigger signals in the first cycle after the start of 16-bit timer KB4n operation can be specified by the TKBTMDn bit in the TKBTCTLn register. Setting the TKBTMDn bit to 1 enables the output of the interrupt and trigger signals in the first cycle.

Figure 9 - 40 shows an example of operation when the output of interrupt request and A/D conversion trigger signals is disabled in the first cycle after the start of 16-bit timer KB4n operation (TKBTCTLn.TKBTMDn = 0) in standalone mode and **Figure 9 - 41** shows an example of operation when the output is enabled (TKBTCTLn.TKBTMDn = 1).

Figure 9 - 40 Example of Operation When Output Is Disabled in the First Cycle after the Start of 16-bit Timer KB4n Operation (TKBTCTLn.TKBTMDn = 0) in Standalone Mode

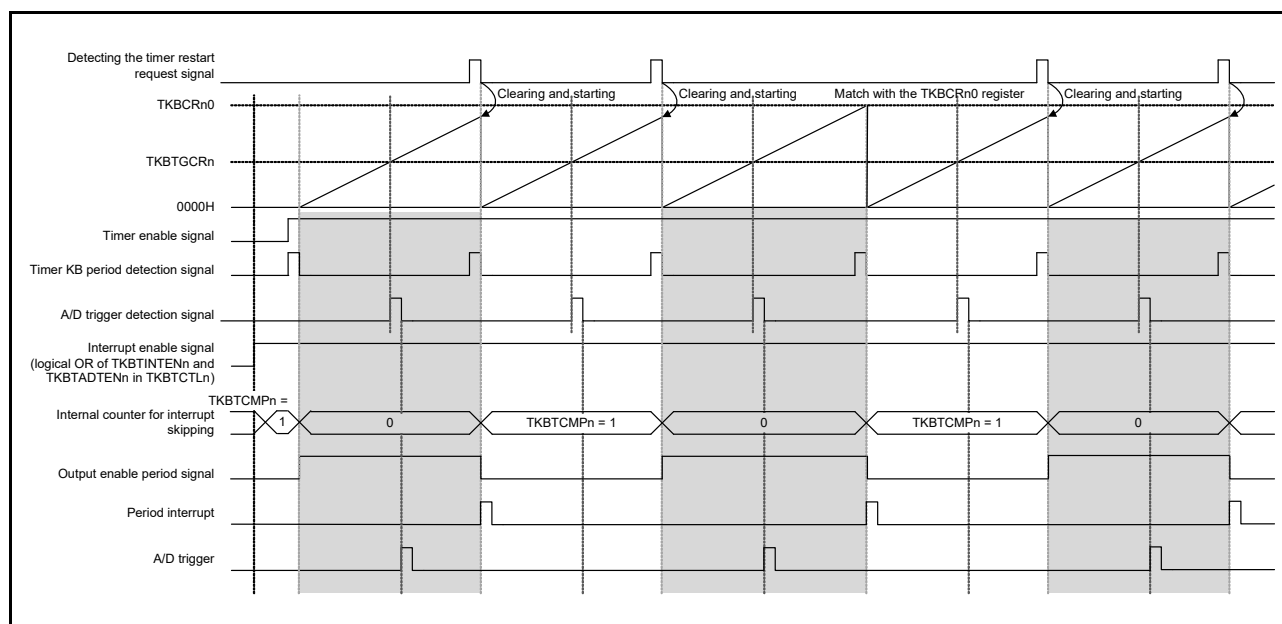
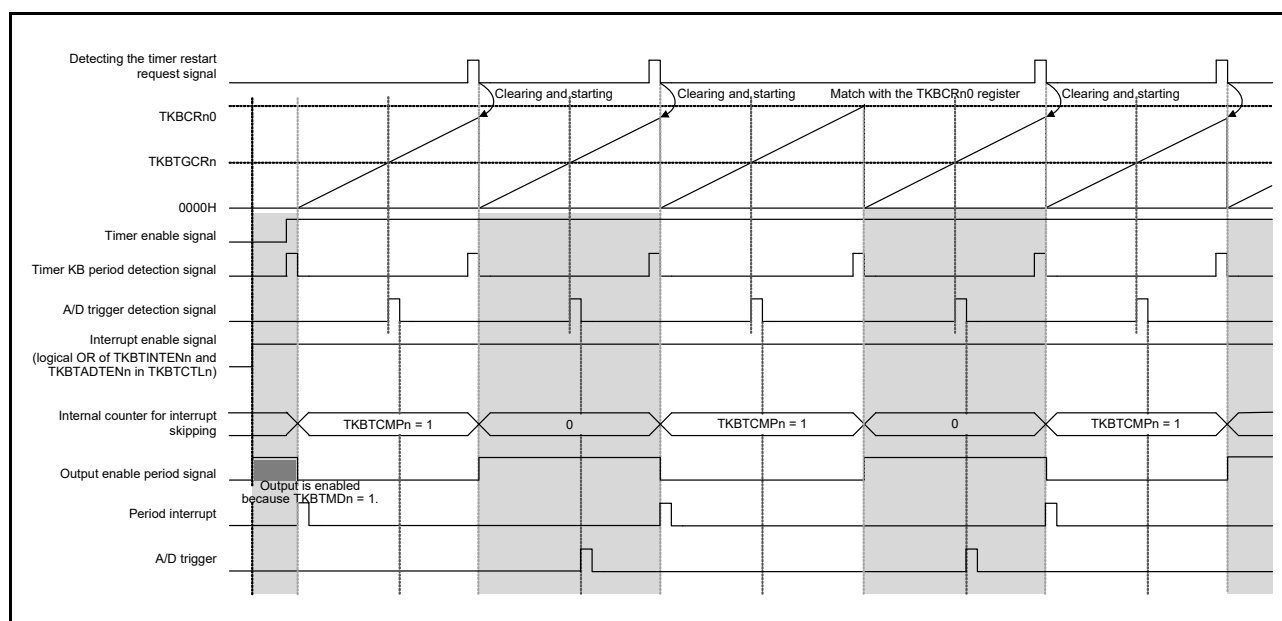


Figure 9 - 41 Example of Operation When Output Is Enabled in the First Cycle after the Start of 16-bit Timer KB4n Operation (TKBTCTLn.TKBTMDn = 1) in Standalone Mode



When a 16-bit timer KB4n is operating in simultaneous start/clear mode, counter operation is the same for both the master and slaves. Note that since skipping control can be set for each of the timers, control over skipping period interrupts and A/D triggers can be set up individually per timer. **Figure 9 - 42** shows an example of operation when output is disabled in the first cycle after the start of 16-bit timer KB4n operation ($\text{TKBTCTLn.TKBTMDn} = 0$) in simultaneous start/clear mode and **Figure 9 - 43** shows an example of operation when the output is enabled ($\text{TKBTCTLn.TKBTMDn} = 1$).

Figure 9 - 42 Example of Operation When Output Is Disabled in the First Cycle after the Start of 16-bit Timer KB4n Operation ($\text{TKBTCTLn.TKBTMDn} = 0$) in Simultaneous Start/Clear Mode (Master: Skipping of One Output Pulse; Slave: Skipping of Three Output Pulses)

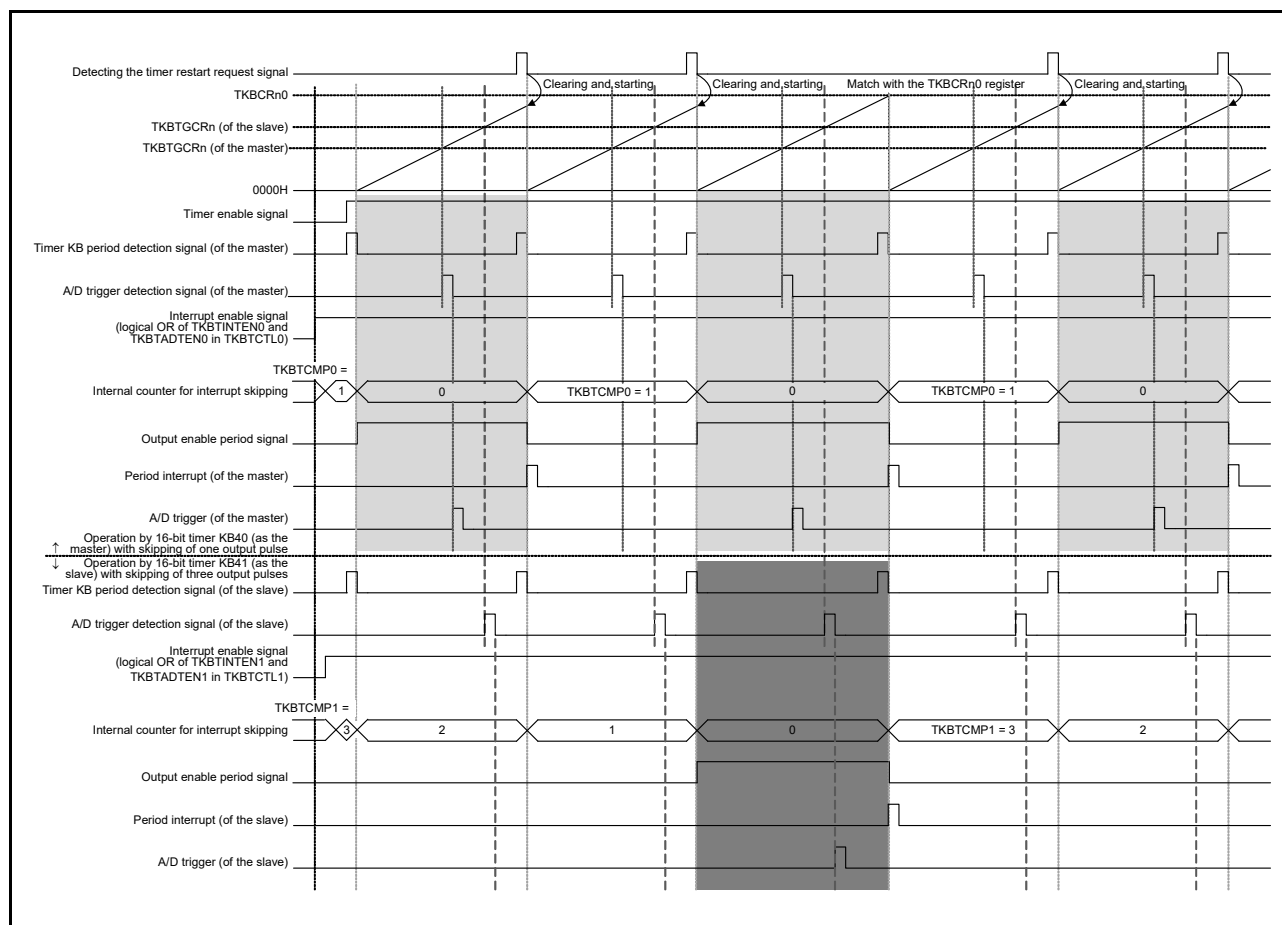
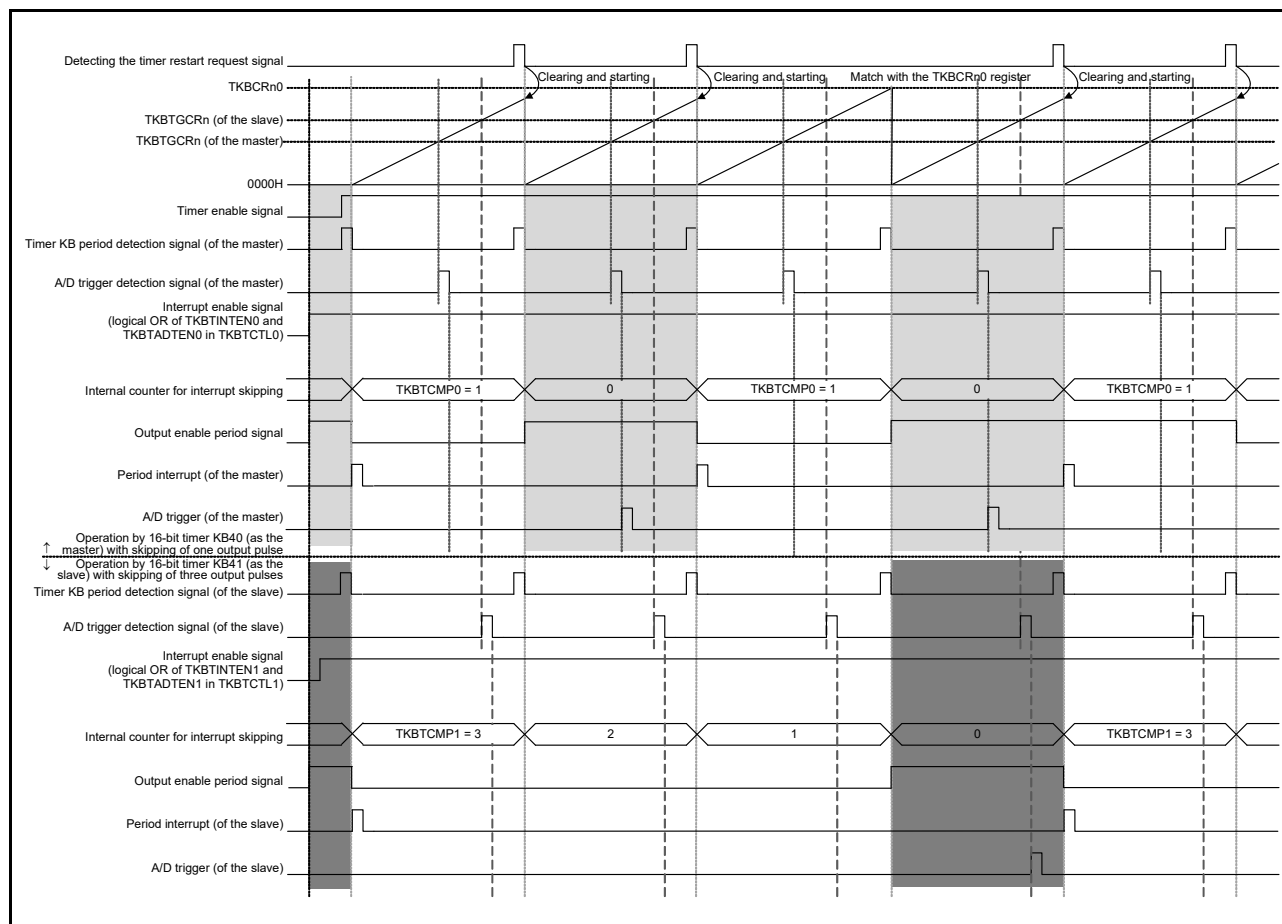


Figure 9 - 43 Example of Operation When Output Is Enabled in the First Cycle after the Start of 16-bit Timer KB4n Operation (TKBTCTLn.TKBTMDn = 1) in Simultaneous Start/Clear Mode (Master: Skipping of One Output Pulse; Slave: Skipping of Three Output Pulses)



9.4.6 Standalone mode (with the period controlled by the TKBCRn0 register)

1. Outline of functions

In standalone mode, the TKBCRn0 register is used to define the period, the TKBCRn0 and TKBCRn1 registers are used to generate TKBOn0 output, and the TKBCRn2 and TKBCRn3 registers are used to generate TKBOn1 output. The duty cycle can be set within the range from 0% to 100% and the period and duty cycle can be calculated by using the following formulas.

[Formulas for TKBOn0 output]

Pulse cycle = (TKBCRn0 setting + 1) × counter clock period

Duty cycle [%] = (TKBCRn1 setting / (TKBCRn0 setting + 1)) × 100

Output with 0% duty cycle: TKBCRn1 setting = 0000H

Output with 100% duty cycle: TKBCRn1 setting ≥ TKBCRn0 setting + 1

[Formulas for TKBOn1 output]

Duty cycle [%] = ((TKBCRn3 setting – TKBCRn2 setting) / (TKBCRn0 setting + 1)) × 100

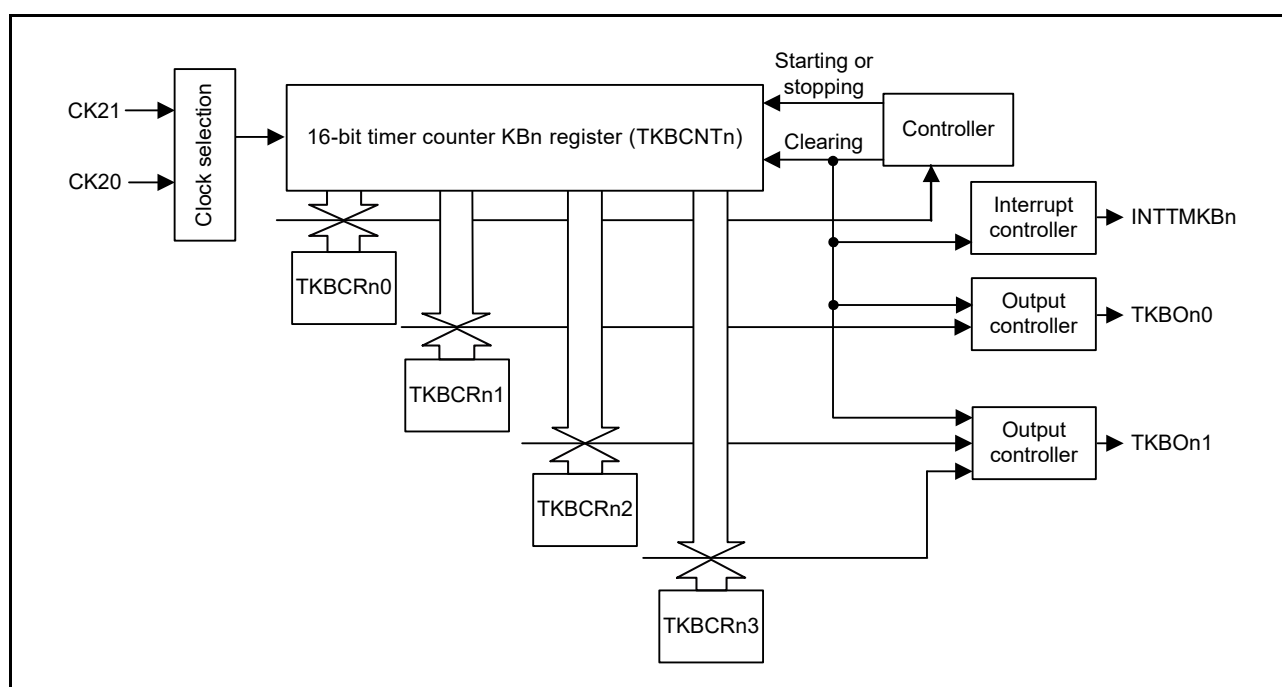
Output with 0% duty cycle: TKBCRn3 setting = TKBCRn2 setting

Output with 100% duty cycle: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution Make sure that the TKBCRn2 setting is no greater than the TKBCRn3 setting.

Figure 9 - 44 shows the configuration in standalone mode.

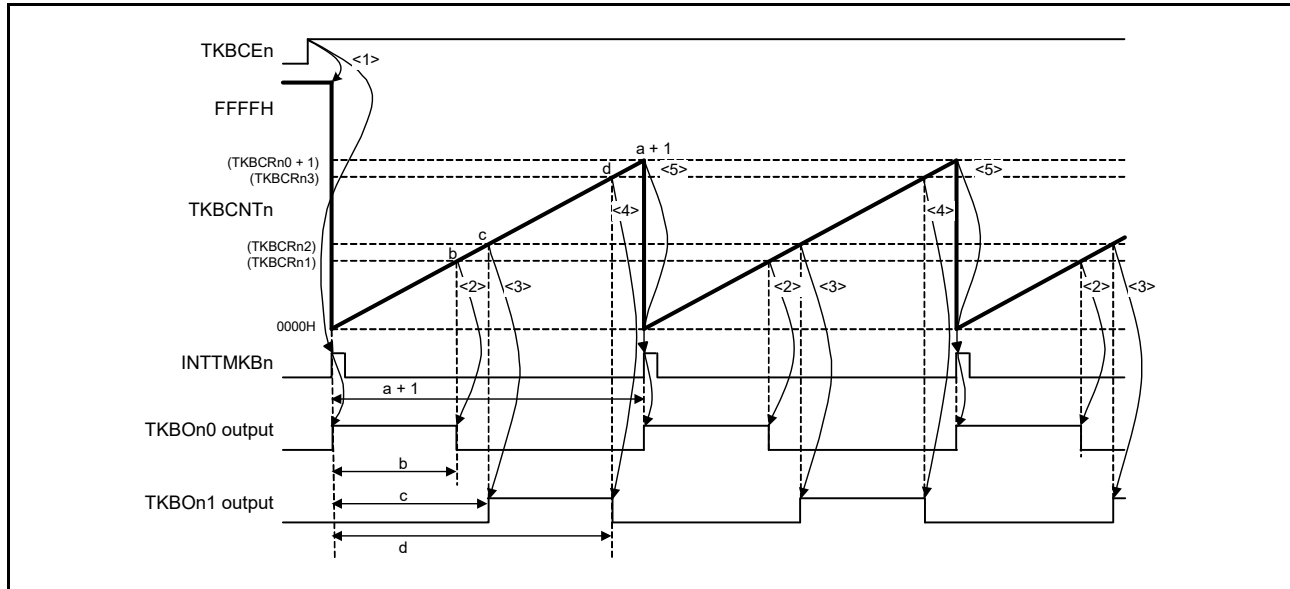
Figure 9 - 44 Configuration in Standalone Mode (with the Period Controlled by the TKBCRn0 Register)



2. Outline of operation

Figure 9 - 45 shows an example of the timing of operations in standalone mode.

Figure 9 - 45 Example of the Timing of Operations in Standalone Mode under Period Control by the TKBCRn0 Register: The Default Level for Output Is Low (TKBTODn0 = 0) and the Active Level Is High (TKBTOLn0 = 0).



The following describes an example of operations in standalone mode under period control by the TKBCRn0 register. <1> to <5> in **Figure 9 - 45** are described below.

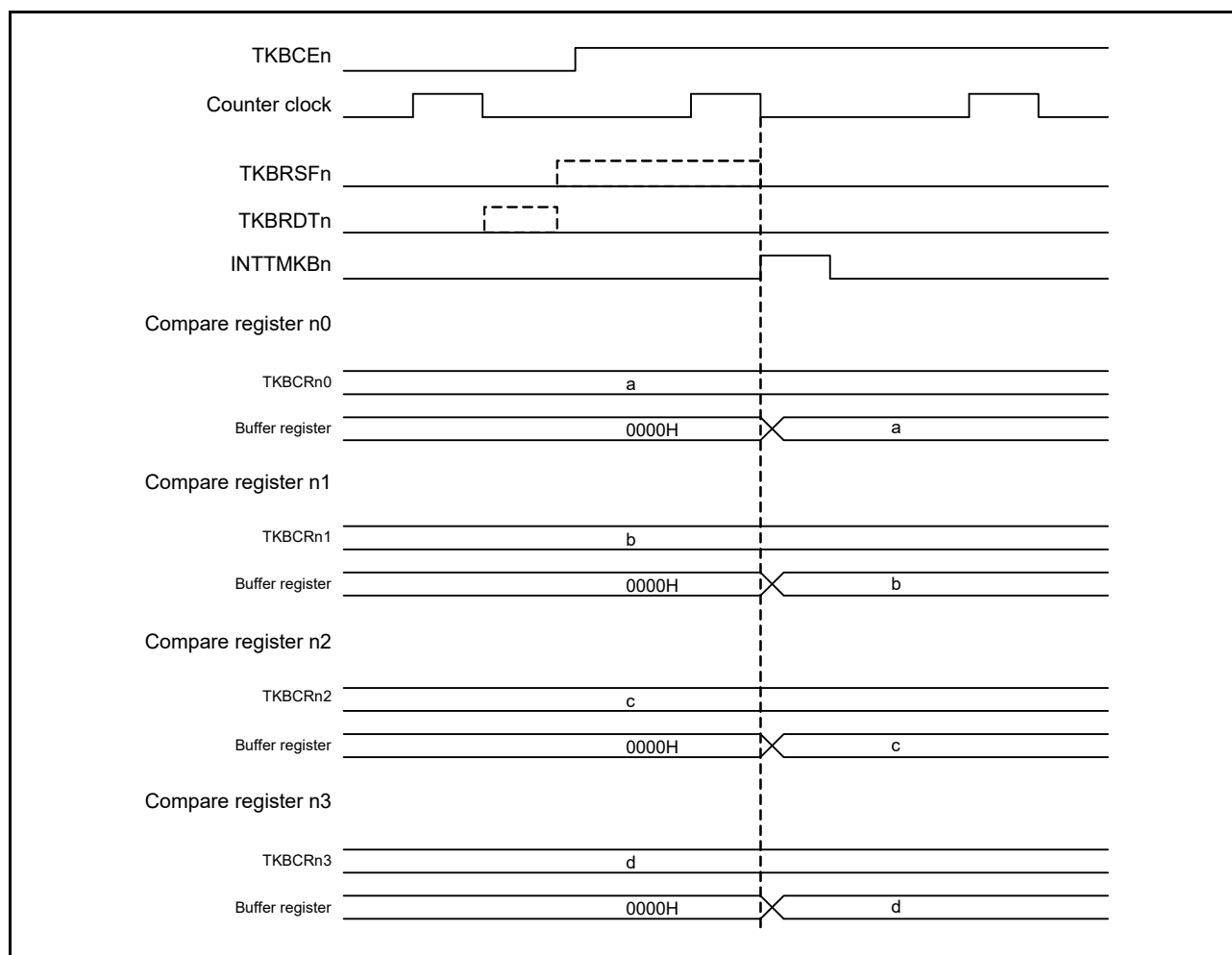
- <1> Setting the TKBCEn bit to 1 resets the 16-bit timer counter KBn register (TKBCNTn) to 0000H from FFFFH in synchronization with the counter clock and makes the counter start counting up. At the same time, an INTTMKBn interrupt is output and the TKBOn0 output changes from the default level specified in the TKBTODn0 bit of the TKBIOCn0 register to the active level (high in this example) specified in the TKBTOLn0 bit. Note that the TKBOn1 output is kept at the default level specified in the TKBTODn1 bit.
- <2> When the value of the TKBCNTn counter matches the value specified in 16-bit timer KB compare register n1 (TKBCRn1), the TKBOn0 output goes to the inactive level.
- <3> When the value of the TKBCNTn counter matches the value specified in 16-bit timer KB compare register n2 (TKBCRn2), the TKBOn1 output goes to the active level.
- <4> When the value of the TKBCNTn counter matches the value specified in 16-bit timer KB compare register n3 (TKBCRn3), the TKBOn1 output goes to the inactive level.
- <5> When the value of the TKBCNTn counter matches the value specified in 16-bit timer KB compare register n0 (TKBCRn0), an INTTMKBn interrupt is output on the next cycle of the counter clock and the TKBOn0 output goes to the active level. The TKBCNTn counter starts counting up from 0000H.
- <6> The above operations <2> to <5> are repeated.

3. Operation of batch overwriting when the counter starts operating

The internal buffer registers of the compare registers in the 16-bit timer KB4n are updated at the same time as counter operation starts following a counter clock input generated after 1 has been written to the TKBCEn bit of the TKBCTLn1 register.

In the case of batch overwriting at the time counter operation starts, and only in that case, writing 1 to the TKBRDTn bit of the TKBTRGn register is not required (see **Figure 9 - 46**).

Figure 9 - 46 Batch Overwriting: Timing of Updating the Buffers When the Counter Starts Operating



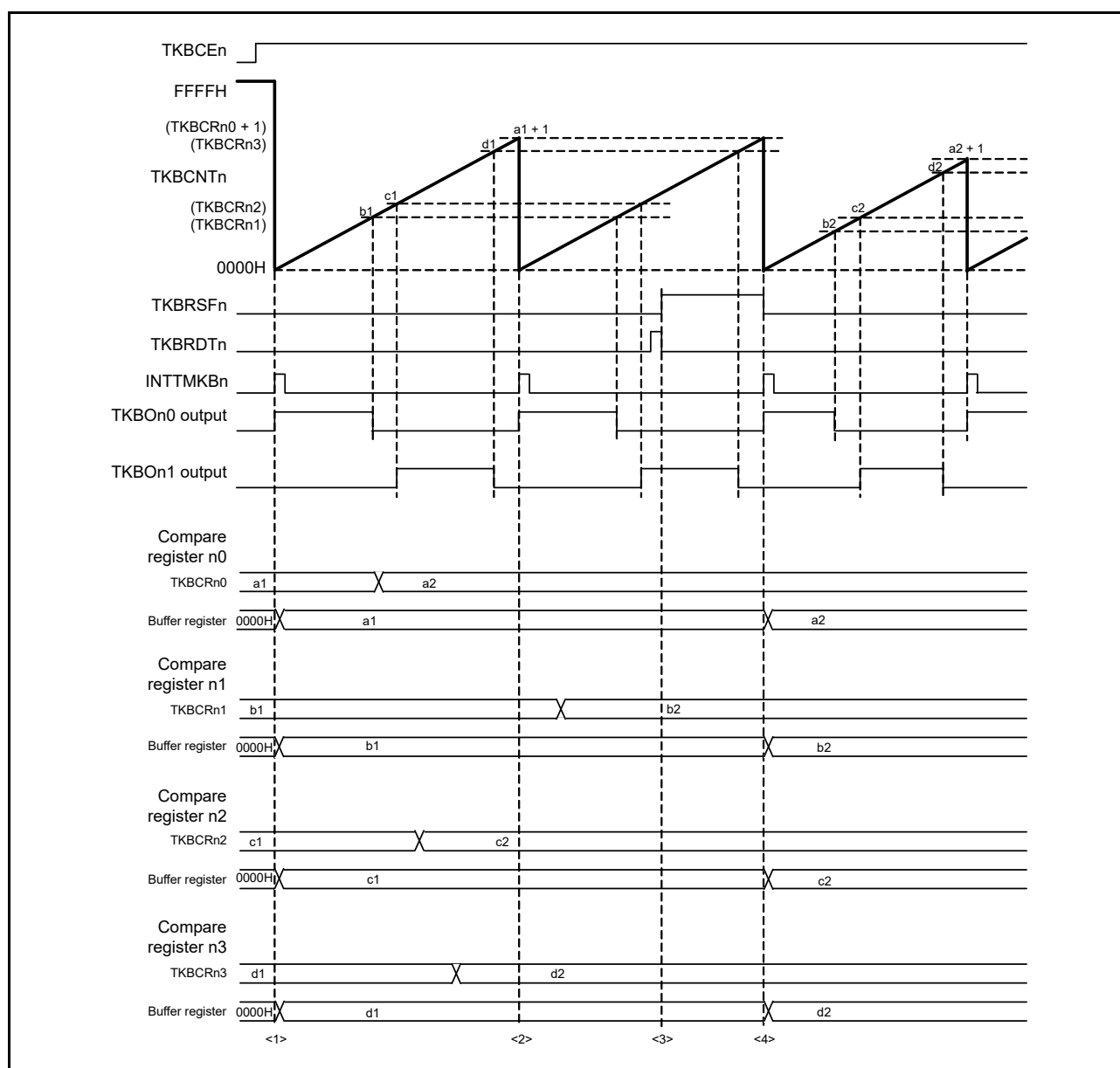
Remark Writing 1 to the TKBRDTn bit while the value of the TKBCEn bit is 0 sets the TKBRSFn flag to 1. The TKBRSFn flag is cleared to 0 at the time of the start of counting (generation of the trigger to start counting).

4. Operation of batch overwriting when the buffer registers are updated during counting

Batch overwriting of the compare registers in a 16-bit timer KB4n is triggered by writing 1 to the TKBRDTn bit and their internal buffer registers are updated simultaneously with the next time the counter is cleared (in response to a match between the TKBCNTn counter and TKBCRn0 register). The TKBRSFn flag is provided to indicate the period from writing of 1 to the TKBRDTn bit until the completion of batch overwriting (see **Figure 9 - 47**).

- <1> The settings of the compare registers are transferred to their buffer registers at the same time as the start of counting by the TKBCNTn counter after the TKBCEn bit has been changed from 0 to 1.
- <2> If the value of the TKBRDTn bit is not 1, batch overwriting does not proceed when the counter is cleared after the TKBCRnm and TKBTGCRn registers have been overwritten.
- <3> Writing 1 to the TKBRDTn bit sets the batch overwriting pending flag (the TKBRSFn flag) to 1.
- <4> The settings of the compare registers are transferred to their buffer registers in response to counter clearing while the value of the TKBRSFn flag is 1. At the same time, the value of the TKBRSFn flag becomes 0.

Figure 9 - 47 Batch Overwriting: Timing of Updating the Buffers during Counting



5. Sample register settings in standalone mode (with the period controlled by the TKBCRn0 register)

TKBCTLn0	15	14	13	12	11	10	9	8
	TKBTGMn 1 0	— 0	TKBSSEn1 1/0	TKBDIE1 1/0	TKBTGMn 0 0	— 0	TKBSSEn0 1/0	TKBDIE0 1/0
TKBCTLn1	7	6	5	4	3	2	1	0
	TKBMFE1 0	TKBIRSn2 1/0	TKBIRSn1 0	TKBIRSn0 0	TKBIHE1 0	TKBTSE1 0	TKBSTSn1 0	TKBSTSn0 0
TKBCTLn1	7	6	5	4	3	2	1	0
	TKBCE1 1/0	— 0	— 0	TKBCKSn 1/0	TKBSCM1 0	— 0	TKBMDn1 0	TKBMDn0 0
TKBIOcn0	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
TKBIOcn1	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
TKBCTLn2	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
TKBCTLn2	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINSn2 0	TKBINSn1 0	TKBINSn0 0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H to FFFFH
TKBSIRn1	0000H to FFFFH
TKBSSRn0	00H to 0FH
TKBSSRn1	00H to 0FH
TKBDNRn0	00H to F0H
TKBDNRn1	00H to F0H
TKBMFRn	0000H



: The setting is fixed in this mode.



: Setting a value is not required (leave it as the default value).

9.4.7 Standalone mode (with the period controlled by external trigger input)

1. Outline of functions

The standalone mode allows period control not only by the TKBCRn0 register but also by an external trigger input (timer restart). Using this function enables the implementation of, for example, PFC control in the critical conduction mode.

For external trigger input detection, the input signal selected by setting the following bits is used.

- The TKBSTSn[1:0] bits in the 16-bit timer KB operation control register n0
- The TKBINSn[2:0] bits in the 16-bit timer KB operation control register n2

When an external trigger input is detected, the TKBCNTn counter is cleared to 0000H and the TKBOn0 and TKBOn1 outputs respectively go to the active level and inactive level. When the values of the TKBCRn0 register and the TKBCNTn counter match before detecting an external trigger input, the TKBCNTn counter is cleared to 0000H and continues counting.

For the formulas for TKBOn0 and TKBOn1 outputs when no external trigger input is detected under period control by the TKBCRn0 register, see **9.4.6 Standalone mode (with the period controlled by the TKBCRn0 register)**.

The formulas for TKBOn0 and TKBOn1 outputs under period control by external trigger input are as follows:

[Formulas for TKBOn0 output]

Pulse cycle = (counter value on detection of external trigger input + 1) × counter clock period

Duty cycle [%] = (TKBCRn1 setting / (counter value on detection of external trigger input + 1)) × 100

Output with 0% duty cycle: TKBCRn1 setting = 0000H

Output with 100% duty cycle: TKBCRn1 setting ≥ counter value on detection of external trigger input + 1

[Formulas for TKBOn1 output]

Pulse cycle = (counter value on detection of external trigger input + 1) × counter clock period

Duty cycle [%] = ((TKBCRn3 setting – TKBCRn2 setting) / (counter value on detection of external trigger input + 1)) × 100

Output with 0% duty cycle: TKBCRn3 setting = TKBCRn2 setting

Output with 100% duty cycle: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ counter value on detection of external trigger input + 1

Caution Make sure that the TKBCRn2 setting is no greater than the TKBCRn3 setting.

Figure 9 - 48 shows the configuration in standalone mode (with the period controlled by external trigger input) and **Table 9 - 3** lists the assignment of external triggers in standalone mode (with the period controlled by external trigger input).

Figure 9 - 48 Configuration in Standalone Mode (with the Period Controlled by External Trigger Input)

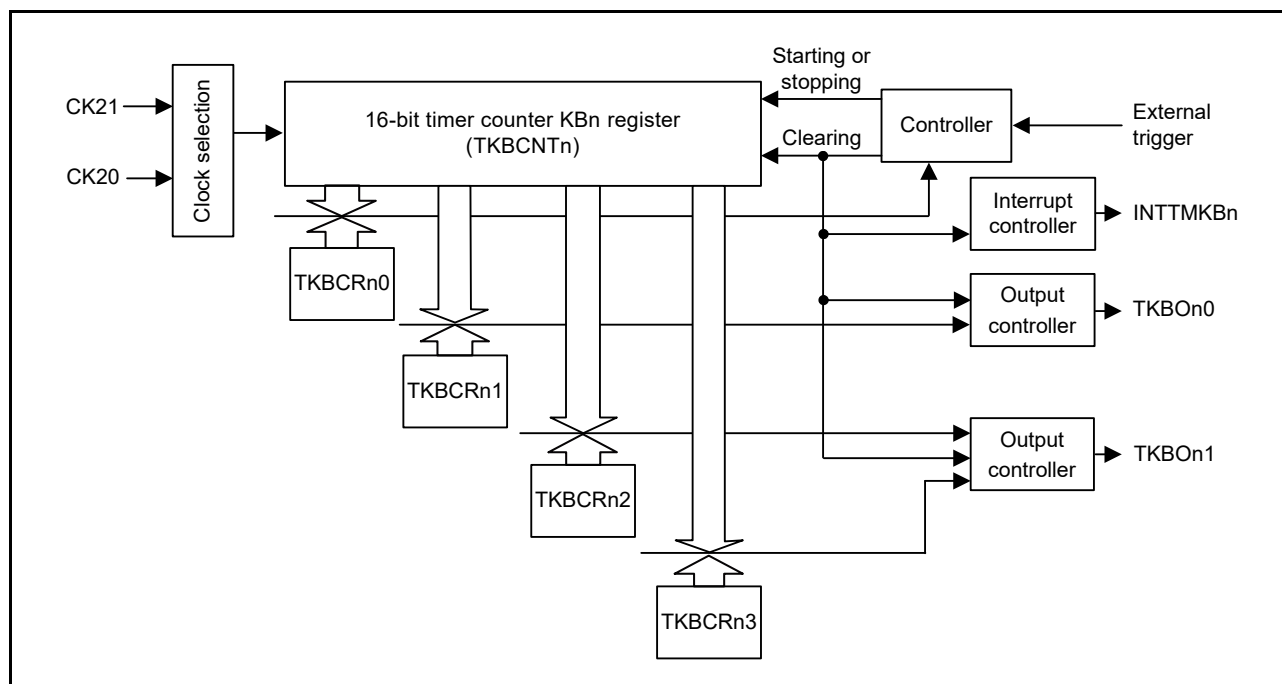


Table 9 - 3 Assignment of External Triggers in Standalone Mode (with the Period Controlled by External Trigger Input)

Item	16-bit Timer KB40	16-bit Timer KB41	16-bit Timer KB42
INTP0	✓	✓	✓
INTP1	✓	✓	✓
INTP2	✓	✓	✓
INTP3	✓	✓	✓
INTP4	✓	✓	✓
INTP5	✓	✓	✓
INTP6	✓	✓	✓
INTP7	✓	✓	✓
Comparator 0	✓	✓	✓
Comparator 1	✓	✓	✓

2. Operation of batch overwriting when the buffer registers are updated during counting under period control by external trigger input in standalone mode while the TKBTSn bit is set to 1

Setting the TKBTSn bit of the TKBCTLn0 register to 1 under period control by external trigger input in standalone mode enables clearing of the counter and batch overwriting of the compare registers on an external trigger input after 1 has been written to the TKBRDTn bit.

In the same way as clearing of the counter, batch overwriting also proceeds on an external trigger input following a match between the TKBCRn0 register and TKBCNTn counter after 1 has been written to the TKBRDTn bit.

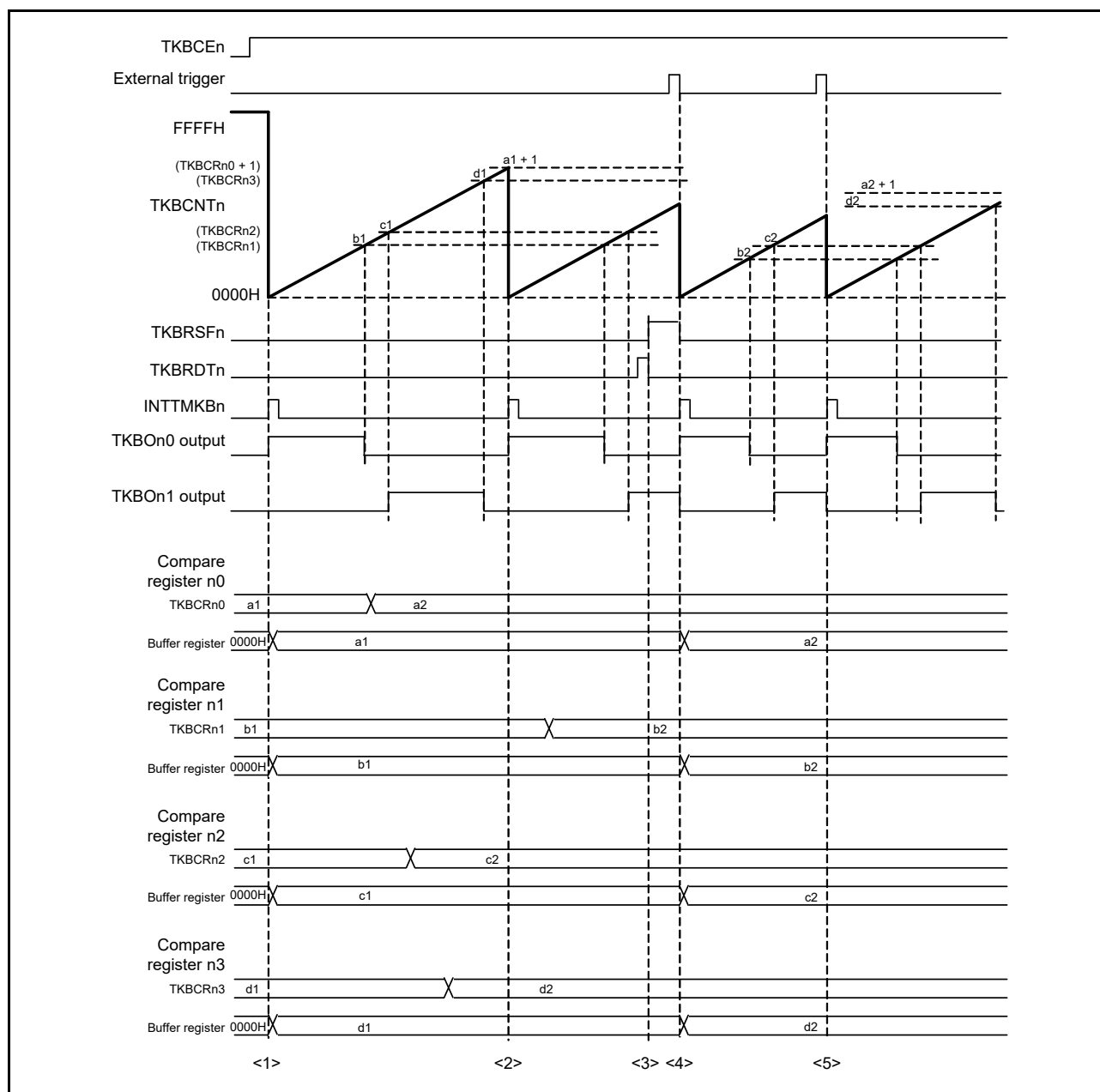
The source of the external trigger input is selected by setting the following bits.

- The TKBSTSn[1:0] bits in the TKBCTLn0 register
- The TKBINSn[2:0] bits in the TKBCTLn2 register

Figure 9 - 49 shows an example of the timing of batch overwriting operation when the TKBTSn bit is set to 1.

- <1> The settings of the compare registers are transferred to their buffer registers at the same time as the start of counting by the TKBCNTn counter after the TKBCEn bit has been changed from 0 to 1.
- <2> If the value of the TKBRDTn bit is not 1, batch overwriting does not proceed when the counter is cleared after the TKBCRnm and TKBTGCRn registers have been overwritten.
- <3> Writing 1 to the TKBRDTn bit sets the batch overwriting pending flag (the TKBRSFn flag) to 1.
- <4> The settings of the compare registers are transferred to their buffer registers in response to counter clearing following an external trigger input while the TKBTSn bit and TKBRSFn flag are set to 1. At the same time, the value of the TKBRSFn flag becomes 0.
- <5> If the value of the TKBRDTn bit is not 1, batch overwriting does not proceed when the counter is cleared following an external trigger input.

Figure 9 - 49 Batch Overwriting: Timing of Updating the Buffers during Counting under Period Control by External Trigger Input in Standalone Mode When the TKBTSn Bit Is Set to 1



3. Operation of batch overwriting when the buffer registers are updated during counting under period control by external trigger input in standalone mode while the TKBTSn bit is set to 0

The following is an example of updating the buffers under period control by external trigger input in standalone mode when the TKBTSn bit of the TKBCTLn0 register is set to 0. In this case, when 1 is written to the TKBRDTn bit and the value of the batch overwriting pending flag (the TKBRSFn flag) is 1, the counter is cleared on detection of external trigger input, but batch overwriting of the compare registers does not proceed at this time.

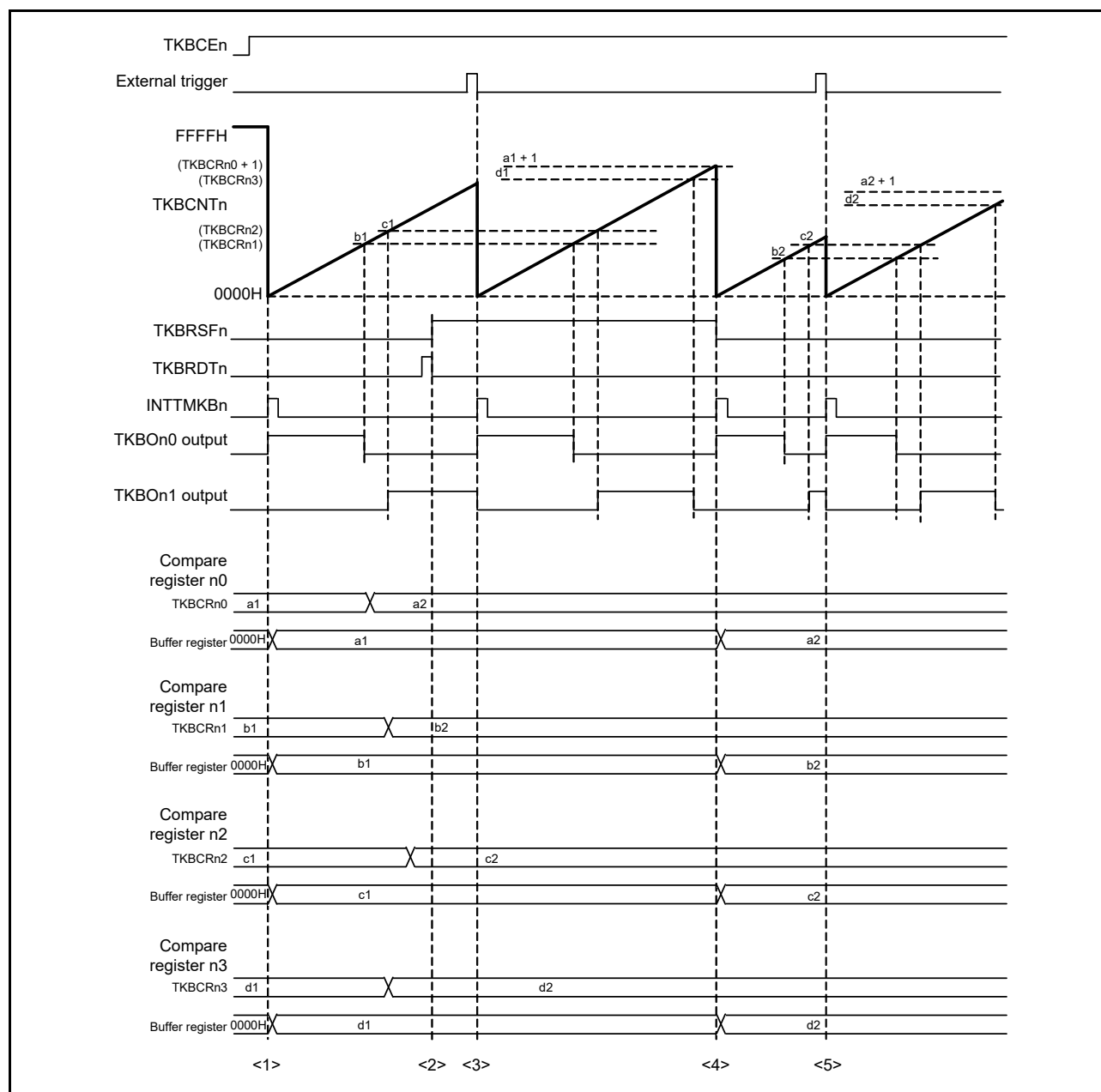
The source of the external trigger input is selected by setting the following bits.

- The TKBSTSn[1:0] bits in the TKBCTLn0 register
- The TKBINSn[2:0] bits in the TKBCTLn2 register

Figure 9 - 50 shows an example of the timing of batch overwriting operation when the TKBTSn bit is set to 0.

- <1> The settings of the compare registers are transferred to their buffer registers at the same time as the start of counting by the TKBCNTn counter after the TKBCEn bit has been changed from 0 to 1.
- <2> Writing 1 to the TKBRDTn bit after the TKBCRnm and TKBTGCRn registers have been overwritten sets the batch overwriting pending flag (the TKBRSFn flag) to 1.
- <3> If the value of the TKBTSn bit is not 1, batch overwriting does not proceed when the counter is cleared on an external trigger input.
- <4> The settings of the compare registers are transferred to their buffer registers in response to counter clearing following a match between the TKBCNTn counter and TKBCRn0 register while the TKBRSFn flag is set to 1. At the same time, the value of the TKBRSFn flag becomes 0.
- <5> If the values of the TKBTSn bit and TKBRSFn flag are not 1, batch overwriting does not proceed when the counter is cleared following an external trigger input.

Figure 9 - 50 Batch Overwriting: Timing of Updating the Buffers during Counting under Period Control by External Trigger Input in Standalone Mode When the TKBTSEn Bit Is Set to 0



4. Sample register settings in standalone mode (with the period controlled by external trigger input)

TKBCTLn0	15	14	13	12	11	10	9	8
	TKBTGMn 1 1/0	— 0	TKBSSEn1 1/0	TKBDIE n1 1/0	TKBTGMn 0 1/0	— 0	TKBSSEn0 1/0	TKBDIE n0 1/0
TKBCTLn1	7	6	5	4	3	2	1	0
	TKBMFEn 1/0	TKBIRSn2 1/0	TKBIRSn1 0	TKBIRSn0 0	TKBIHE n 1/0	TKBTSE n 1/0	TKBSTSn1 1/0	TKBSTSn0 1/0
TKBCTLn1	7	6	5	4	3	2	1	0
	TKBCE n 1/0	— 0	— 0	TKBCKSn 1/0	TKBSCMn 0	— 0	TKBMDn1 0	TKBMDn0 0
TKBIOcn0	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
TKBIOcn1	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
TKBCTLn2	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
TKBCTLn2	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINSn2 1/0	TKBINSn1 1/0	TKBINSn0 1/0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H to FFFFH
TKBSIRn1	0000H to FFFFH
TKBSSRn0	00H to 0FH
TKBSSRn1	00H to 0FH
TKBDNRn0	00H to F0H
TKBDNRn1	00H to F0H
TKBMFRn	0000H to FFFFH



: The setting is fixed in this mode.



: Setting a value is not required (leave it as the default value).

9.4.8 Simultaneous start/stop mode

1. Outline of functions

When multiple KB4n 16-bit timers are to be used as a master and slaves, the slave KB4m 16-bit timers can be started and stopped at the same times as the master KB40 16-bit timer is started and stopped. In this case, select the standalone mode (TKBMD0[1:0] = 00B) for the master and the simultaneous start/stop mode (TKBMD0[1:0] = 01B) for the slaves.

The simultaneous start/stop mode only handles synchronization of the timing with which the master and slaves start and stop.

When different counter clocks (CK0 and CK1) are selected for the master and slaves, setting the TKBSCM0 bit of the master to 1 enables the master and slaves to start counting at the same time. After that, the master and slave timers operate individually.

When the KB41 and KB42 timers operate as slaves with different clocks, set both the TKBSCM0 bit of the KB40 timer and the TKBSCMm bit of the slave for which the same clock as the master is selected to 1.

Caution 1. Only the KB40 16-bit timer can operate as the master.

Caution 2. The clock selected for the master must be faster than or at the same frequency as that for the slaves.

Relationship of Selected Clocks between the Master and Slaves	Relationship between CK0 and CK1	TKBSCMn Bit	Availability
Selecting the same clock for the master and slaves	—	0	✓
Selecting different clocks for the master and slaves	The clock selected for the master is faster than that selected for the slaves.	1	✓
Selecting different clocks for the master and slaves	The clock selected for the master is slower than that selected for the slaves.	—	×

For the formulas to calculate TKBOn0 and TKBOn1 outputs in simultaneous start/stop mode, see **9.4.6 Standalone mode (with the period controlled by the TKBCRn0 register)** and **9.4.7 Standalone mode (with the period controlled by external trigger input)**.

Remark n = 0 to 2; m = 1, 2

2. Simultaneous start/stop mode

Master: Sample register settings in standalone mode

TKBCTL00	15	14	13	12	11	10	9	8
	TKBTGM0 1 1/0	— 0	TKBSSE01 1/0	TKBDIE01 1/0	TKBTGM0 0 1/0	— 0	TKBSSE00 1/0	TKBDIE00 1/0
TKBCTL01	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	TKBIRS02 1/0	TKBIRS01 0	TKBIRS00 0	TKBIHE0 1/0	TKBTSE0 1/0	TKBSTS01 1/0	TKBSTS00 1/0
TKBIOC00	7	6	5	4	3	2	1	0
	TKBCE0 1/0	— 0	— 0	TKBCKS0 1/0	TKBSCM0 1/0	— 0	TKBMD01 0	TKBMD00 0
TKBIOC01	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOL01 1/0	TKBTOL00 1/0	TKBTOD01 1/0	TKBTOD00 1/0
TKBCTL02	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOE01 1/0	TKBTOE00 1/0
TKBCTL02	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
TKBCTL02	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINS02 1/0	TKBINS01 1/0	TKBINS00 1/0

TKBCR00	0000H to FFFFH
TKBCR01	0000H to FFFFH
TKBCR02	0000H to FFFFH
TKBCR03	0000H to FFFFH
TKBTGCR0	0000H to FFFFH
TKBSIR00	0000H to FFFFH
TKBSIR01	0000H to FFFFH
TKBSSR00	00H to 0FH
TKBSSR01	00H to 0FH
TKBDNR00	00H to F0H
TKBDNR01	00H to F0H
TKBMFR0	0000H to FFFFH



: The setting is fixed in this mode.



: Setting a value is not required (leave it as the default value).

3. Simultaneous start/stop mode

Slaves: Sample register settings in simultaneous start/stop mode

TKBCTLm0	15	14	13	12	11	10	9	8
	TKBTGMm 1 1/0	— 0	TKBSSEm 1 1/0	TKBDIEm1 1/0	TKBTGMm 0 1/0	— 0	TKBSSEm 0 1/0	TKBDIEm0 1/0
	7	6	5	4	3	2	1	0
	TKBMFEEm 1/0	TKBIRSm2 1/0	TKBIRSm1 0	TKBIRSm0 0	TKBIHEm 1/0	TKBTSEm 1/0	TKBSTSm 1 1/0	TKBSTSm 0 1/0
TKBCTLm1	7	6	5	4	3	2	1	0
	TKBCEm 0	— 0	— 0	TKBCKSm 1/0	TKBSCMm 1/0	— 0	TKBMDm1 0	TKBMDm0 1
TKBIOCm0	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOLm1 1/0	TKBTOLm0 1/0	TKBTODm 1 1/0	TKBTODm 0 1/0
TKBIOCm1	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEm 1 1/0	TKBTOEm 0 1/0
TKBCTLm2	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINSm2 1/0	TKBINSm1 1/0	TKBINSm0 1/0

TKBCRm0	0000H to FFFFH
TKBCRm1	0000H to FFFFH
TKBCRm2	0000H to FFFFH
TKBCRm3	0000H to FFFFH
TKBTGCRm	0000H to FFFFH
TKBSIRm0	0000H to FFFFH
TKBSIRm1	0000H to FFFFH
TKBSSRm0	00H to 0FH
TKBSSRm1	00H to 0FH
TKBDNRm0	00H to F0H
TKBDNRm1	00H to F0H
TKBMFRm	0000H to FFFFH



: The setting is fixed in this mode.



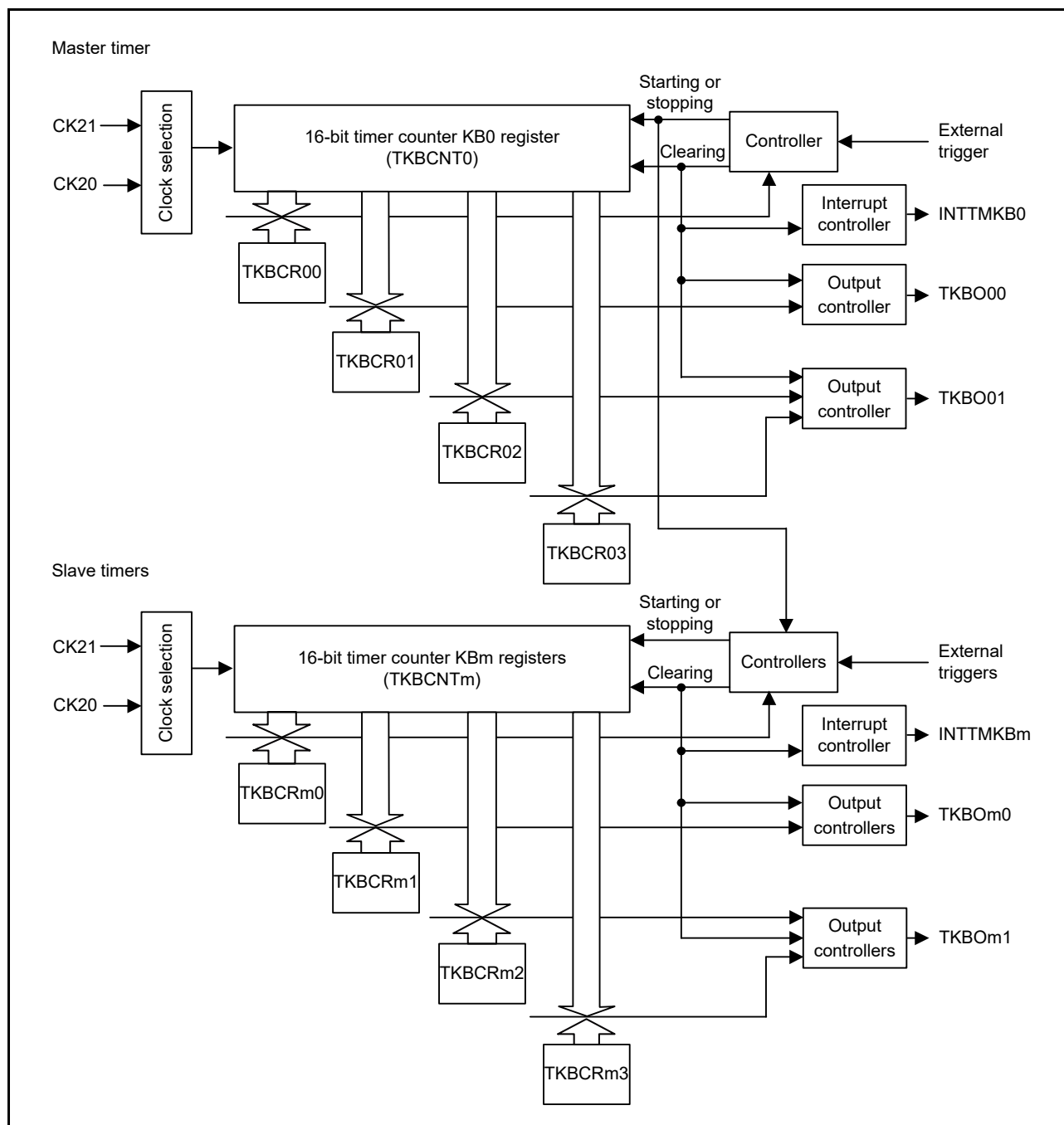
: Setting a value is not required (leave it as the default value).

Remark m = 1, 2

4. Configuration in simultaneous start/stop mode

Figure 9 - 51 shows the configuration in simultaneous start/stop mode.

Figure 9 - 51 Configuration in Simultaneous Start/Stop Mode

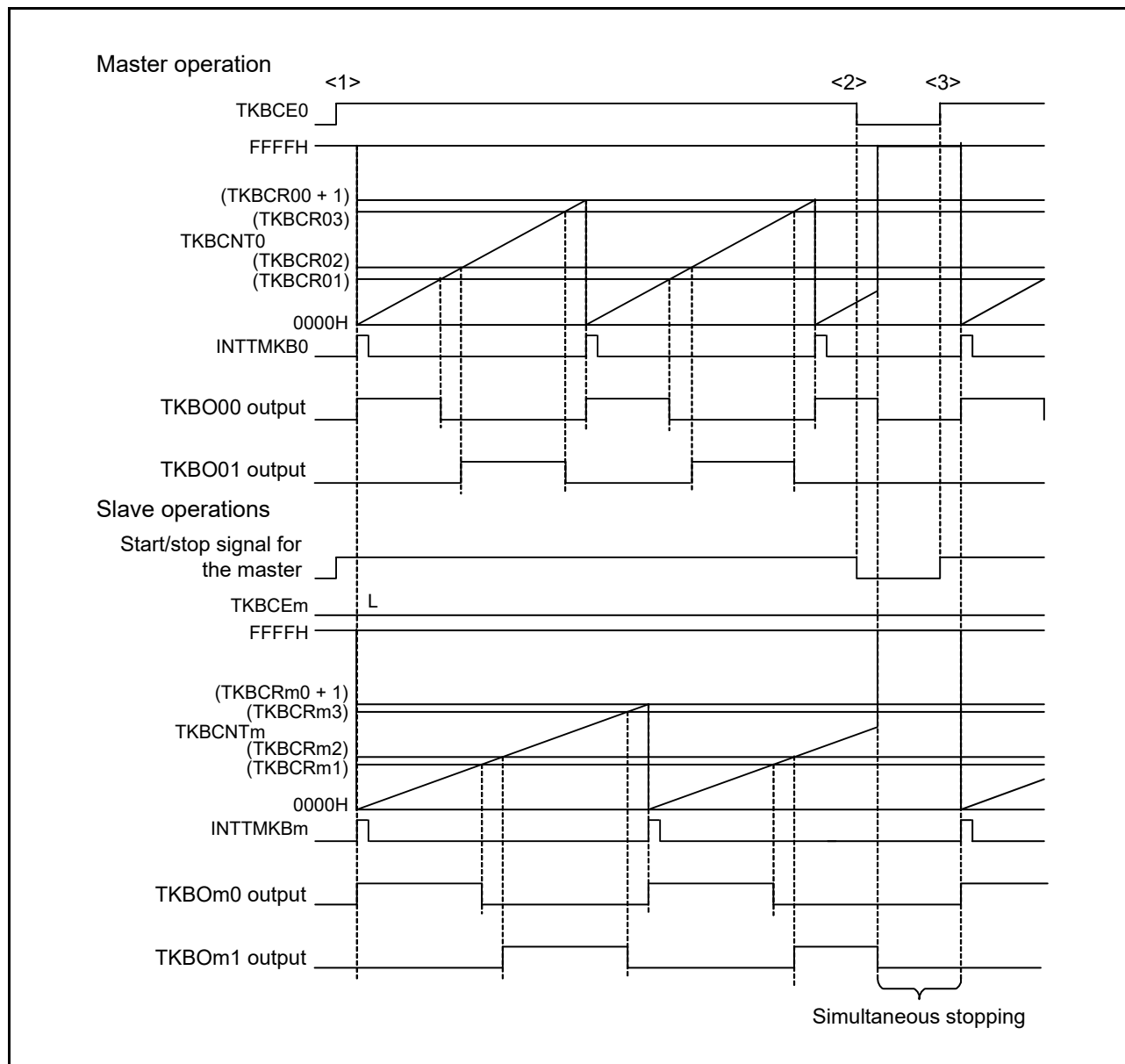


Remark m = 1, 2

5. Outline of operation

Figure 9 - 52 shows an example of the timing of operations in simultaneous start/stop mode.

Figure 9 - 52 Example of the Timing of Operations in Simultaneous Start/Stop Mode under Period Control by the TKBCR00 Register: The Default Level for Output Is Low (TKBTODnp = 0) and the Active Level Is High (TKBTOLnp = 0).



Remark m = 1, 2

The following describes an example of operations in simultaneous start/stop mode. <1> to <3> in **Figure 9 - 52** are described below.

- <1> Setting the TKBCE0 bit of the master to 1 resets the 16-bit timer counter KB0 (TKBCNT0) of the master and the 16-bit timer counters KBm (TKBCNTm) of the slaves to 0000H from FFFFH in synchronization with the counter clock and makes the counters start counting up. At the same time, INTTMKB0 and INTTMKBm interrupts are respectively output from the master and slaves and the TKBO00 and TKBOm0 outputs change from the default level to the active level (high in this example). For further details on the operation, see **Figure 9 - 45 Example of the Timing of Operations in Standalone Mode under Period Control by the TKBCRn0 Register**.
- <2> Setting the TKBCE0 bit to 0 stops counting up by the TKBCNT0 counter of the master and the TKBCNTm counters of the slaves and sets them to FFFFH in synchronization with the input clock for the given KB4n 16-bit timer. At the same time, the outputs of both the master and slaves change to their default levels. This state is retained until the TKBCE0 bit of the master is set to 1.
- <3> Setting the TKBCE0 bit of the master to 1 repeats the same sequence of operations starting from <1>.

9.4.9 Simultaneous start/clear mode

1. Outline of functions

When multiple KB4n 16-bit timers are to be used as the master and slaves, the counters for the slave KB4m 16-bit timers can be started and cleared at the same times as counting by the master KB40 16-bit timer starts or stops and as counter clearing and batch overwriting. In this case, operate the master in standalone mode (TKBMD0[1:0] = 00B) and the slaves in simultaneous start/clear mode (TKBMDm[1:0] = 10B).

Be sure to use the TKBCKS0 and TKBCKSm bits to select the same frequency-divided clock for the master and slaves.

For the formulas to calculate TKBO00 and TKBO01 outputs of the master, see **9.4.6 Standalone mode (with the period controlled by the TKBCRn0 register)** and **9.4.7 Standalone mode (with the period controlled by external trigger input)**.

Batch overwriting is controlled by writing 1 to the TKBRDT0 bit of the master.

Confirm the TKBRSF0 flag of the master before applying batch overwriting. The TKBCNTm counters of the slaves are cleared at the same time as clearing of the TKBCNT0 counter of the master.

Batch overwriting is applied to the compare registers of the slaves at the same time as batch overwriting for the master.

Since the slaves operate according to the cycle of the master (TKBCR00 setting + 1 or counter value on detection of external trigger input + 1), the TKBCRm0 registers of the slaves serve for setting the timing of the TKBOm0 outputs going to the active level.

INTTMKBm interrupts are generated at the same times as counting by the TKBCNT0 counter starts and as clearing of the TKBCNT0 counter. When skipping control is not to be used, the INTTMKBm interrupts have the same output timing as the INTTMKB0 interrupt.

The duty cycle of the slaves can be set within the range from 0% to 100% and calculated by using the following formulas.

[Formulas for TKBOm0 outputs of the slaves]

Pulse cycle = cycle of the master × counter clock period

Duty cycle [%] = ((TKBCRm1 setting – TKBCRm0 setting)/cycle of the master) × 100

Output with 0% duty cycle: TKBCRm1 setting = TKBCRm0 setting

Output with 100% duty cycle: TKBCRm0 setting = 0000H, TKBCRm1 setting ≥ cycle of the master

Caution Make sure that the TKBCRm0 setting is no greater than the TKBCRm1 setting.

[Formulas for TKBOm1 outputs of the slaves]

Pulse cycle = cycle of the master × counter clock period

Duty cycle [%] = ((TKBCRm3 setting – TKBCRm2 setting)/cycle of the master) × 100

Output with 0% duty cycle: TKBCRm3 setting = TKBCRm2 setting

Output with 100% duty cycle: TKBCRm2 setting = 0000H, TKBCRm3 setting ≥ cycle of the master

Caution Make sure that the TKBCRm2 setting is no greater than the TKBCRm3 setting.

Remark n = 0 to 2; m = 1, 2

2. Simultaneous start/clear mode

Master: Sample register settings in standalone mode

TKBCTL00	15	14	13	12	11	10	9	8
	TKBTGM0 1 0	— 0	TKBSSE01 1/0	TKBDIE01 1/0	TKBTGM0 0 0	— 0	TKBSSE00 1/0	TKBDIE00 1/0
TKBCTL01	7	6	5	4	3	2	1	0
	TKBMFE0 1/0	TKBIRS02 1/0	TKBIRS01 0	TKBIRS00 0	TKBIHE0 0	TKBTSE0 1/0	TKBSTS01 1/0	TKBSTS00 1/0
TKBIOC00	7	6	5	4	3	2	1	0
	TKBCE0 1/0	— 0	— 0	TKBCKS0 1/0	TKBSCM0 0	— 0	TKBMD01 0	TKBMD00 0
TKBIOC01	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOE01 1/0	TKBTOE00 1/0
TKBCTL02	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINS02 1/0	TKBINS01 1/0	TKBINS00 1/0

TKBCR00	0000H to FFFFH
TKBCR01	0000H to FFFFH
TKBCR02	0000H to FFFFH
TKBCR03	0000H to FFFFH
TKBTGCR0	0000H to FFFFH
TKBSIR00	0000H to FFFFH
TKBSIR01	0000H to FFFFH
TKBSSR00	00H to 0FH
TKBSSR01	00H to 0FH
TKBDNR00	00H to F0H
TKBDNR01	00H to F0H
TKBMFR0	0000H to FFFFH



: The setting is fixed in this mode.



: Setting a value is not required (leave it as the default value).

3. Simultaneous start/clear mode

Slaves: Sample register settings in simultaneous start/clear mode

TKBCTLm0	15	14	13	12	11	10	9	8
	TKBTGMm 1 0	— 0	TKBSSEm 1 1/0	TKBDIEm1 1/0	TKBTGMm 0 0	— 0	TKBSSEm 0 1/0	TKBDIEm0 1/0
	7	6	5	4	3	2	1	0
	TKBMFEm 0	TKBIRSm2 1/0	TKBIRSm1 0	TKBIRSm0 0	TKBIHEm 0	TKBTSEm 0	TKBSTSm 1 0	TKBSTSm 0 0
TKBCTLm1	7	6	5	4	3	2	1	0
	TKBCEm 0	— 0	— 0	TKBCKSm 1/0	TKBSCMm 0	— 0	TKBMDm1 1	TKBMDm0 0
TKBIOCm0	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOLm1 1/0	TKBTOLm0 1/0	TKBTODm 1 1/0	TKBTODm 0 1/0
TKBIOCm1	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEm 1 1/0	TKBTOEm 0 1/0
TKBCTLm2	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINSm2 0	TKBINSm1 0	TKBINSm0 0

TKBCRm0	0000H to FFFFH
TKBCRm1	0000H to FFFFH
TKBCRm2	0000H to FFFFH
TKBCRm3	0000H to FFFFH
TKBTGCRm	0000H to FFFFH
TKBSIRm0	0000H to FFFFH
TKBSIRm1	0000H to FFFFH
TKBSSRm0	00H to 0FH
TKBSSRm1	00H to 0FH
TKBDNRm0	00H to F0H
TKBDNRm1	00H to F0H
TKBMFRm	0000H



: The setting is fixed in this mode.



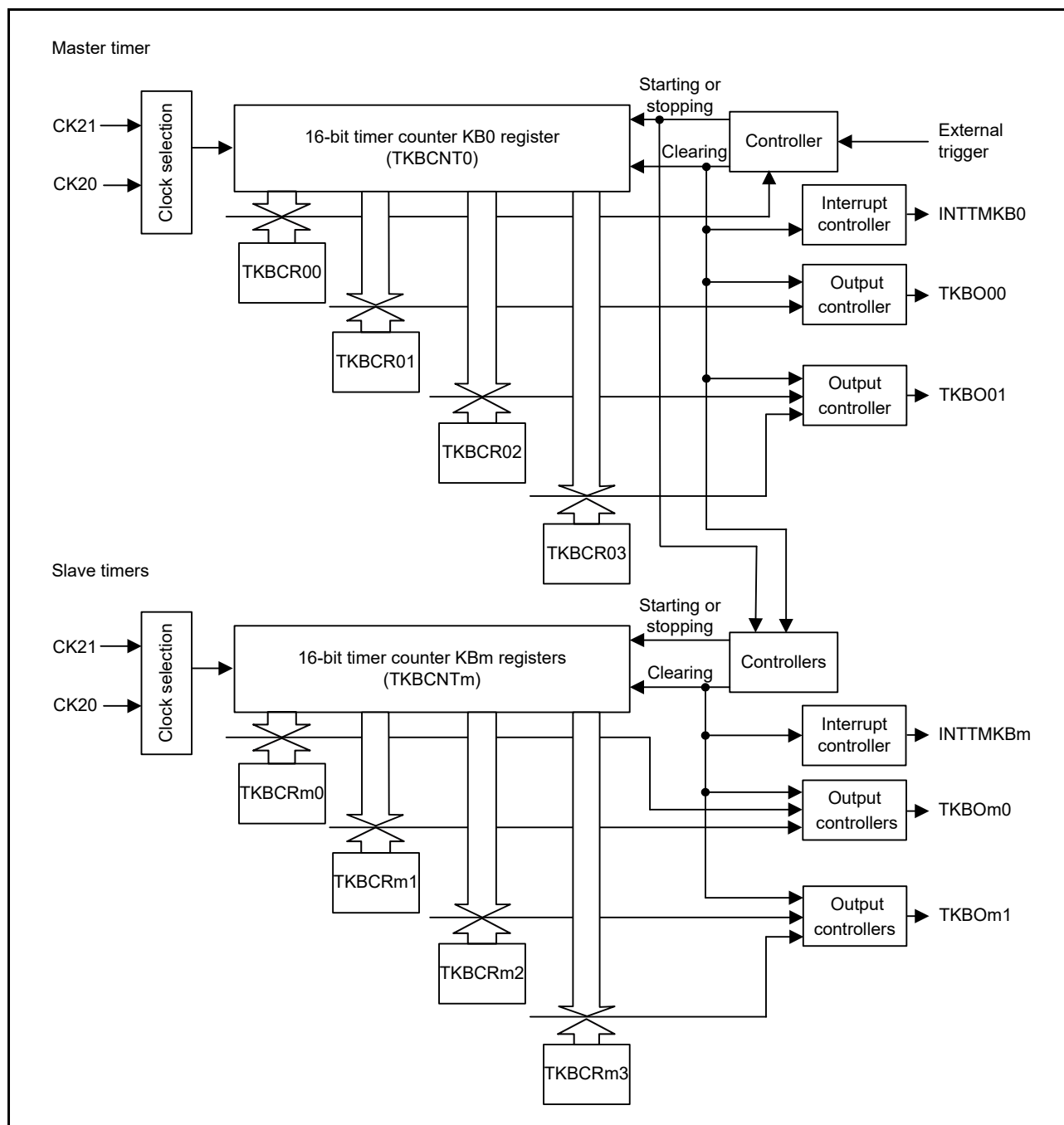
: Setting a value is not required (leave it as the default value).

Remark m = 1, 2

4. Configuration in simultaneous start/clear mode (with the period controlled by the master)

Figure 9 - 53 shows the configuration in simultaneous start/clear mode.

Figure 9 - 53 Configuration in Simultaneous Start/Clear Mode (with the Period Controlled by the Master)

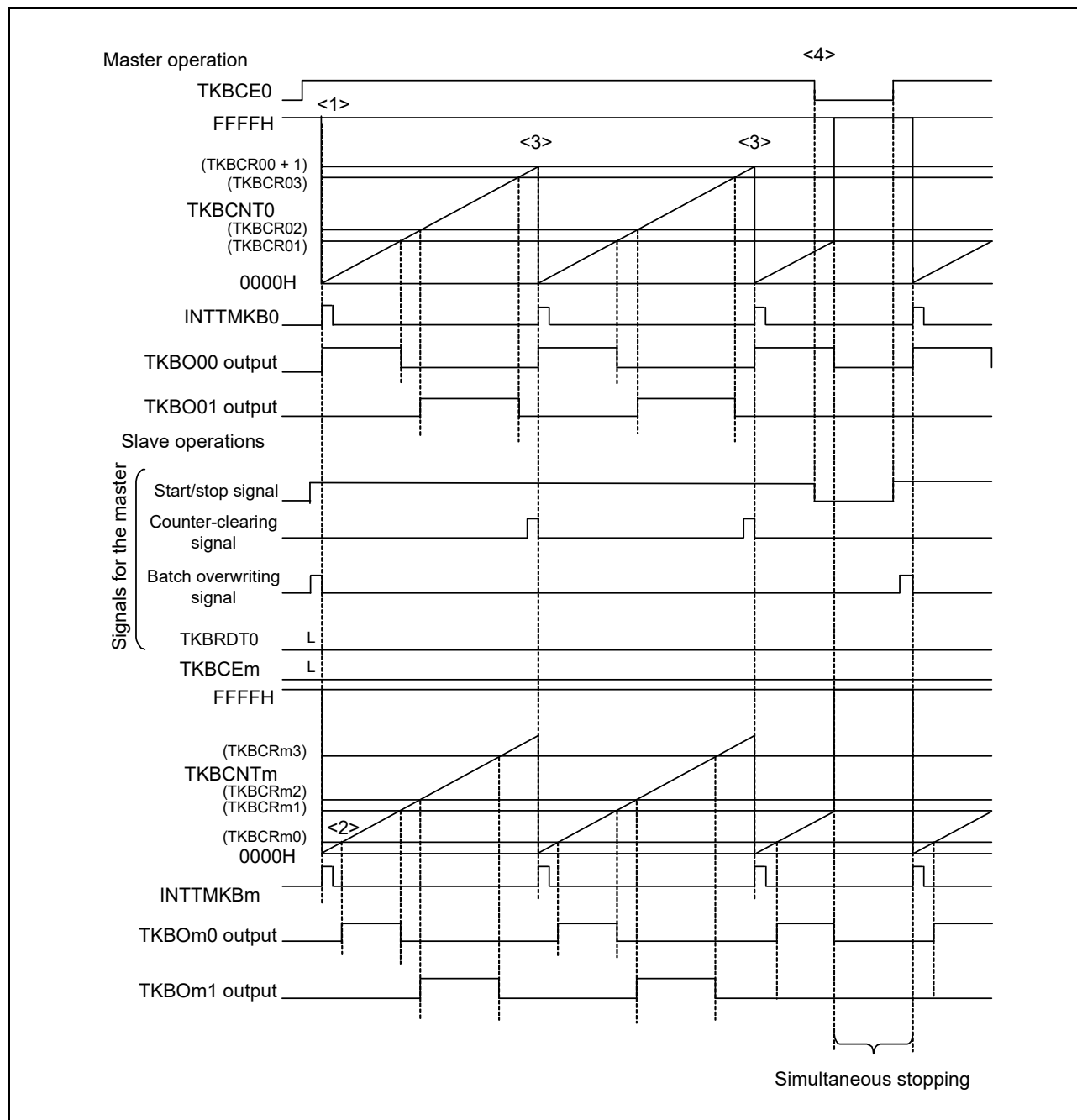


Remark m = 1, 2

5. Outline of operation

Figure 9 - 54 shows an example of the timing of operations in simultaneous start/clear mode.

Figure 9 - 54 Example of the Timing of Operations in Simultaneous Start/Clear Mode under Period Control by the Master: The Default Level for Output Is Low (TKBTODnp = 0) and the Active Level Is High (TKBTOLnp = 0).

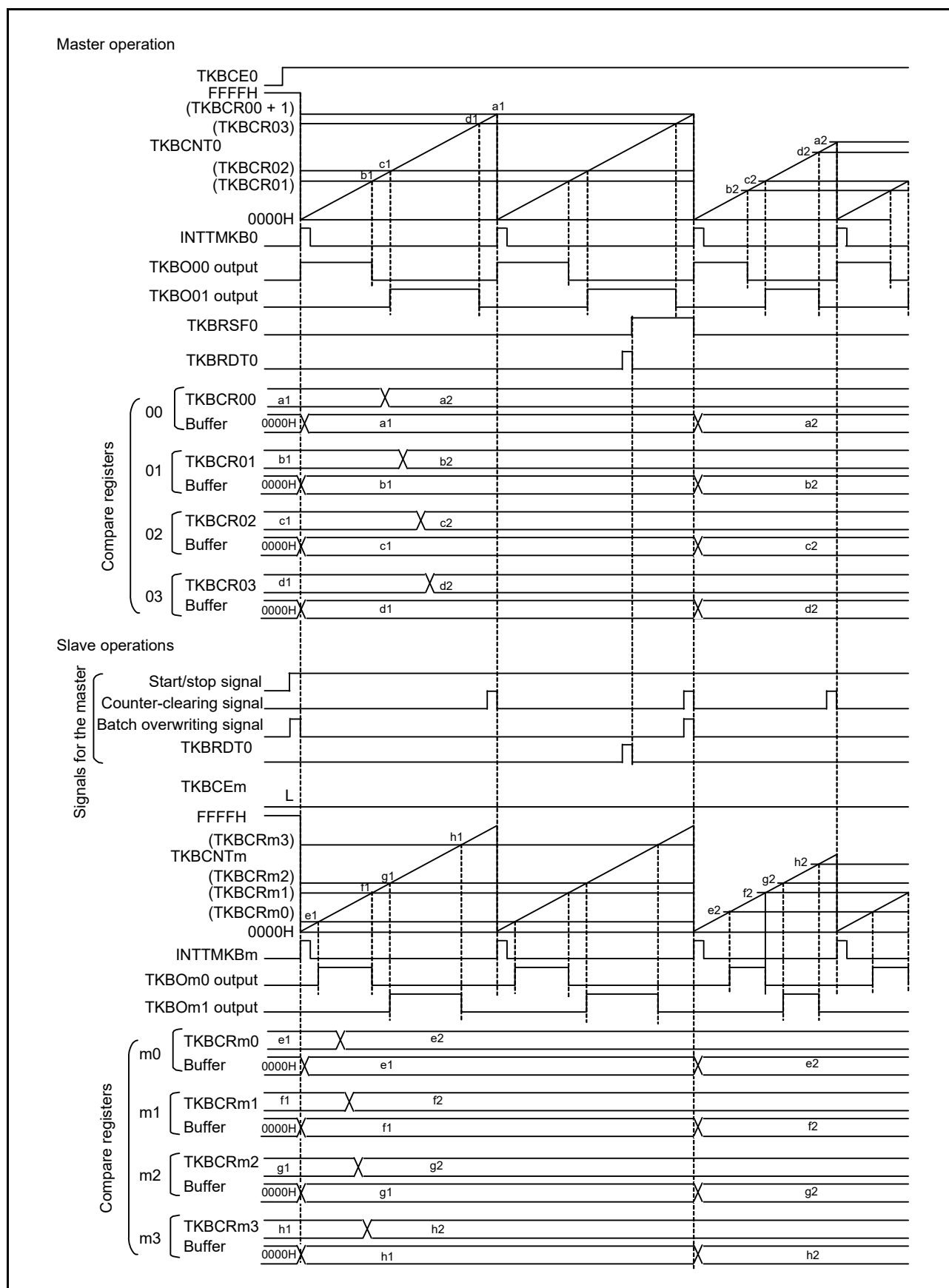


The following describes an example of operations in simultaneous start/clear mode. <1> to <4> in **Figure 9 - 54** are described below.

- <1> Setting the TKBCE0 bit of the master to 1 resets the 16-bit timer counter KB0 (TKBCNT0) of the master and the 16-bit timer counters KBm (TKBCNTm) of the slaves to 0000H from FFFFH in synchronization with the counter clock and makes the counters start counting up. At the same time, an INTTMKB0 interrupt is output from the master and the TKBO00 output changes from the default level to the active level (high in this example).
- <2> When the value of a TKBCNTm counter matches the value specified in the corresponding TKBCRm0 register, the TKB0m0 output of the slave goes to the active level. For further details on the operation, see **Figure 9 - 45 Example of the Timing of Operations in Standalone Mode under Period Control by the TKBCRn0 Register**.
- <3> When the value of the TKBCNT0 counter matches the value specified in the TKBCR00 register, the counter-clearing signal for the master is output and the 16-bit timer counters of the master and slaves (TKBCNT0 and TKBCNTm) are cleared at the same time.
- <4> Setting the TKBCE0 bit to 0 stops counting up by the TKBCNT0 counter of the master and the TKBCNTm counters of the slaves and sets them to FFFFH in synchronization with the input clock for the given KB4n 16-bit timer. At the same time, the outputs of both the master and slaves change to their default levels. This state is retained until the TKBCE0 bit of the master is set to 1.

Figure 9 - 55 shows an example of the timing of batch overwriting operation in simultaneous start/clear mode. In this case, setting the TKBRDT0 bit of the master to 1 also applies batch overwriting to the slaves at the time of the next clearing.

Figure 9 - 55 Example of Timing of Batch Overwriting Operation under Period Control by the Master in Simultaneous Start/Clear Mode



9.4.10 Interleaved power factor correction (PFC) output mode

1. Outline of functions

The interleaved PFC output mode allows interleaved output of the signal for controlling a PFC circuit which regulates the harmonic currents of a power supply. As an interleaved PFC circuit can regulate peak input current to a greater extent than a single PFC circuit, the interleaved approach enables reduction of the sizes of parts and usage with higher-powered power units. This mode supports a PFC circuit under control in the critical conduction mode.

Interleaved PFC control requires two inputs for zero current detection and two PWM outputs for use in switching. Interleaved PFC control is implemented for the 16-bit timers KB40, KB41, and KB42 by combining the INTP0 external interrupt input with TKBOn0 timer output and the INTP1 external interrupt input with TKBOn1 timer output. The TKBOn0 and TKBOn1 pins output pulses in response to input of the INTP0 and INTP1 signals, respectively. In this case, the TKBOn1 output is controlled so that its phase is shifted by 180 degrees relative to the output timing of TKBOn0.

Remark Single PFC control can be implemented in the standalone mode (with the period controlled by external trigger input). For more details, see **9.4.7 Standalone mode (with the period controlled by external trigger input)**. PFC in the continuous current mode can be implemented in the standalone mode (with the period controlled by the TKBCRn0 register). For more details, see **9.4.6 Standalone mode (with the period controlled by the TKBCRn0 register)**.

Use the TKBCRn0 register to set the pulse cycle of the 16-bit timer KB4n when no external interrupt input on INTP0 is detected. Use the TKBCRn1 and TKBCRn3 registers to respectively set the width at the active level per cycle for TKBOn0 and TKBOn1 outputs.

Remark In interleaved power factor correction (PFC) output mode, the TKBCRn2 register is not used.

Set the same value in each of the following pairs of bits: TKBTOLn0 and TKBTODn0, TKBTOLn1 and TKBTODn1 so that the active level becomes high when the default level is low and the active level becomes low when the default level is high.

[Formulas for TKBOn0 and TKBOn1 outputs]

Pulse cycle = (counter value on detection of INTP0 external interrupt input + 1) × counter clock period

Pulse cycle (max.)^{Note 1} = (TKBCRn0 setting + 1) × counter clock period

Width at the active level per cycle for TKBOn0 output = TKBCRn1 setting × counter clock period

Width at the active level per cycle for TKBOn1 output = TKBCRn3 setting × counter clock period

Width of the cycle-to-cycle phase difference during TKBOn1 output^{Note 2} = INT((width of the previous cycle – 1)^{Note 3}/2 + 1) × counter clock period

Note 1. The pulse cycle of the 16-bit timer KB4n when no external interrupt input on INTP0 is detected

Note 2. This does not apply to operation under condition 7.

Note 3. The setting of the TKBCRn0 register is used during operation under condition 1.

2. Outline of operation

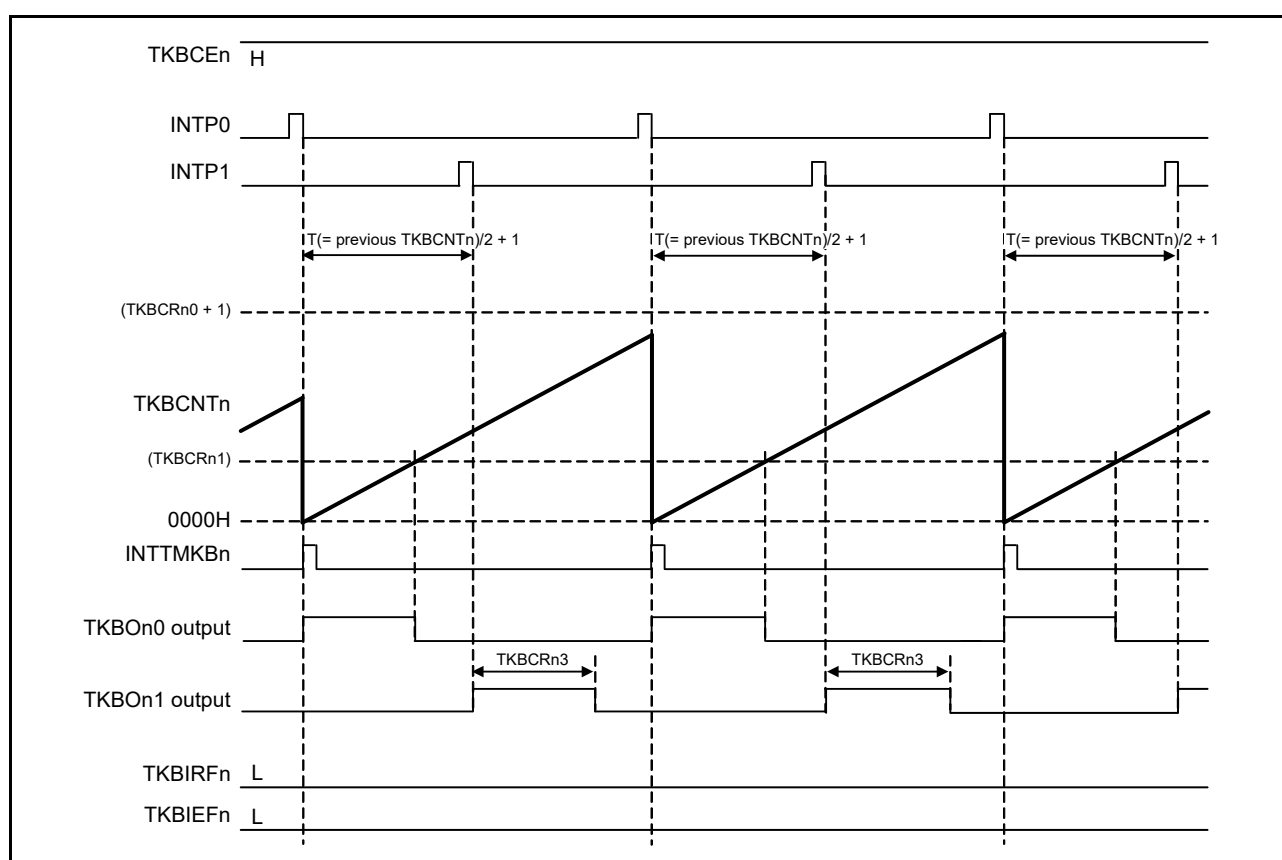
Figure 9 - 56 gives an overview of operations in interleaved PFC output mode.

In interleaved PFC output mode, the TKBCNTn counter is cleared and starts counting up from 0000H by using INTP0 as a trigger. The TKBOn0 output is placed at the active level at this time and is returned to the inactive level when the values of the TKBCNTn counter and TKBCRn1 register match.

The TKBOn1 output is placed at the active level by using INTP1, which is in a different phase to that of INTP0, as a trigger and is returned to the inactive level after the width at the active level has reached the value of the TKBCRn3 register.

A further pulse of INTP0 is input before a match between the TKBCNTn counter and TKBCRn0 register and the above operations are repeated.

Figure 9 - 56 Outline of Operations in Interleaved PFC Output Mode: The Default Level for Output Is Low (TKBTODnp = 0) and the Active Level Is High (TKBTOLnp = 0).



3. Conditions for TKBOn1 output in the interleaved PFC output mode

Control of the TKBOn1 output is determined in accord with the conditions stated in the table below.

(1/2)

Condition No.	Sub No.	Judgment 1 (INTP0 Input)	Judgment 2 (Matching with TKBCRn0/INTP1 Input)	Judgment 3 (Cycle Width)	Output	Error
1	—	First cycle (generate a waveform setting T for TKBCRn0)	—	—	Output at T/2	—
2	—	No detection of INTP0 input	Matching of TKBCNTn and TKBCRn0 (INTP1 input detection is ignored.)	The subsequent cycle length is no less than T/2.	Output at T/2	—
3	—	↑	↑	The subsequent cycle length is less than T/2.	Output is retained.	—
4	—	Cycle subsequent to condition 3	—	—	Output at T/2	—
5	—	The first detection of the INTP0 input ^{Note 1}	—	—	Output at T/2	—
6	a	The second or a subsequent detection of the INTP0 input ^{Note 2}	When TKBIRSn[2:0] = 0xxB: Detection of INTP1 (within the range from the previous falling edge of TKBOn1 to T/2)	The subsequent cycle length is no less than T/2.	Output at T/2	—
	b			The subsequent cycle length is less than T/2.	Output is retained.	—
	c		When TKBIRSn[2:0] = 100B: Detection of INTP1 (within the range from the previous falling edge of TKBOn1 to T/2)	The subsequent cycle length is no less than T/2.	Output at T/2	—
	d			The subsequent cycle length is less than T/2.	Output is retained.	The TKBIRFn bit is set on INTP0 trigger input.
7	a	The second or a subsequent detection of the INTP0 input ^{Note 2}	When TKBIRSn[2:0] = 0xxB: Detection of INTP1 (within the range from T/2 to T/2 + T/(TKBIRSn[1:0] setting))	—	Output on trigger input	—
	b		When TKBIRSn[2:0] = 100B or 101B: Detection of INTP1 (within the range from T/2 to TKBCRn0 setting)			—
	c		When TKBIRSn[2:0] = 101B: Detection of INTP1 (within the range from the previous falling edge of TKBOn1 to T/2)			—

(2/2)

Condition No.	Sub No.	Judgment 1 (INTP0 Input)	Judgment 2 (Matching with TKBCRn0/INTP1 Input)	Judgment 3 (Cycle Width)	Output	Error
8	a	The second or a subsequent detection of the INTP0 input ^{Note 2}	When $TKBIRSn[2:0] = 0xB$: Detection of INTP1 (after the range of $T/2 + T/(TKBIRSn[1:0] \text{ setting})$) or no detection of INTP1	The subsequent cycle length is no less than the range of $T/2 + T/(TKBIRSn[1:0] \text{ setting})$	Output is retained.	The $TKBIRFn$ bit is set at $T/2 + T/(TKBIRSn[1:0] \text{ setting})$.
	b		When $TKBIRSn[2:0] = 0xB$: No detection of INTP1	The subsequent cycle length is no less than $T/2$ and also less than the range of $T/2 + T/(TKBIRSn[1:0] \text{ setting})$		—
	c		When $TKBIRSn[2:0] = 100B$ or $101B$: No detection of INTP1	—		The $TKBIRFn$ bit is set on INTP0 trigger input.
9	a	Cycle subsequent to condition 6d, 8, 9b, or 9c	—	The subsequent cycle length is no less than $T/2$.	Output at $T/2$	—
	b		When $TKBIRSn[2:0] = 0xB$: —	The subsequent cycle length is less than $T/2$.	Output is retained.	—
	c		When $TKBIRSn[2:0] = 100B$ or $101B$: —			The $TKBIRFn$ bit is set on INTP0 trigger input.
10	—	Detection of INTP0 input	When $TKBIRSn[2:0] = 0xB$: No detection of INTP1	The subsequent cycle length is less than $T/2$.	Output is retained.	—
11	a	Cycle subsequent to condition 10 or 11b	—	The subsequent cycle length is no less than $T/2$.	Output at $T/2$	—
	b			The subsequent cycle length is less than $T/2$.	Output is retained.	—

Note 1. “The first detection of the INTP0 input” refers to cases where clearing in response to detection of the INTP0 input had not proceeded in the previous cycle.

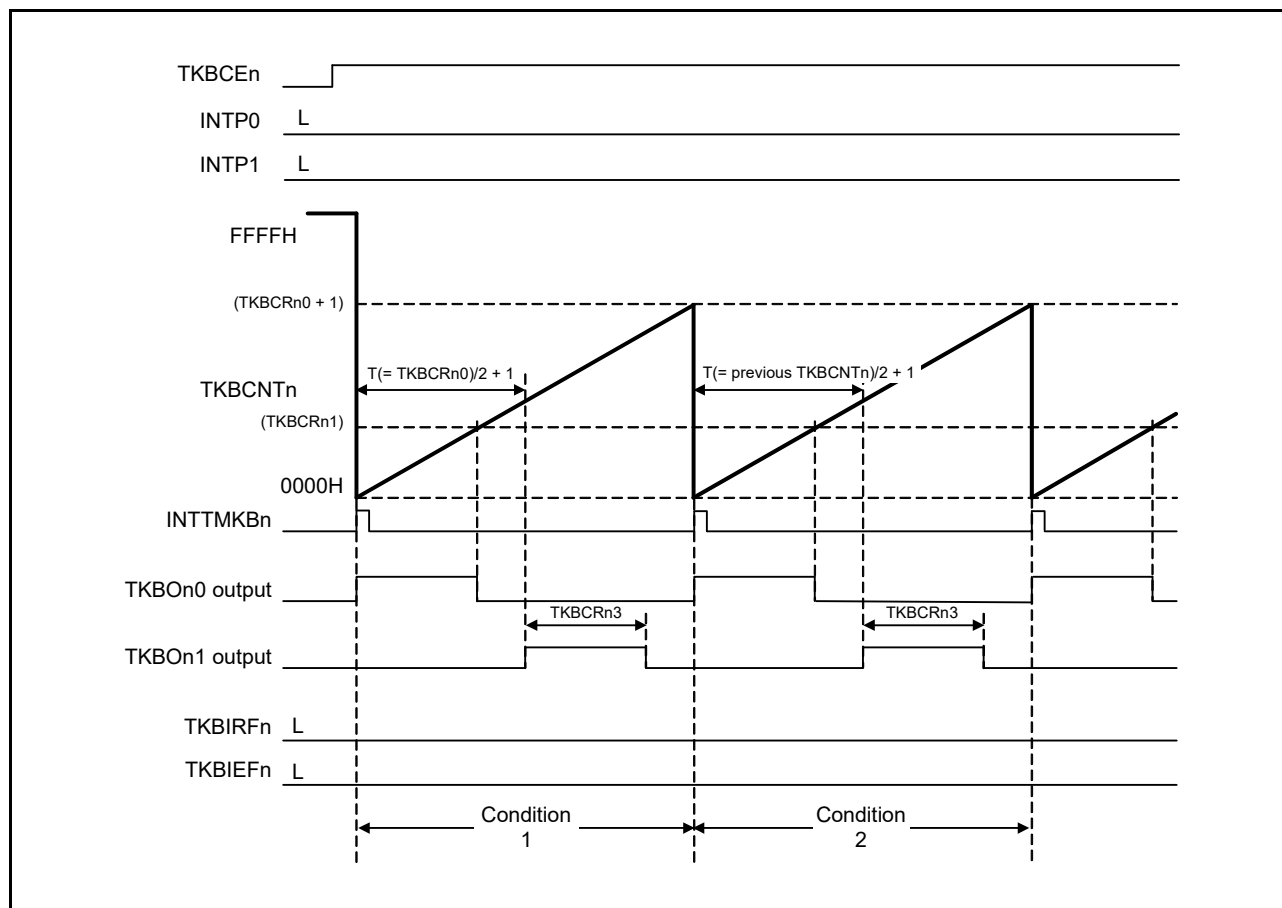
Note 2. “The second or a subsequent detection of the INTP0 input” refers to cases where clearing in response to detection of the INTP0 input had proceeded in the previous cycle.

4. Operations when the $\text{TKBIRSn}[2:0]$ bits are set to 0xxB

Waveforms during operations under various conditions when the $\text{TKBIRSn}[2:0]$ bits are set to 0xxB are given in the following figures.

These waveforms of operations are for cases where the default and active levels for output are low ($\text{TKBTODnp} = 0$) and high ($\text{TKBTOLnp} = 0$), respectively.

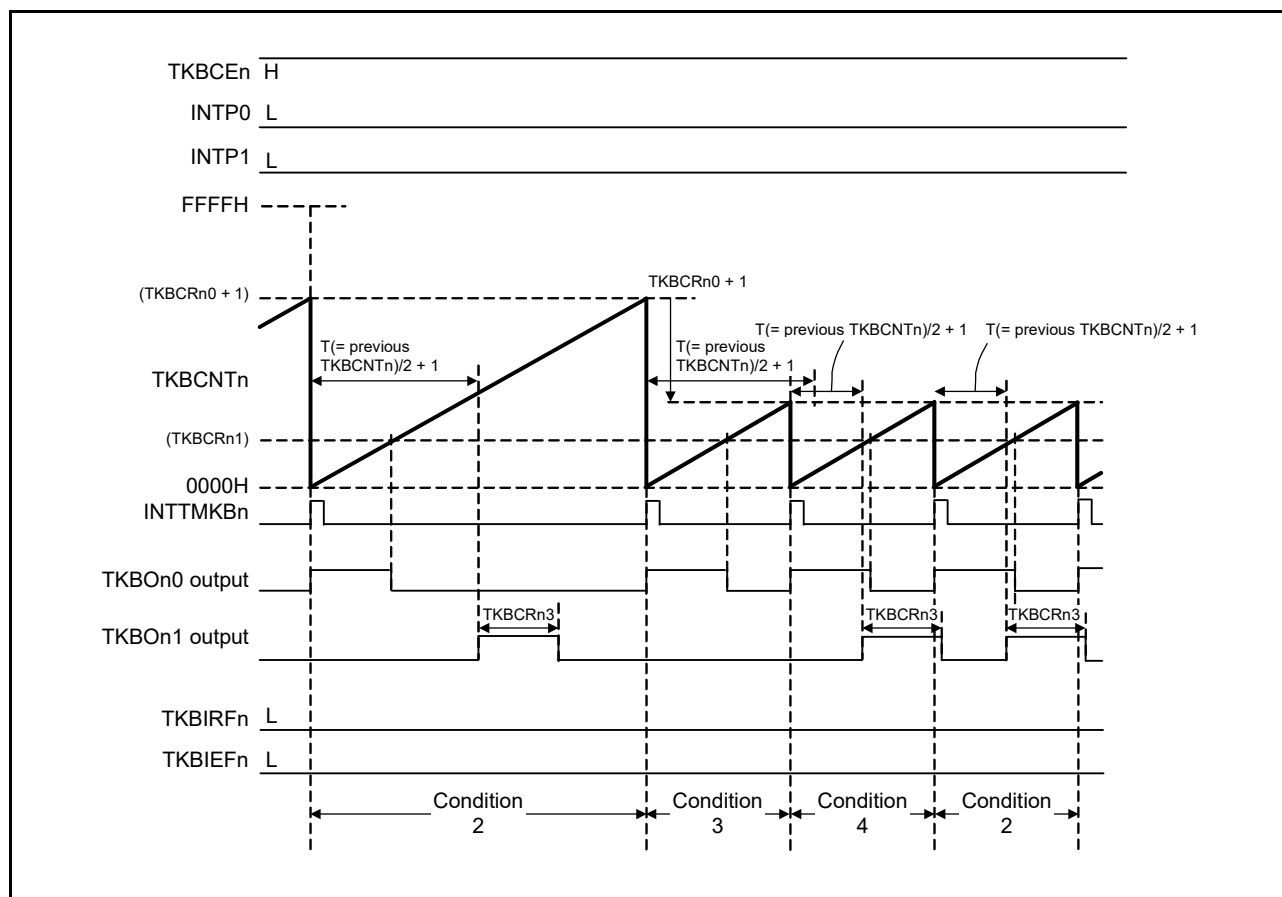
Figure 9 - 57 Timing of Interleaved PFC Output Mode with a Transition from Condition 1 to 2



Condition 1: Only in the first cycle after TKBCEn has been set to 1, if the setting of the TKBCRn0 register is T , a TKBO1 pulse with the width set in the TKBCRn3 register is output at $T/2$.

Condition 2: In the second cycle, a TKBO1 pulse with the width set in the TKBCRn3 register is output at $T/2$ of the previous cycle.

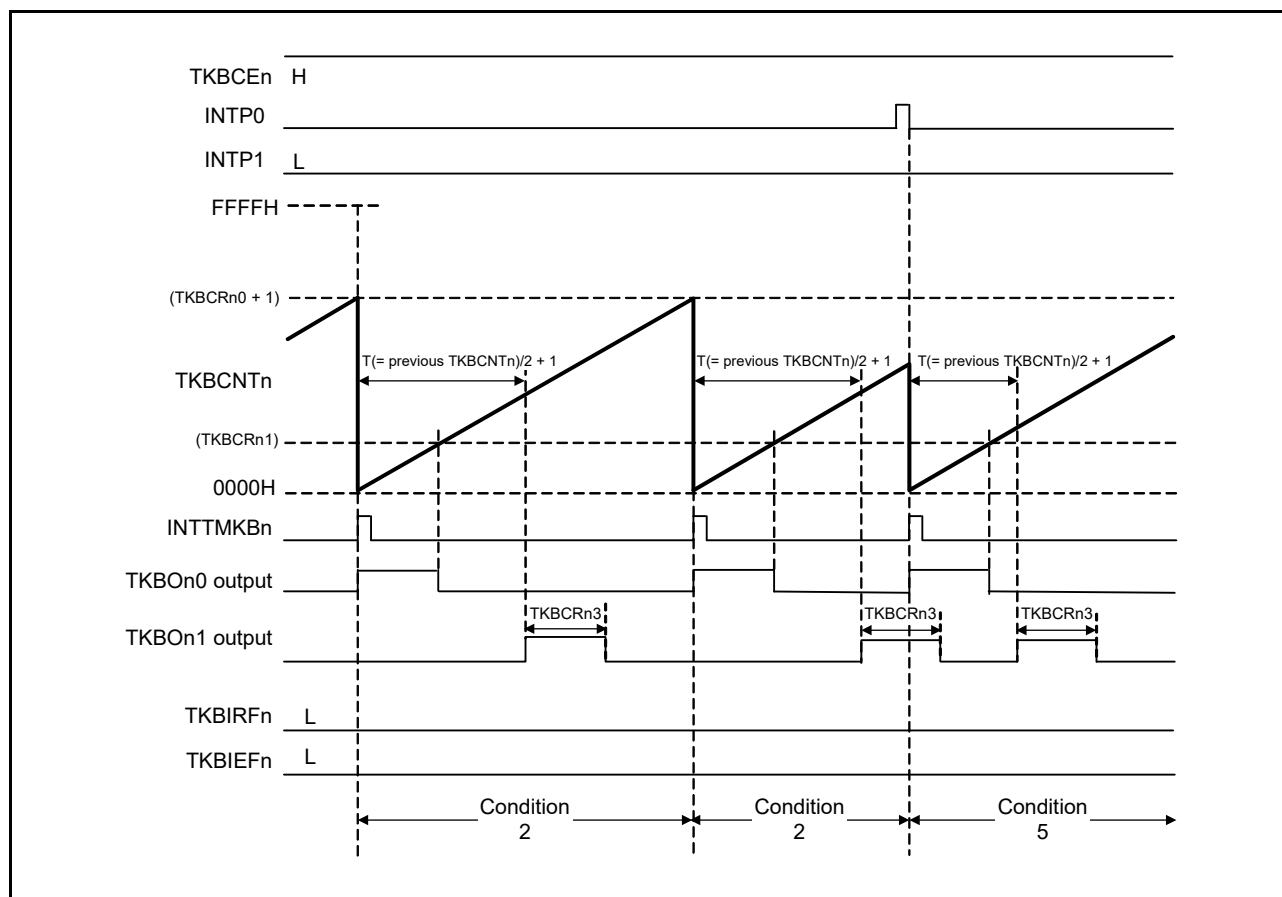
Figure 9 - 58 Timing of Interleaved PFC Output Mode with a Transition from Condition 3 to 4



Condition 3: Securing T/2 of the previous cycle is not possible and the TKBOn1 output is retained.

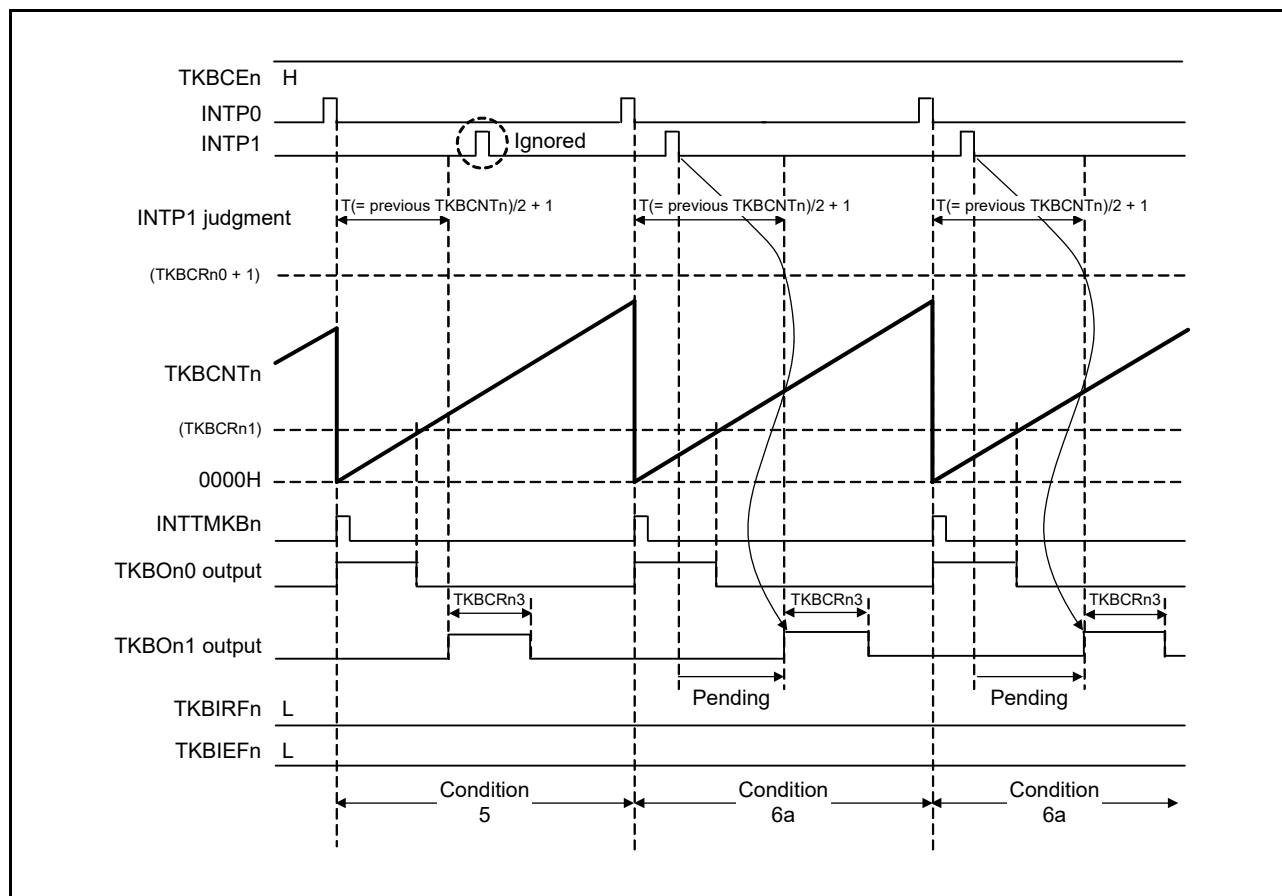
Condition 4: A TKBOn1 pulse with the width set in the TKBCRn3 register is output at T/2 of the previous cycle.

Figure 9 - 59 Timing of Interleaved PFC Output Mode under Condition 5



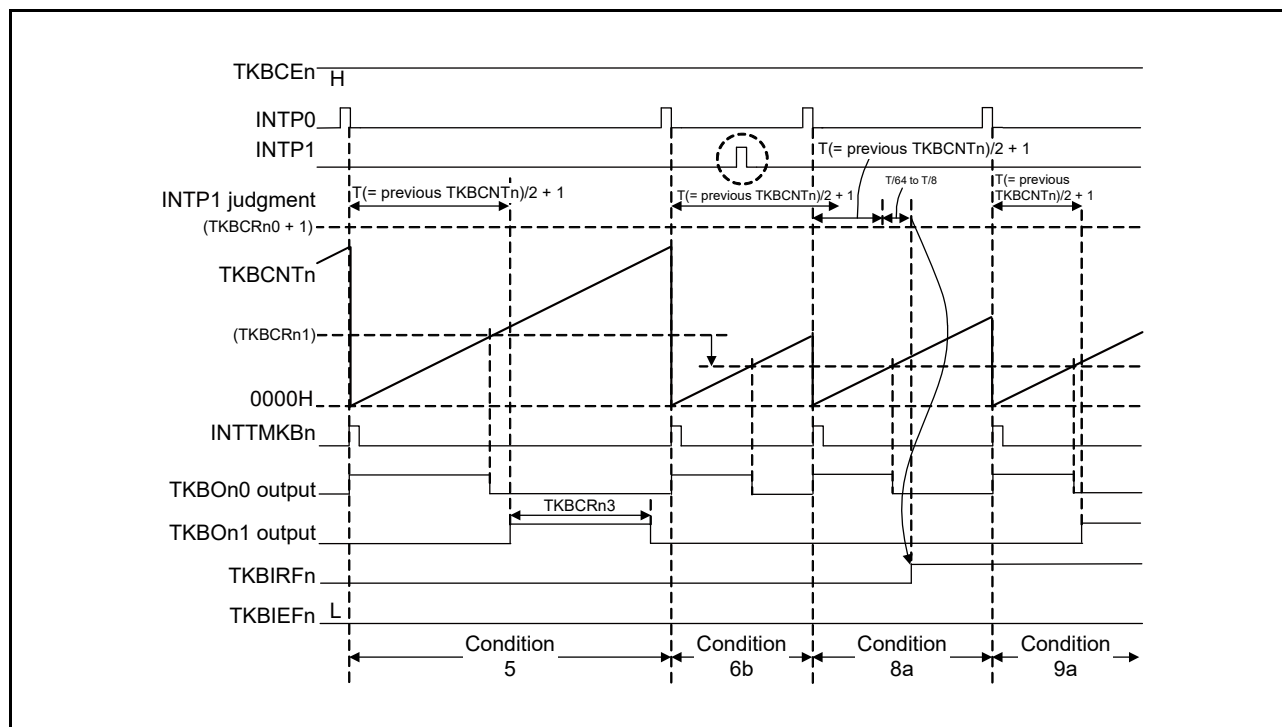
Condition 5: Detecting INTP0 for the first time after the TKBCEn bit has been set to 1 causes the output of a TKBOn1 pulse with the width set in the TKBCRn3 register at $T/2$ of the previous cycle. In this case, detection or non-detection of INTP1 is irrelevant.

Figure 9 - 60 Timing of Interleaved PFC Output Mode under Condition 6a: INTP1 Is Detected within T/2 of the Previous Cycle When the Cycle Length Is No Less Than T/2 of That for the Previous Cycle.



Condition 6a: When INTP1 is detected within T/2 of the previous cycle, a TKBOn1 pulse with the width set in the TKBCRn3 register is output at T/2 of the previous cycle.

Figure 9 - 61 Timing of Interleaved PFC Output Mode under Condition 6b: INTP1 Is Detected When the Cycle Length Is Less Than $T/2$ of That for the Previous Cycle.

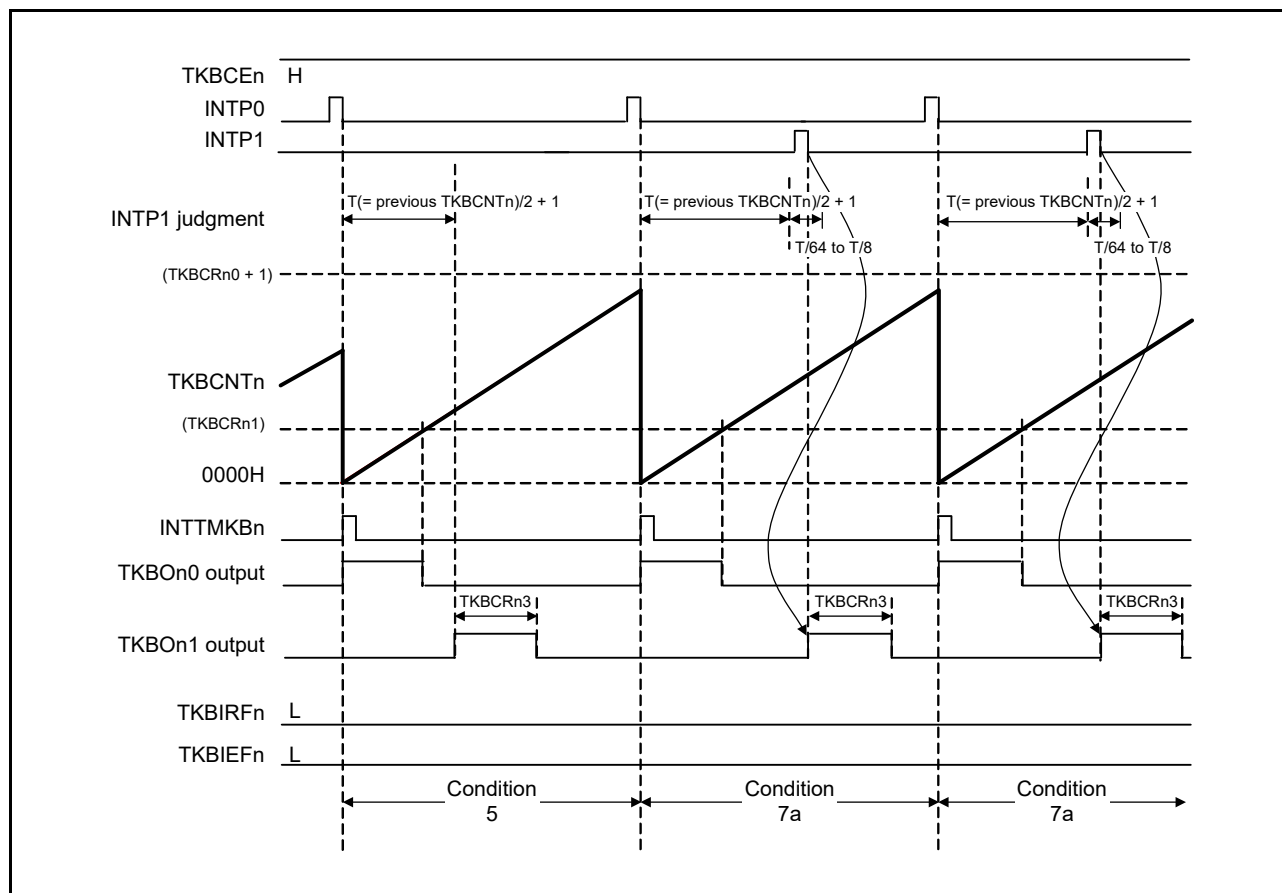


Condition 6b: INTP1 is detected but securing $T/2$ of the previous cycle is not possible. The TKBOn1 output is retained.

Condition 8a: INTP1 is not detected within $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $TKBIRSn[1:0]$ bits) of the previous cycle, and the TKBOn1 output is retained. The TKBIRFn flag is set to 1 at this time.

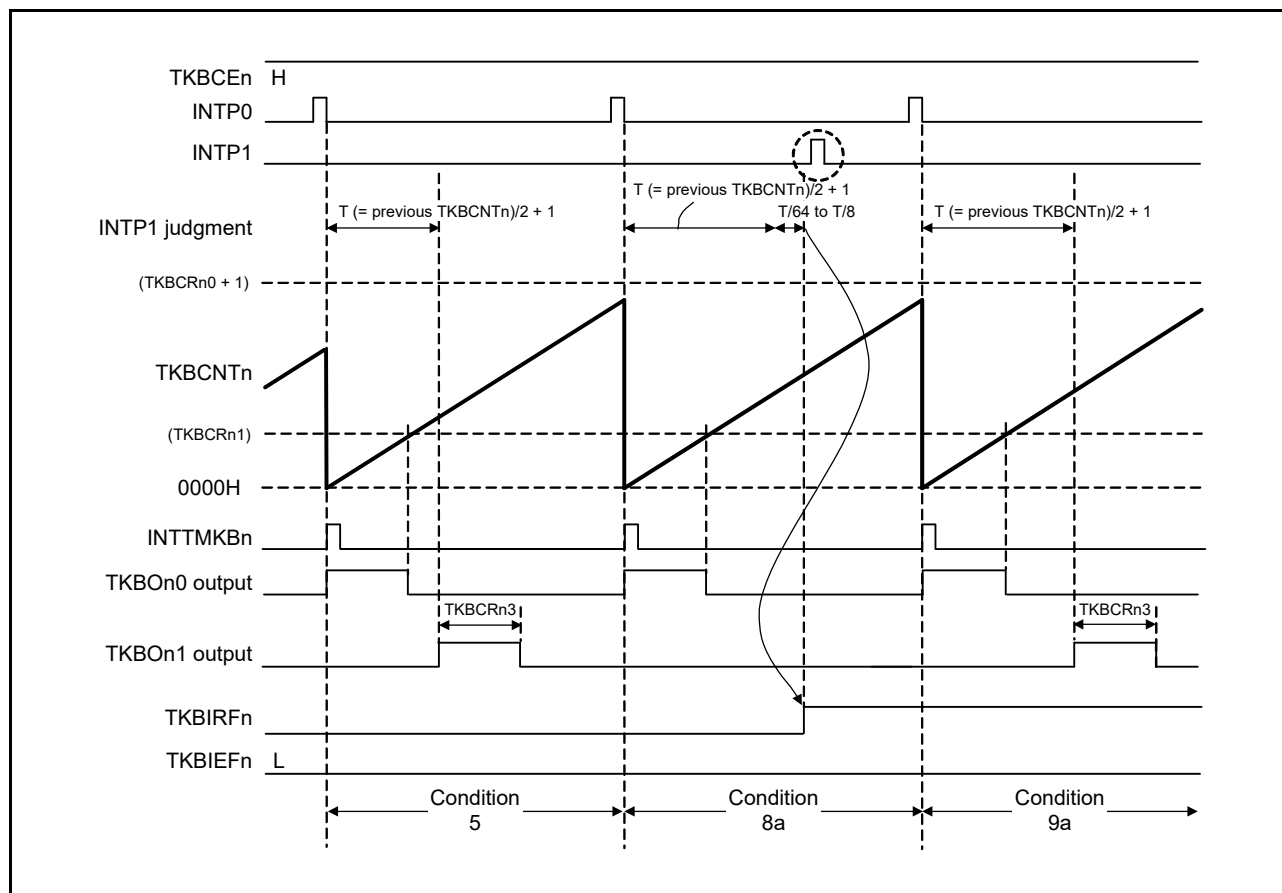
Condition 9a: A TKBOn1 pulse with the width set in the TKBCRn3 register is output at $T/2$ of the previous cycle.

Figure 9 - 62 Timing of Interleaved PFC Output Mode under Condition 7a: INTP1 Is Detected within the Range from $T/2$ to $T/2 + T/m$ (m : 8, 16, 32, or 64; Set by the $TKBIRSn[1:0]$ Bits) of the Previous Cycle.



Condition 7a: INTP1 is detected within the range from $T/2$ to $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $TKBIRSn[1:0]$ bits) of the previous cycle, and a $TKBOn1$ pulse with the width set in the $TKBcrn3$ register is output.

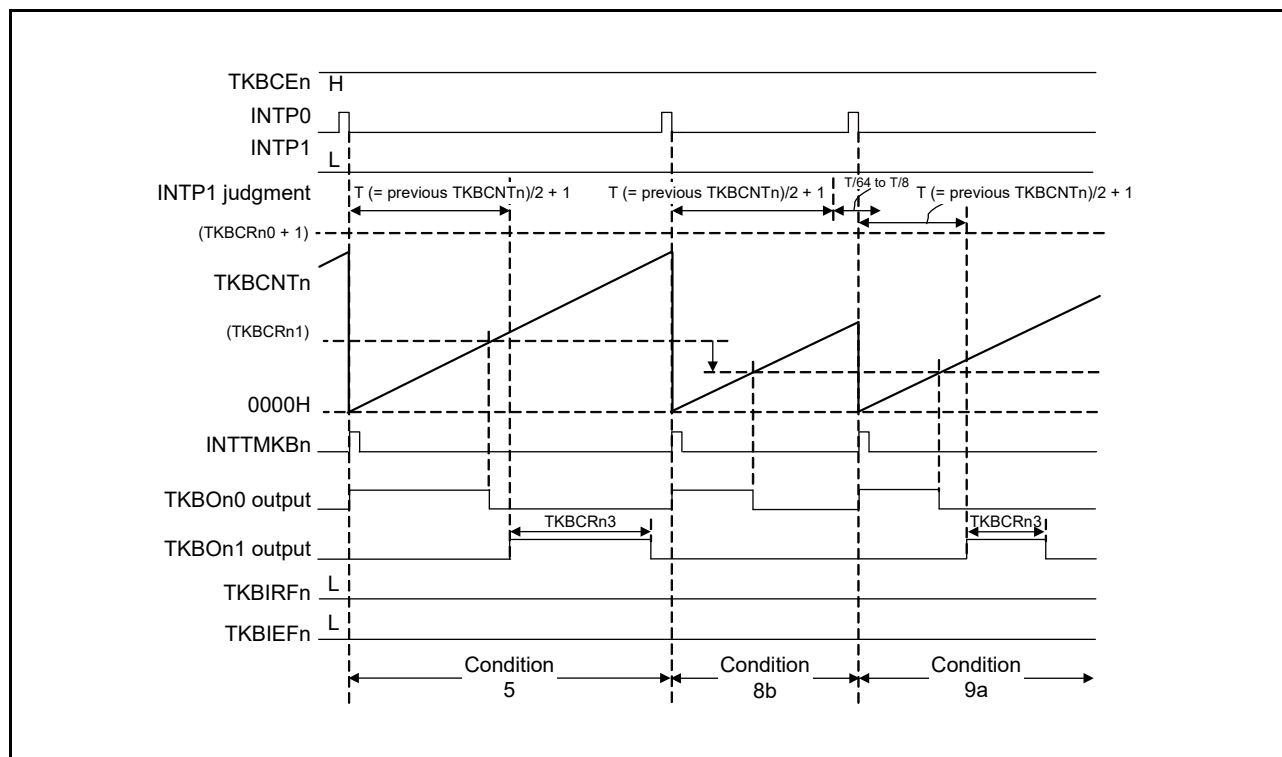
Figure 9 - 63 Timing of Interleaved PFC Output Mode with a Transition from Condition 8a to 9a: INTP1 Is Detected after the Range of $T/2 + T/m$ (m : 8, 16, 32, or 64; Set by the $TKBIRSn[1:0]$ Bits) of the Previous Cycle.



Condition 8a: INTP1 is not detected within $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $TKBIRSn[1:0]$ bits) of the previous cycle, and the $TKBOn1$ output is retained. The $TKBIRFn$ flag is set to 1 at this time.

Condition 9a: A $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output at $T/2$ of the previous cycle.

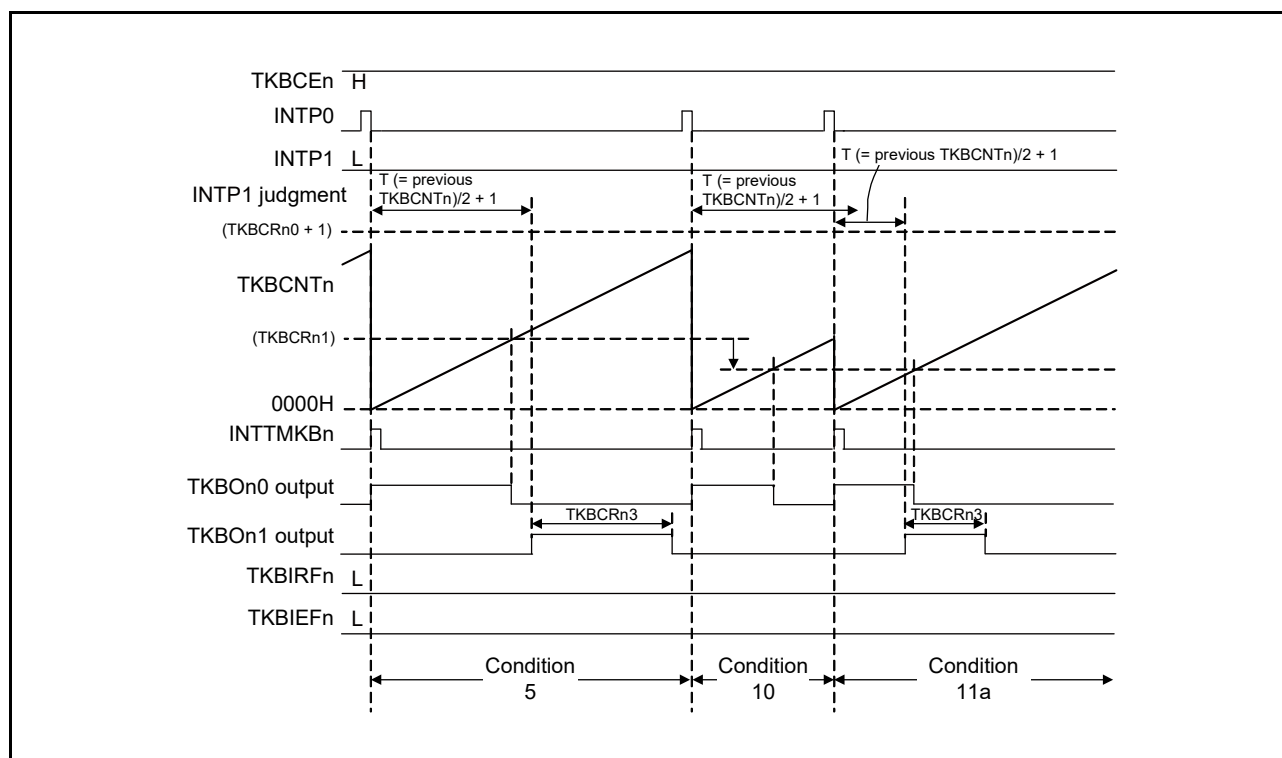
Figure 9 - 64 Timing of Interleaved PFC Output Mode under Condition 8b: INTP1 Is Not Detected in the Current Cycle When the Cycle Length Is No Less Than $T/2$ of That for the Previous Cycle and Less Than $T/2 + T/m$ (m: 8, 16, 32, or 64; Set by the $TKBIRS_n[1:0]$ Bits) of That for the Previous Cycle.



Condition 8b: The $TKBOn1$ output is retained because $INTP0$ was detected before detecting $INTP1$.

Condition 9a: A $TKBOn1$ pulse with the width set in the $TKBCR_n3$ register is output at $T/2$ of the previous cycle.

Figure 9 - 65 Timing of Interleaved PFC Output Mode with a Transition from Condition 10 to 11a: INTP1 Is Not Detected in the Current Cycle When the Cycle Length Is Less Than $T/2$ of That for the Previous Cycle.



Condition 10: Securing $T/2$ of the previous cycle is not possible and the TKBOn1 output is retained.

Condition 11a: A TKBOn1 pulse with the width set in the TKBCRn3 register is output at $T/2$ of the previous cycle.

5. Operations when the $\text{TKBIRSn}[2:0]$ bits are set to 100B or 101B

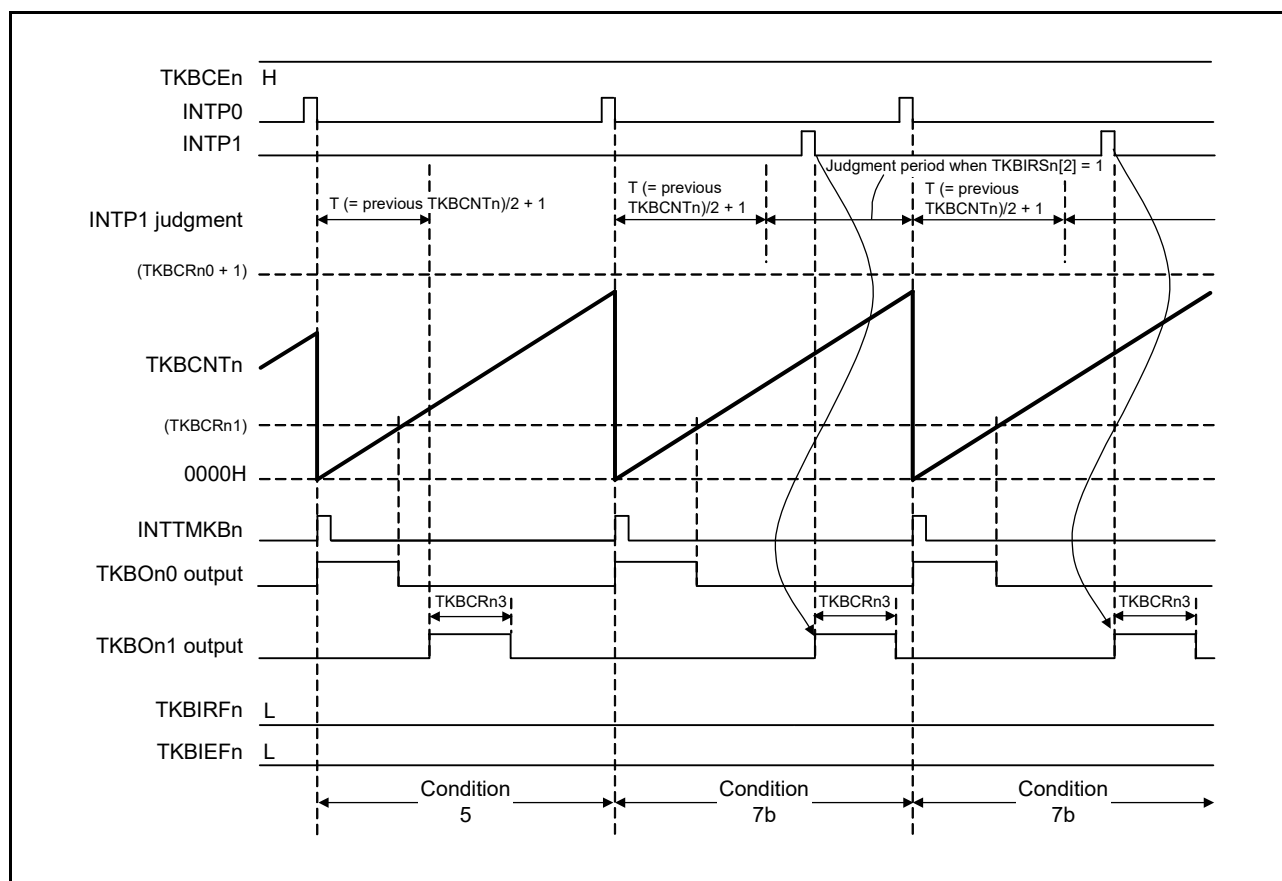
Waveforms during operations under various conditions when the $\text{TKBIRSn}[2:0]$ bits are set to 100B or 101B are given in the following figures.

These waveforms of operations are for cases where the default and active levels for output are low ($\text{TKBTODnp} = 0$) and high ($\text{TKBTOLnp} = 0$), respectively.

As operations under conditions 1 to 5 with $\text{TKBIRSn}[2:0]$ set to 100B or 101B are the same as those with $\text{TKBIRSn}[2:0]$ set to 0xxB, see **Figure 9 - 57** to **Figure 9 - 59**.

In **Figure 9 - 63**, INTP1 is detected after the range of $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $\text{TKBIRSn}[1:0]$ bits) of the previous cycle has elapsed and the value of the $\text{TKBIRSn}[2:0]$ bits is 0xxB. In this case, condition 8a applies to control over the TKBOn1 output. If, on the other hand, the value of the $\text{TKBIRSn}[2:0]$ bits is 100B or 101B and INTP1 is detected as shown in the figure below, condition 7b applies. **Figure 9 - 66** shows the waveforms during operation under condition 7b.

Figure 9 - 66 Timing of Interleaved PFC Output Mode under Condition 7b: INTP1 Is Detected after $T/2$ of the Previous Cycle. [$\text{TKBIRSn}[2:0] = 100\text{B}$ or 101B]

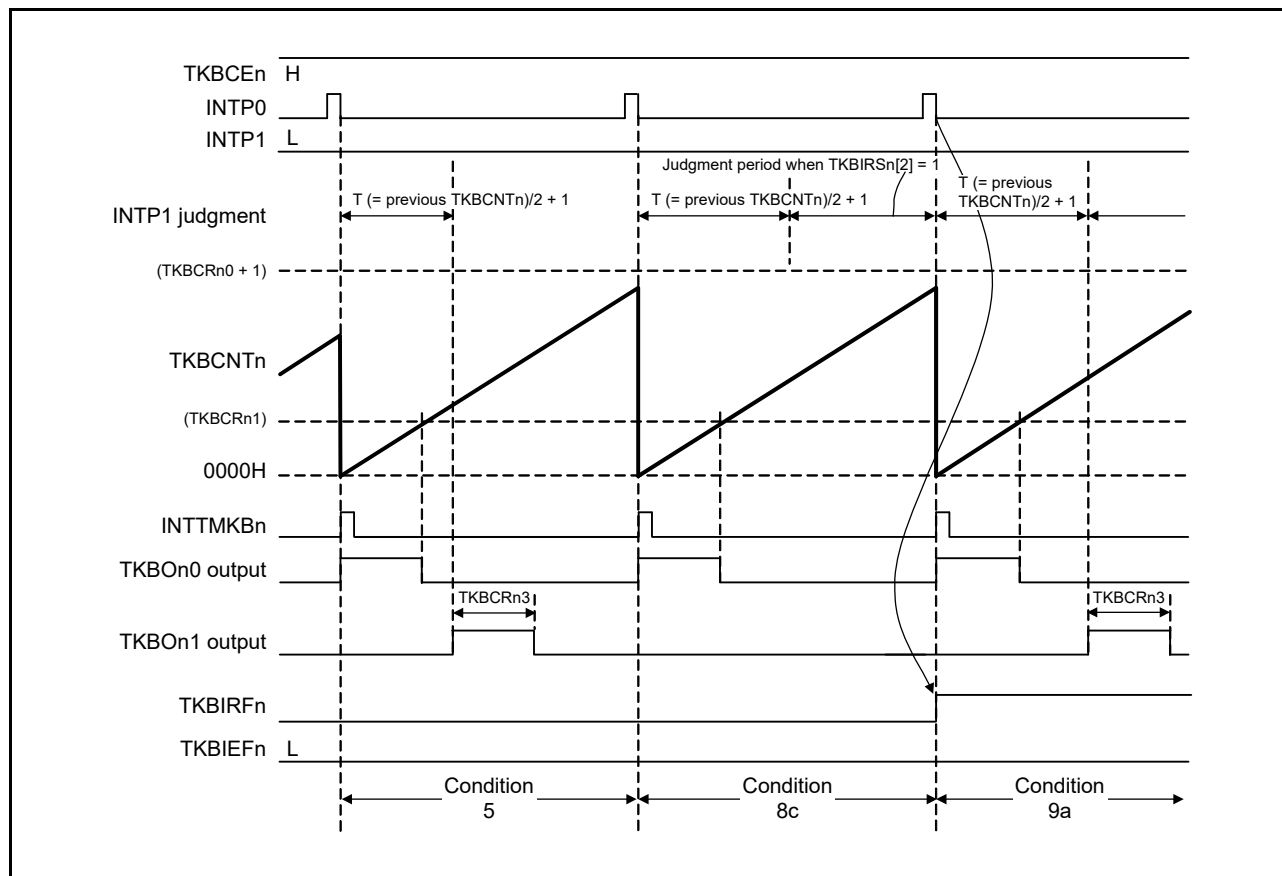


Condition 7b: INTP1 is detected after $T/2$ of the previous cycle, and a TKBOn1 pulse with the width set in the TKBCRn3 register is output.

Assume the case where INTP1 is not detected within the cycle and the value of the $TKBIRSn[2:0]$ bits is 0xxB. In this case, condition 8a also applies to control over the $TKBOn1$ output. If, on the other hand, the value of the $TKBIRSn[2:0]$ bits is 100B or 101B and INTP1 is not detected as shown in the figure below, condition 8c applies.

Figure 9 - 67 shows the waveforms during operation with a transition from condition 8c to condition 9a.

Figure 9 - 67 Timing of Interleaved PFC Output Mode under Conditions 8c and 9a: INTP1 Is Not Detected within the Cycle. [$TKBIRSn[2:0] = 100B$ or $101B$]



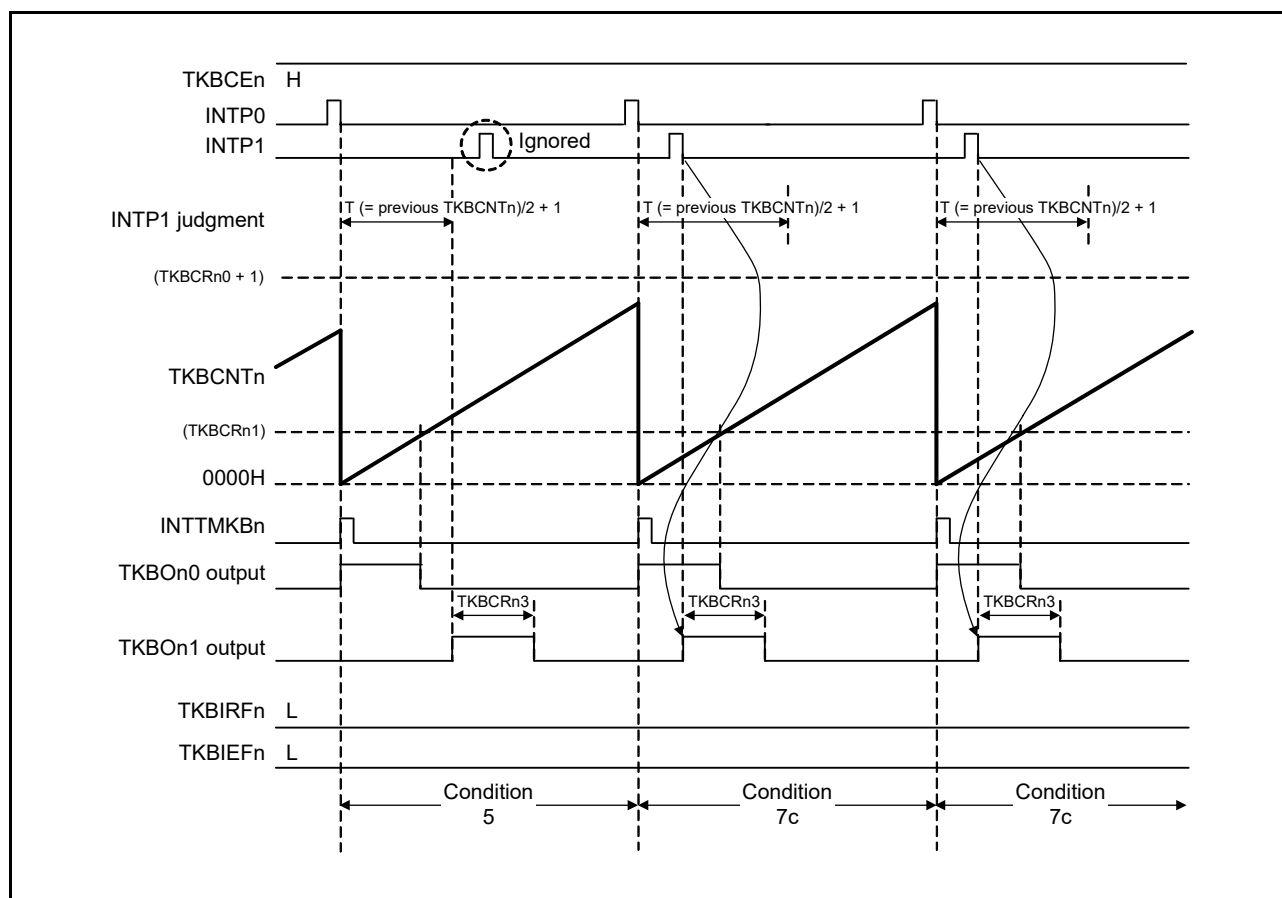
Condition 8c: INTP1 is not detected within the cycle and the $TKBOn1$ output is retained. The $TKBIRFn$ flag is set to 1 on detection of INTP0.

Condition 9a: A $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output at $T/2$ of the previous cycle.

In **Figure 9 - 60**, INTP1 is detected within $T/2$ of the previous cycle and the value of the $TKBIRSn[2:0]$ bits is 0xxB. In this case, condition 6a applies to control over the $TKBOn1$ output. If, on the other hand, the value of the $TKBIRSn[2:0]$ bits is 101B, condition 7c applies. **Figure 9 - 68** shows the waveforms during operation under condition 7c.

Note that if the $TKBIRSn[2:0]$ setting is 100B and the other judgement conditions are the same as those for condition 6a, condition 6c applies to control over the $TKBOn1$ output. A description of operation under condition 6c is omitted because the operation is the same as that under condition 6a.

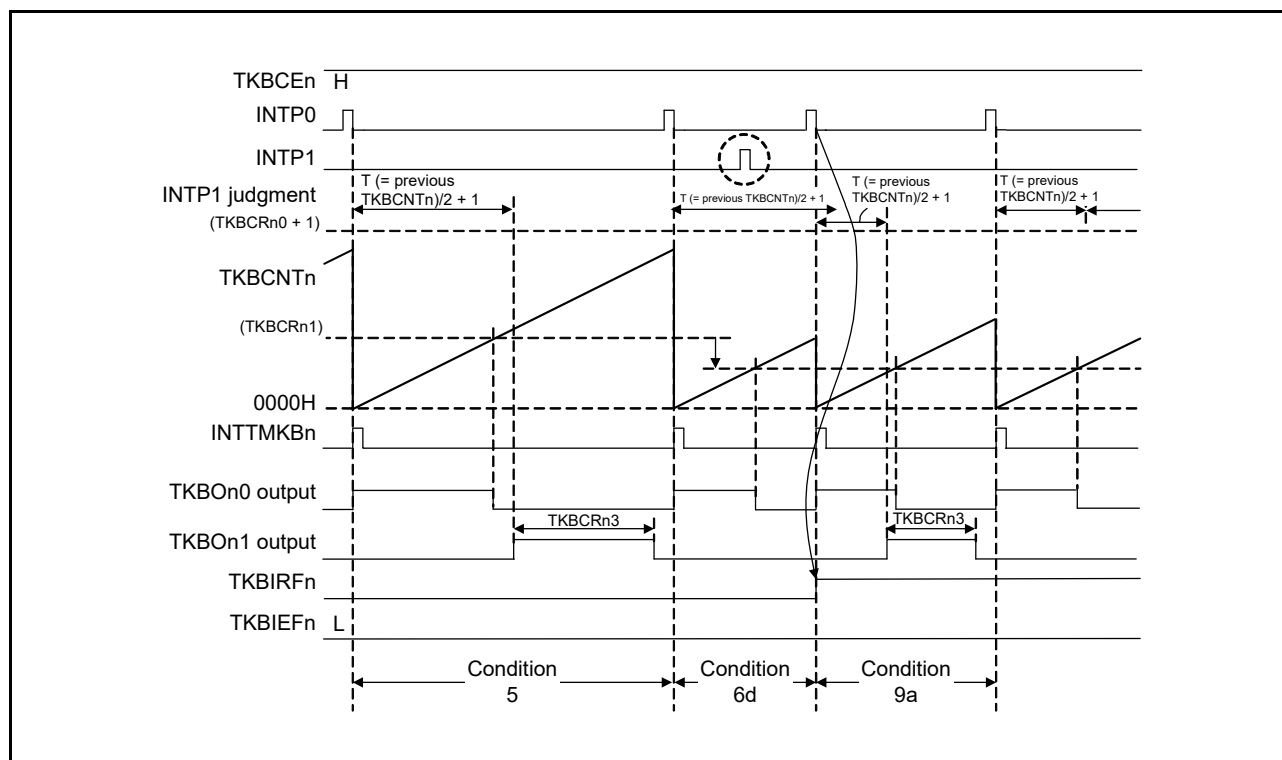
Figure 9 - 68 Timing of Interleaved PFC Output Mode under Condition 7c: INTP1 Is Detected within $T/2$ of the Previous Cycle When the Cycle Length Is No Less Than $T/2$ of That for the Previous Cycle. [$TKBIRSn[2:0] = 101B$]



Condition 7c: INTP1 is detected within $T/2$ of the previous cycle, and a $TKBOn1$ pulse with the width set in the $TKBcRn3$ register is output.

In **Figure 9 - 61**, INTP1 is detected when the cycle length is less than $T/2$ of that for the previous cycle and the value of the $TKBIRSn[2:0]$ bits is 0xxB. In this case, condition 6b applies to control over the $TKBOn1$ output. If, on the other hand, the value of the $TKBIRSn[2:0]$ bits is 100B, condition 6d applies. **Figure 9 - 69** shows the waveforms during operation under condition 6d.

Figure 9 - 69 Timing of Interleaved PFC Output Mode under Condition 6d: INTP1 Is Detected When the Cycle Length Is Less Than $T/2$ of That for the Previous Cycle. [$TKBIRSn[2:0] = 100B$]

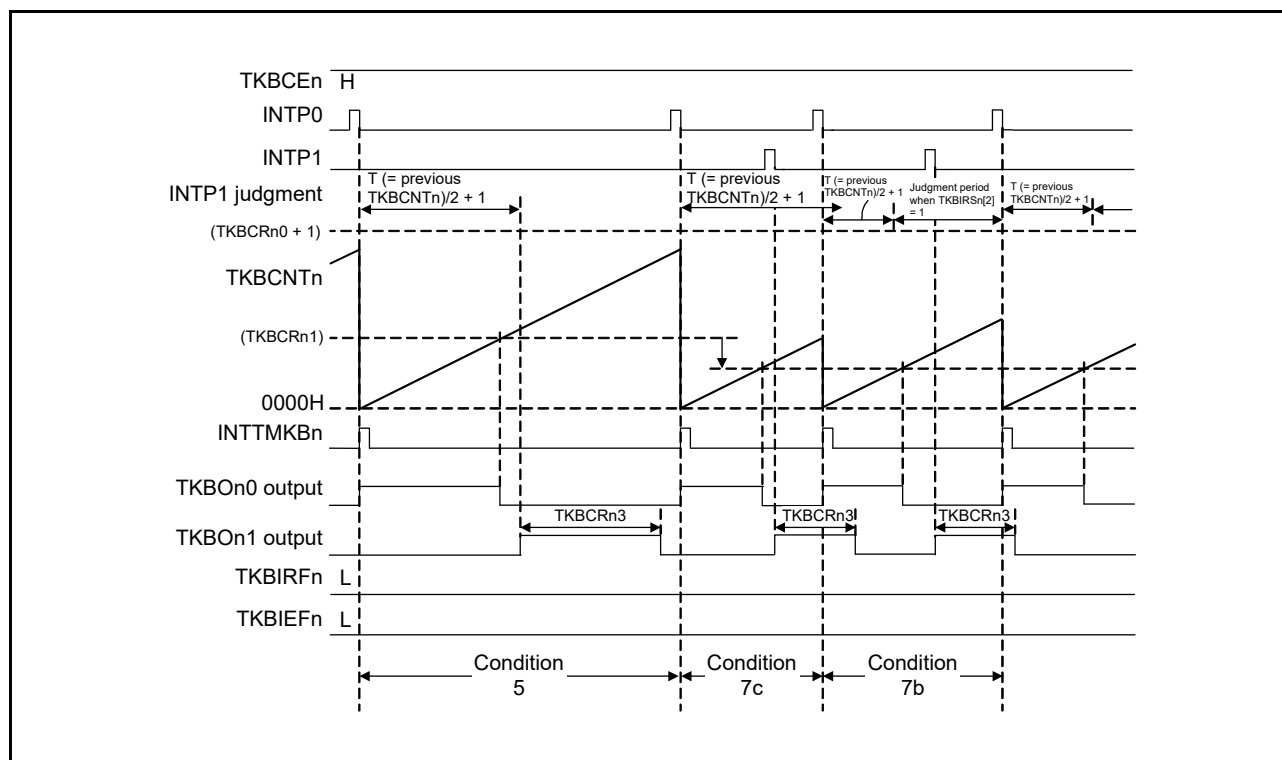


Condition 6d: INTP1 is detected but securing $T/2$ of the previous cycle is not possible. The $TKBOn1$ output is retained. The $TKBIRFn$ flag is set to 1 at this time.

Condition 9a: A $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output at $T/2$ of the previous cycle.

In **Figure 9 - 69**, INTP1 is detected when the cycle length is less than $T/2$ of that for the previous cycle and the value of the $TKBIRSn[2:0]$ bits is 100B. In this case, condition 6d applies to control over the $TKBOn1$ output. If, on the other hand, the value of the $TKBIRSn[2:0]$ bits is 101B, condition 7c applies. **Figure 9 - 70** shows the waveforms during operation under condition 7c.

Figure 9 - 70 Timing of Interleaved PFC Output Mode under Condition 7c: INTP1 Is Detected When the Cycle Length Is Less Than $T/2$ of That for the Previous Cycle. [$TKBIRSn[2:0] = 101B$]



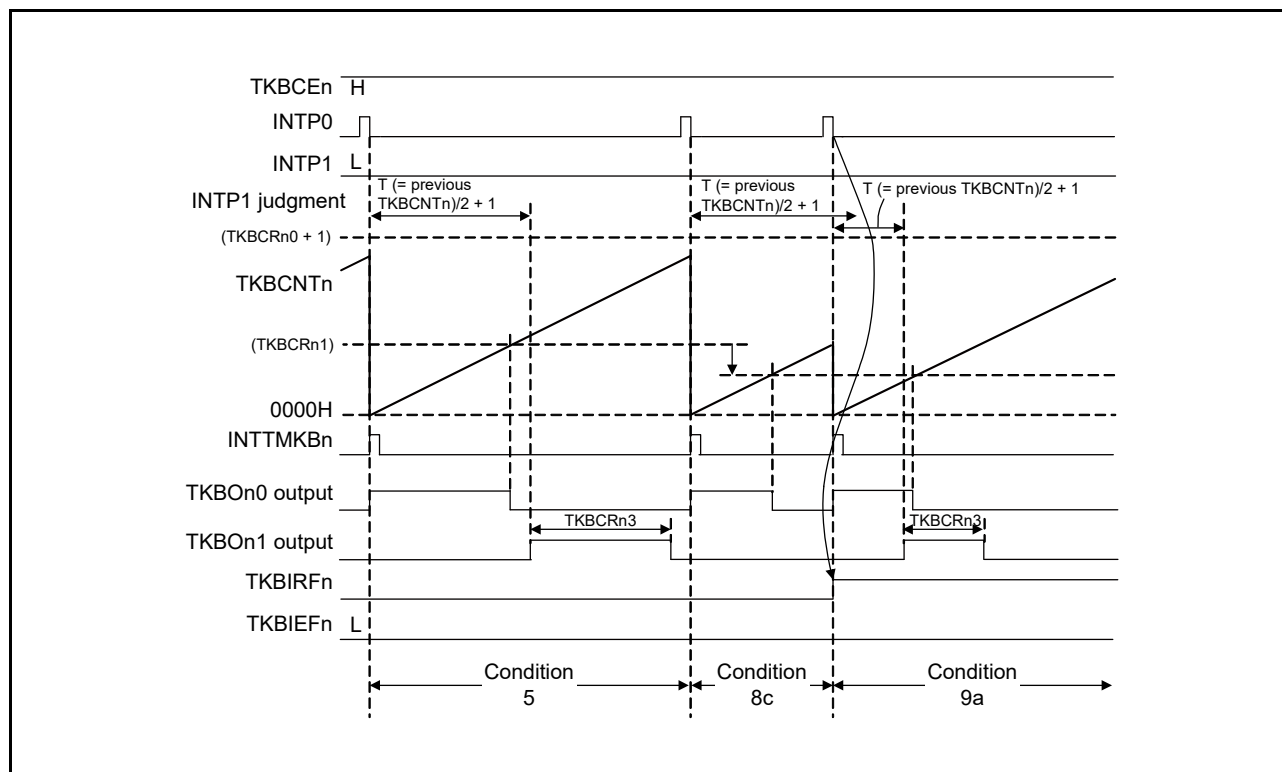
Condition 7c: INTP1 is detected within $T/2$ of the previous cycle, and a $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output.

Condition 7b: INTP1 is detected after $T/2$ of the previous cycle, and a $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output.

In **Figure 9 - 65**, INTP1 is not detected in the current cycle when the cycle length is less than $T/2$ of that for the previous cycle and the value of the $TKBIRSn[2:0]$ bits is 0xxB. In this case, condition 10 applies to control over the $TKBOn1$ output. If, on the other hand, the value of the $TKBIRSn[2:0]$ bits is 100B or 101B, condition 8c applies.

Figure 9 - 71 shows the waveforms during operation under condition 8c.

Figure 9 - 71 Timing of Interleaved PFC Output Mode with a Transition from Condition 8c to 9a: INTP1 Is Not Detected in the Current Cycle When the Cycle Length Is Less Than $T/2$ of That for the Previous Cycle. [$TKBIRSn[2:0] = 100B$ or $101B$]



Condition 8c: Securing $T/2$ of the previous cycle is not possible and the $TKBOn1$ output is retained. The $TKBIRFn$ flag is set to 1 on detection of INTP0.

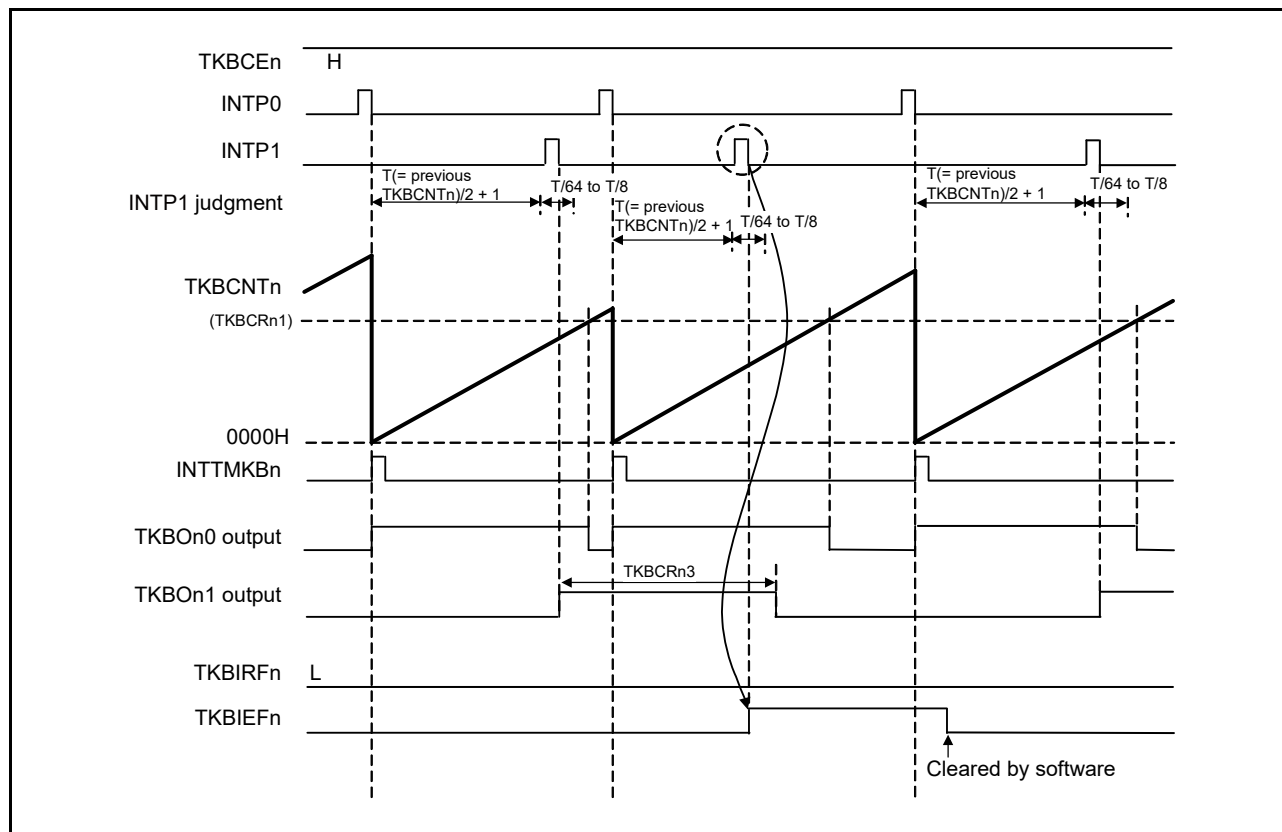
Condition 9a: A $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output at $T/2$ of the previous cycle.

6. Idiosyncratic points in operations of the TKBOn1 output

Regarding idiosyncratic points in operations of the TKBOn1 output, **Figure 9 - 72** shows the waveforms when INTP1 input is detected while the TKBOn1 output is at the active level, and **Figure 9 - 73** shows the waveforms when the period over which the previous TKBOn1 output is at the active level crosses over into the next cycle and exceeds the state retention period.

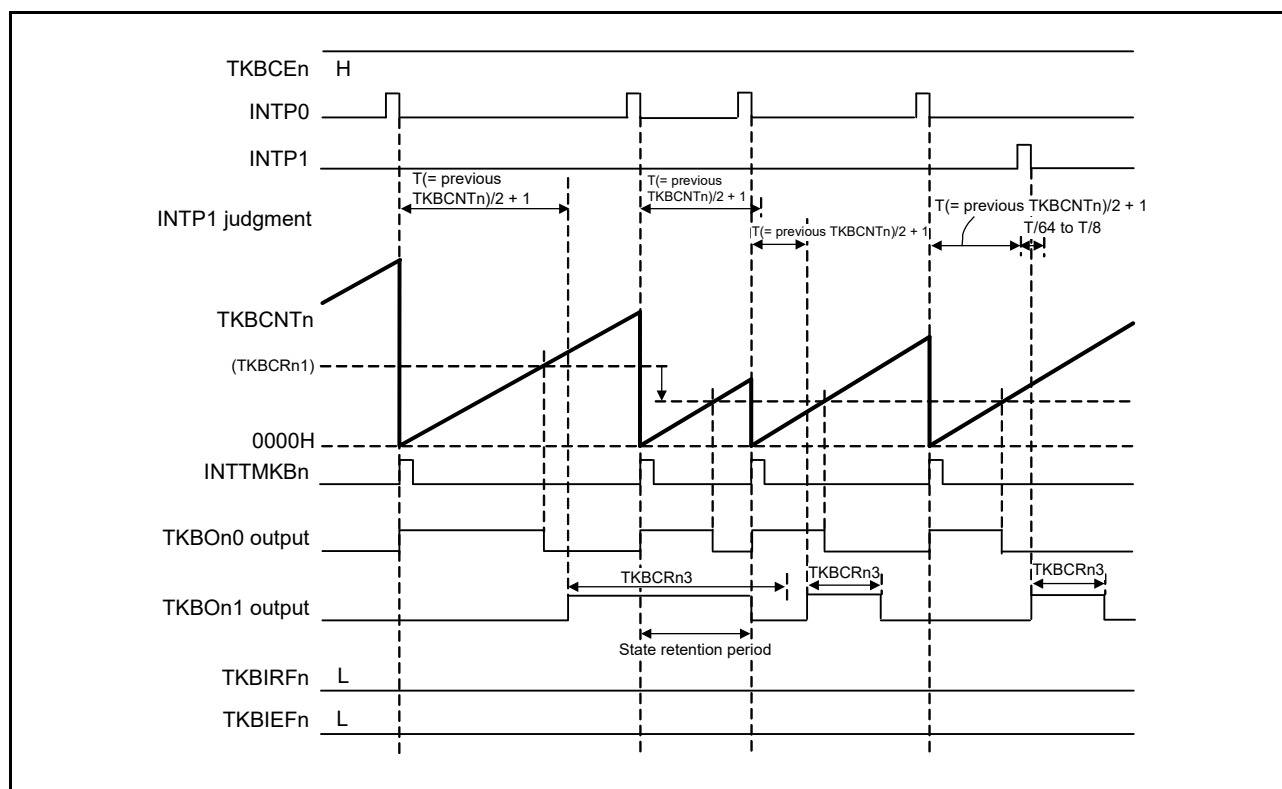
These waveforms of operations are for cases where the default and active levels for output are low (TKBTODnp = 0) and high (TKBTOLnp = 0), respectively.

Figure 9 - 72 Timing of Interleaved PFC Output Mode: Another INTP1 Input Is Redundantly Detected While the TKBOn1 Output Is at the Active Level.



When INTP1 input is detected while the TKBOn1 output is at the active level, this trigger is ignored. The TKBIEFn flag is set to 1 at this time.

Figure 9 - 73 Timing of Interleaved PFC Output Mode: The Period over Which TKBOn1 Output Is at the Active Level Crosses over into the Next Cycle and Exceeds the State Retention Period.



When the period over which TKBOn1 output is at the active level crosses over into the next cycle and exceeds the state retention period, the output is forcibly placed at the default level at the time the next cycle starts following the state retention period.

7. List of register settings in interleaved PFC output mode

TKBCTLn0	15	14	13	12	11	10	9	8
	TKBTGMn 1 1/0	— 0	TKBSSEn1 0	TKBDIE1 0	TKBTGMn 0 1/0	— 0	TKBSSEn0 0	TKBDIE0 0
	7	6	5	4	3	2	1	0
	TKBMFEn 1/0	TKBIRSn2 1/0	TKBIRSn1 1/0	TKBIRSn0 1/0	TKBIHEn 0	TKBTSEn 1	TKBSTSn1 0	TKBSTSn0 0
TKBCTLn1	7	6	5	4	3	2	1	0
	TKBCEn 1/0	— 0	— 0	TKBCKSn 1/0	TKBSCMn 0	— 0	TKBMDn1 1	TKBMDn0 1
TKBIOcn0	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	TKBTOLn1 1/0	TKBTOLn0 1/0	TKBTODn1 1/0	TKBTODn0 1/0
TKBIOcn1	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	— 0	TKBTOEn1 1/0	TKBTOEn0 1/0
TKBCTLn2	15	14	13	12	11	10	9	8
	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0
	7	6	5	4	3	2	1	0
	— 0	— 0	— 0	— 0	— 0	TKBINSn2 0	TKBINSn1 0	TKBINSn0 0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H
TKBSIRn1	0000H
TKBSSRn0	00H
TKBSSRn1	00H
TKBDNRn0	00H
TKBDNRn1	00H
TKBMFRn	0000H to FFFFH



: The setting is fixed in this mode.



: Setting a value is not required (leave it as the default value).

9.5 Optional Functions of 16-bit Timers KB40, KB41, and KB42

Optional functions can be added for the 16-bit timers KB40, KB41, and KB42. The following table shows available optional functions for each operating mode of the 16-bit timers KB40, KB41, and KB42.

Operating Mode		Standalone Mode		Simultaneous Start/Stop Mode		Simultaneous Start/Clear Mode ^{Note 1}	Interleaved PFC Output Mode ^{Note 1}
Period Controlling Method for Each Operating Mode		Period Control by the TKBCRn0 Register	Period Control by Trigger Input	Period Control by the TKBCRn0 Register	Period Control by Trigger Input	Period Control by the Master	Period Control by INTP0 Input and the TKBCRn0 Register
Optional function	A/D conversion start timing signal output	✓	✓	✓	✓	✓	✓
	PWM output dithering ^{Note 2}	✓	✓	✓	✓	✓	×
	PWM output smooth start ^{Note 2}	✓	✓	✓	✓	✓	×
	Masking of external trigger inputs while the TKBOnp output is active ^{Note 2}	×	✓	×	✓	×	✓
	Maximum frequency limiting ^{Note 2}	×	✓	×	✓	×	✓

Note 1. PWM output for IH control cannot be used in the simultaneous start/clear mode and the interleaved PFC output mode.

Note 2. On using PWM output for IH control, the following functions cannot be used: PWM output dithering, PWM output smooth start, masking of external trigger inputs while the TKBOnp output is active, and maximum frequency limiting.

Remark For details of the operation specifications, see **9.4.2 Default level and active level** and **9.4.3 Stopping and restarting operation**. The trigger used for controlling the period indicates the external trigger selected by the TKBSTSn[1:0] and TKBINSn[2:0] bits.

9.5.1 A/D conversion start timing signal output

Output of the A/D conversion start timing signal can be controlled by setting the 16-bit timer KB trigger compare register n (TKBTGCR n) of a 16-bit timer KB4 n . This enables synchronization of the timing of the start of A/D conversion with a 16-bit timer KB4 n . The trigger output signal of a 16-bit timer KB4 n is output on detection of a match between the TKBCNT n and TKBTGCR n registers. Trigger output is available with any timing regardless of the cycle length specified in the TKBCR $n0$ register. The output width of the trigger output signal of a 16-bit timer KB4 n is the width of 1 clock cycle of the timer clock. Trigger output timing from the start of a PWM output cycle can be calculated from the following formula.

Trigger output timing = setting of the TKBTGCR n register $\times$ counter clock period

Caution The trigger output signal of the 16-bit timer KB4 n is not output when the TKBCR $n0$ setting is less than the TKBTGCR n setting.

Figure 9 - 74 A/D Conversion Start Timing Signal Output in the Standalone Mode under Period Control by the TKBCR $n0$ Register

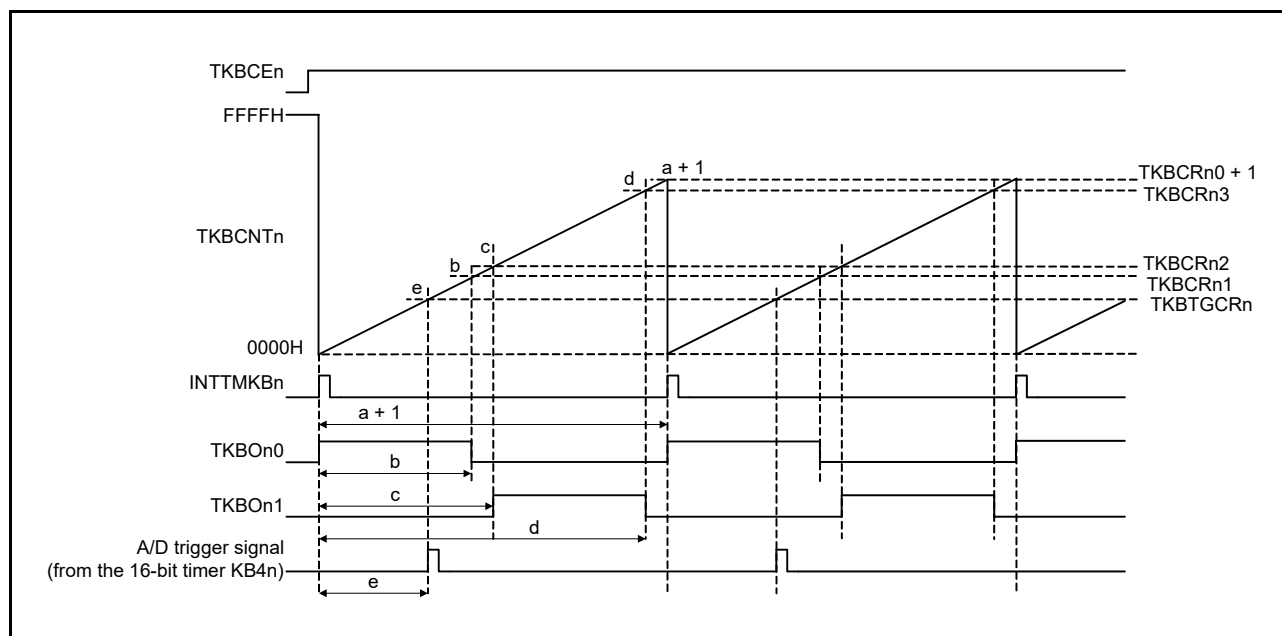
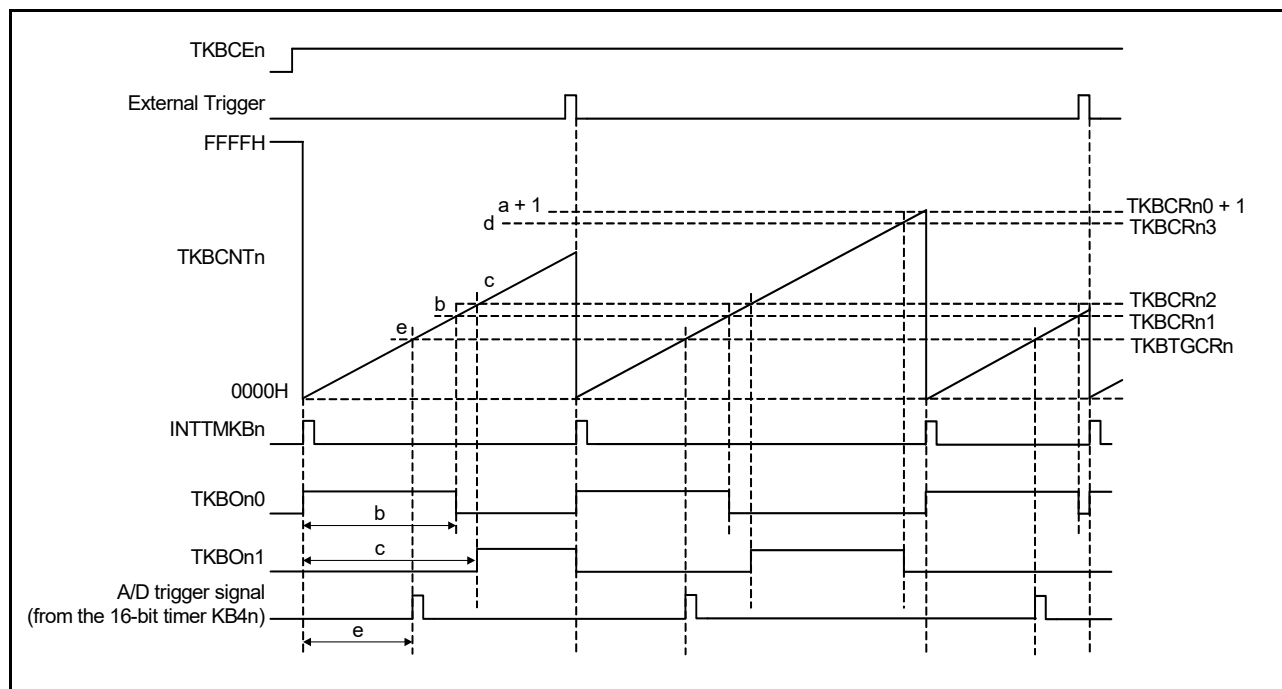


Figure 9 - 75 A/D Conversion Start Timing Signal Output in the Standalone Mode under Period Control by External Trigger Input



9.5.2 PWM output dithering

The 16-bit timers KB40, KB41, and KB42 are capable of producing high-resolution PWM output by using PWM output dithering. Taking 16 cycles of the PWM period as standard, PWM output with 16-times higher resolution as the average can be obtained by extending the active period by one cycle of the counter clock in n cycles ($n = 0$ to 15) of every 16 cycles. The number of cycles in which the active periods of 16 PWM cycles are extended by one cycle of the counter clock is defined by the TKBDNRnp bit. **Table 9 - 4** lists the relationship between the setting of the TKBDNRnp bit and the cycles with the active period extended by one cycle of the counter clock.

Table 9 - 4 Relationship between the Setting of the TKBDNRnp Bit and the Cycles with the Active Period Extended by One Cycle of the Counter Clock

Cycle Number of repetitions (N)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

Remark 1.  Cell cycle: Inactive output via TKBCRn1 and TKBCRn3 register settings

 Cell cycle: Inactive output via TKBCRn1 and TKBCRn3 register settings + 1

Remark 2. $n = 0$ to 2 ; $p = 0, 1$

Figure 9 - 76 Waveforms in Dithering Operation

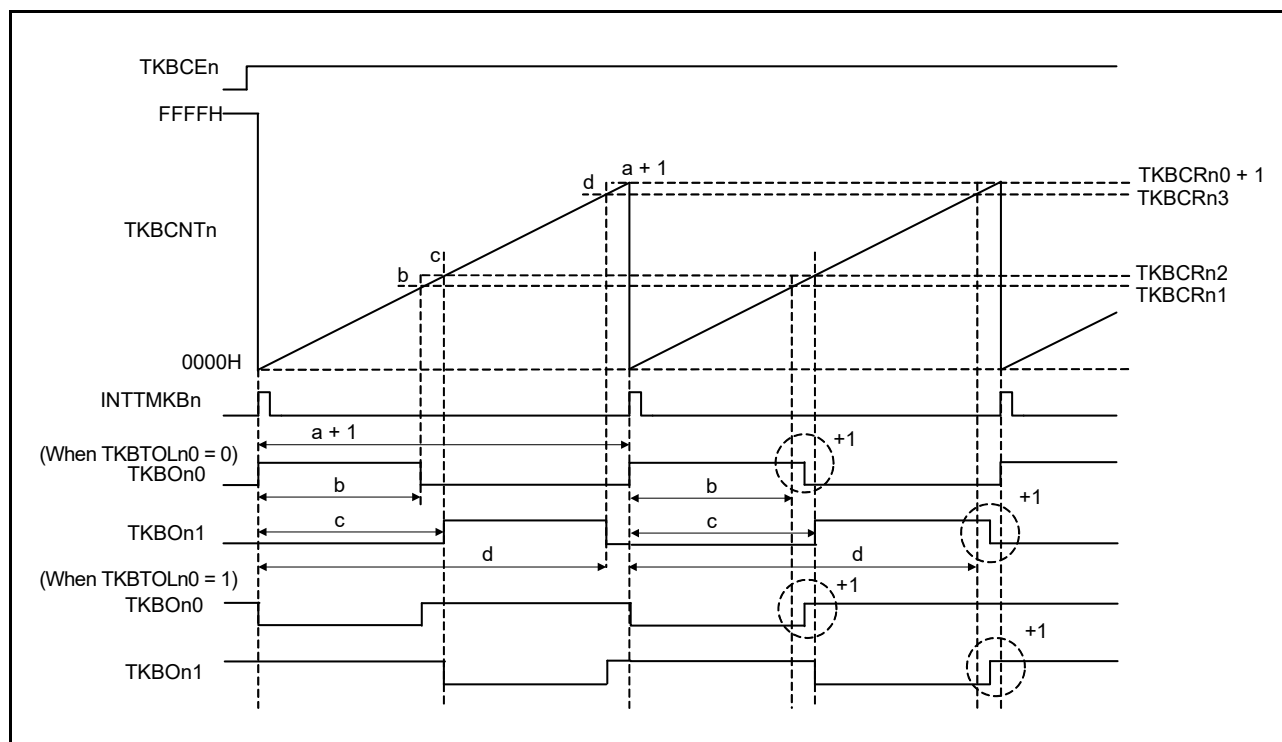
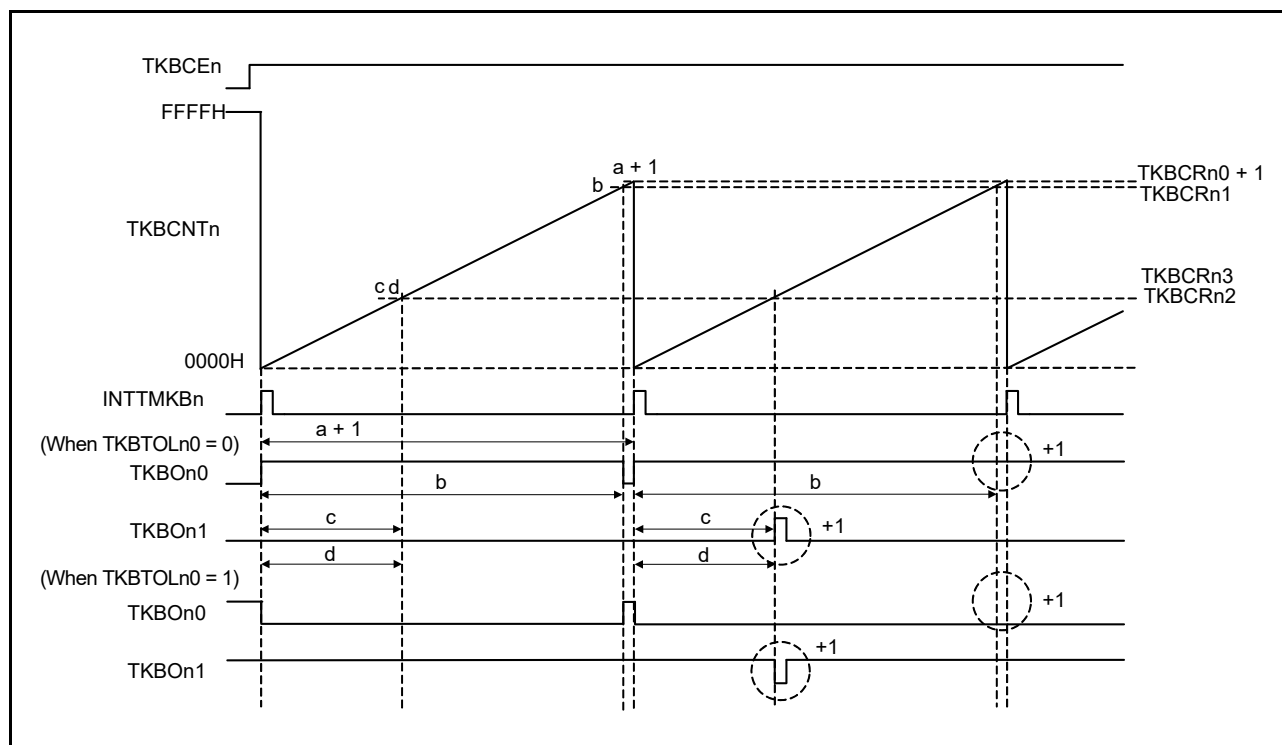
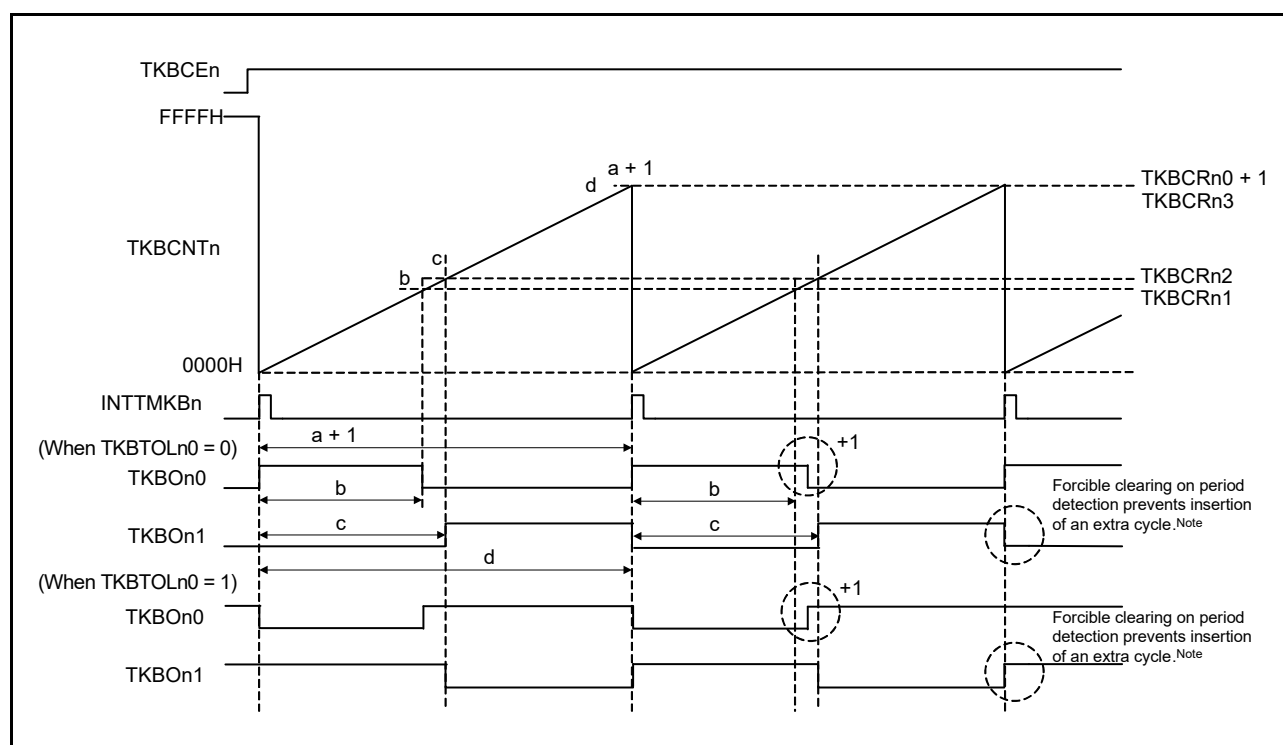
Figure 9 - 77 Waveforms in Dithering Operation When $TKBCRn1 = TKBCRn0$ (the Duty Cycle Starts Close to and Converges on 100%) and $TKBCRn2 = TKBCRn3$ (the Duty Cycle Starts Close to and Converges on 0%)

Figure 9 - 78 Waveforms in Dithering Operation When $TKBCRn3 = TKBCRn0 + 1$



Note	The TKBON1 output is forcibly cleared on detection of a match between the TKBCNTn counter and TKBCRN0 register. This prevents extension by one clock cycle of the width at the active level in the TKBON1 output cycle.

1. Operating modes in which PWM output dithering can be used

The following table shows whether PWM output dithering can be used in each of the modes specified by the `TKBCTLn1.TKBMDn[1:0]` bits.

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (with the period controlled by the <code>TKBCRn0</code> register)	00B	✓
Standalone mode (with the period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (with the period controlled by the <code>TKBCRn0</code> register)	01B	✓
Simultaneous start/stop mode (with the period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (with the period controlled by the master)	10B	✓
Interleaved PFC output mode	11B	×

The `TKBDNRn0` and `TKBDNRn1` registers are respectively used to control PWM output dithering on `TKBOn0` and `TKBOn1`.

Caution 1. [Overwriting the `TKBDNRn0` or `TKBDNRn1` register while the timer is counting (`TKBCEn = 1`)]
Overwriting the `TKBDNRn0` or `TKBDNRn1` register while the timer is counting (`TKBCEn = 1`) is possible because each of these registers has a buffer. At this time, batch overwriting proceeds in response to writing 1 to the `TKBRDTn` bit.

Caution 2. [Access by the `TKBCRLDn0` or `TKBCRLDn1` register]

- The `TKBCRLDn0` register is a 16-bit register mapping the eight lower-order bits of the `TKBCRn1` register and the `TKBDNRn0` register.
- The `TKBCRLDn1` register is a 16-bit register mapping the eight lower-order bits of the `TKBCRn3` register and the `TKBDNRn1` register.
- The value in the `TKBDNRn0` or `TKBDNRn1` register is respectively changed even by the access to the `TKBCRLDn0` or `TKBCRLDn1` register.
- The value in the `TKBCRn1` or `TKBCRn3` register is respectively changed even by the access to the `TKBCRLDn0` or `TKBCRLDn1` register. Note that only the eight lower-order bits of the `TKBCRn1` or `TKBCRn3` register are changed by the access to the `TKBCRLDn0` or `TKBCRLDn1` register.

Caution 3. [For using both PWM output smooth start and PWM output dithering]

- PWM output dithering is disabled while PWM output smooth start is in use (`TKBSSFnp = 1`).
- PWM output dithering is enabled while PWM output smooth start is not in use (`TKBSSFnp = 0`).

9.5.3 PWM output smooth start

The 16-bit timers KB40, KB41, and KB42 support smooth starting of PWM output to control inrush currents and to prevent overvoltages. This function starts at the time a timer starts operating. Processing that users would otherwise have needed to implement through software can easily be accomplished with this optional function of the hardware. The function generates a PWM waveform with the active period per cycle set by the 16-bit timer KB smooth start initial duty register np (TKBSIRnp) and outputs it for n (set by the 16-bit timer KB smooth start step width register np (TKBSSRnp)) + 1 cycles. After that, the active period is incremented by one clock cycle and output of the resulting PWM waveform continues for TKBSSRnp setting + 1 cycles. After that operation is repeated as many times as is required, smooth starting of PWM output is stopped when the active period reaches the value defined in the TKBCRn1 and TKBCRn3 registers.

The 16-bit timer KB smooth start initial duty register np should be set according to the following conditions.

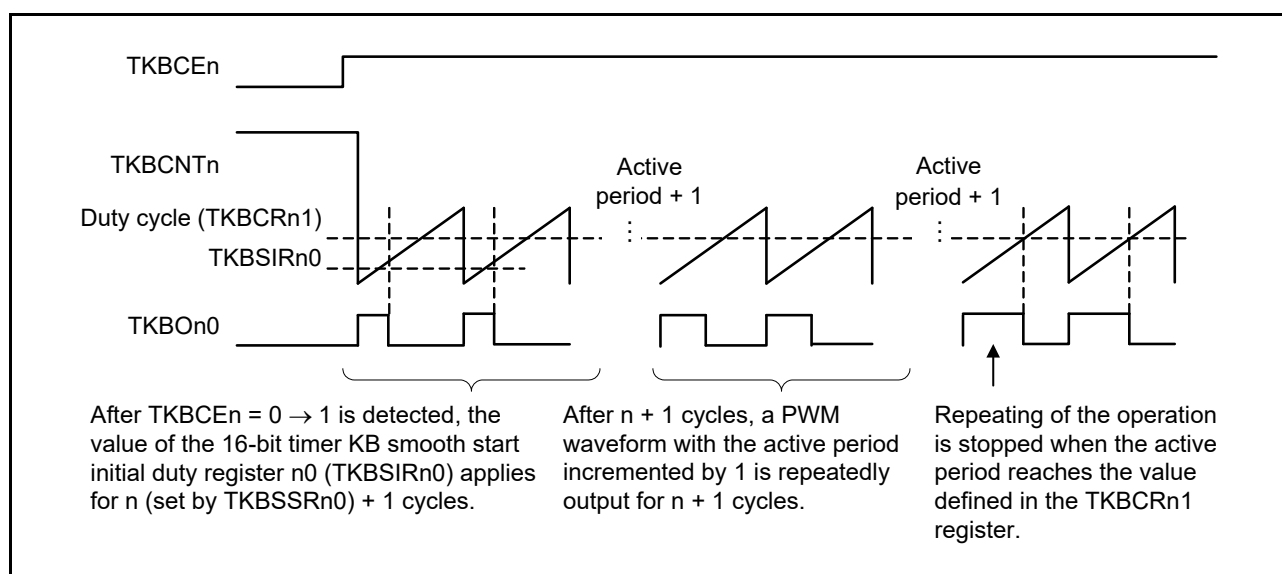
$0000H \leq \text{TKBSIRn0} < \text{TKBCRn1} \leq \text{TKBCRn0} + 1$

$\text{TKBCRn2} \leq \text{TKBSIRn1} < \text{TKBCRn3} \leq \text{TKBCRn0} + 1$

It should be set according to following condition while the simultaneous start/clear mode is in use.

$\text{TKBCRn0} \leq \text{TKBSIRn0} < \text{TKBCRn1} \leq \text{TKBCR00} + 1$ of Master

Figure 9 - 79 PWM Output Smooth Start



1. Operating modes in which PWM output smooth start can be used

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (with the period controlled by the TKBCRn0 register)	00B	✓
Standalone mode (with the period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (with the period controlled by the TKBCRn0 register)	01B	✓
Simultaneous start/stop mode (with the period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (with the period controlled by the master)	10B	✓
Interleaved PFC output mode	11B	×

The pairs of the TKBSIRn0 and TKBSSRn0 registers and of the TKBSIRn1 and TKBSSRn1 registers are respectively used to control smooth starting of PWM output on TKBOn0 and TKBOn1.

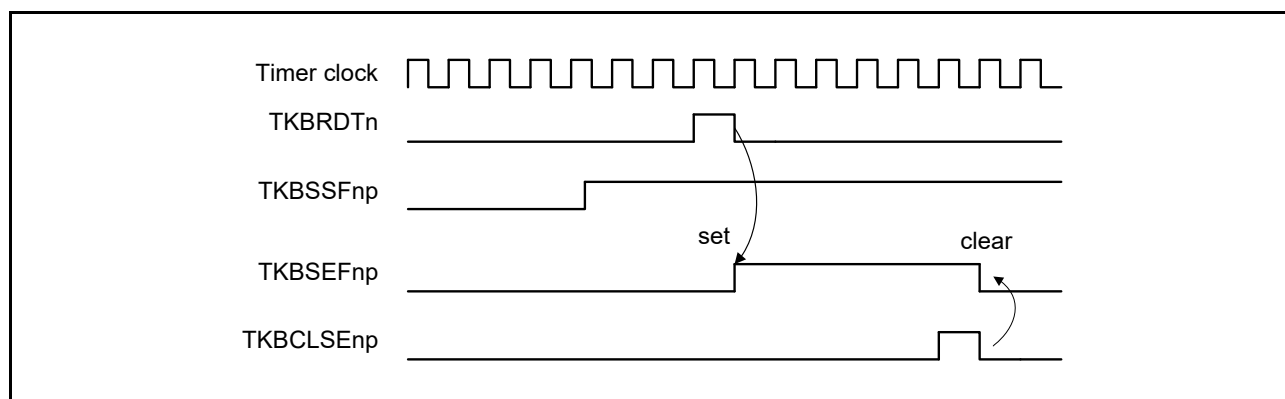
2. Overwriting the TKBSIRn0, TKBSIRn1, TKBSSRn0, or TKBSSRn1 register while the timer is counting (TKBCEn = 1)

Overwriting the TKBSIRn0, TKBSIRn1, TKBSSRn0, or TKBSSRn1 register while the timer is counting (TKBCEn = 1) is available. Each register has a buffer and batch overwriting can be implemented to the registers by writing 1 to the TKBRDTn bit.

3. Overwriting the TKBCRn0, TKBCRn1, TKBCRn2, TKBCRn3, TKBSIRn0, TKBSIRn1, TKBSSRn0, or TKBSSRn1 register while the timer is counting (TKBCEn = 1)

When the TKBRDTn bit is set to 1 while PWM output smooth start is in use (TKBSSFn0 = 1 or TKBSSFn1 = 1), batch overwriting is masked and the TKBSEFnp flag is set. For implementing batch overwriting, clear the TKBSEFnp flag, confirm that the TKBSSFnp flag has become 0, and then set the TKBRDTn bit to 1.

Figure 9 - 80 Overwriting the TKBCRn0, TKBCRn1, TKBCRn2, TKBCRn3, TKBSIRn0, TKBSIRn1, TKBSSRn0, or TKBSSRn1 Register While Smooth Start Is in Use (TKBSSFnp = 1)



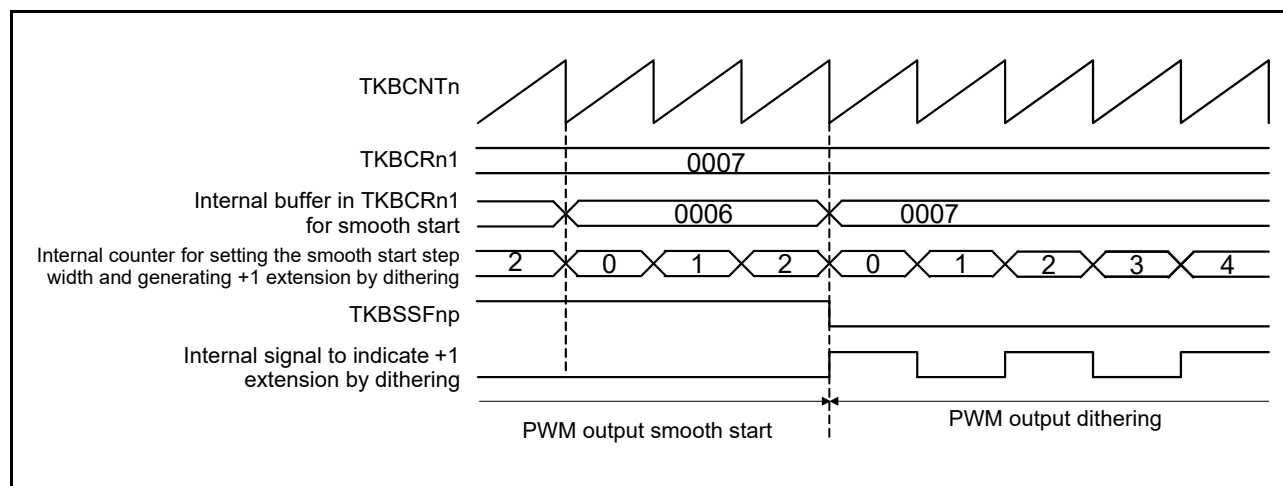
4. For using both PWM output smooth start and PWM output dithering

PWM output dithering is disabled while PWM output smooth start is in use (TKBSSFnp = 1). PWM output dithering is enabled while PWM output smooth start is not in use (TKBSSFnp = 0).

5. Cancellation of PWM output smooth start and the operation of TKBSSFnp

Figure 9 - 81 shows the example when the TKBCRn1, TKBDNRnp, and TKBSSRnp registers are set to 0007H, 70H, and 02H, respectively. At the timing that TKBCRn1 = 0007H and the value of the internal buffer in TKBCRn1 for smooth start match, the TKBSSFnp flag is cleared, and then dithering starts.

Figure 9 - 81 Cancellation of PWM Output Smooth Start and the Operation of TKBSSFnp



9.5.4 Maximum frequency limiting

The 16-bit timers KB40, KB41, and KB42 support this function to regulate the minimum period (maximum frequency) for counter clearing in period control by external trigger input or in the interleaved PFC output mode. On generation of an external trigger input for counter clearing while the counter value is smaller than the value set in the 16-bit timer KB maximum frequency limit setting register n (TKBMFRn), using this function causes the input to be held pending, with the counter being cleared after counting has continued to the value set in the TKBMFRn register.

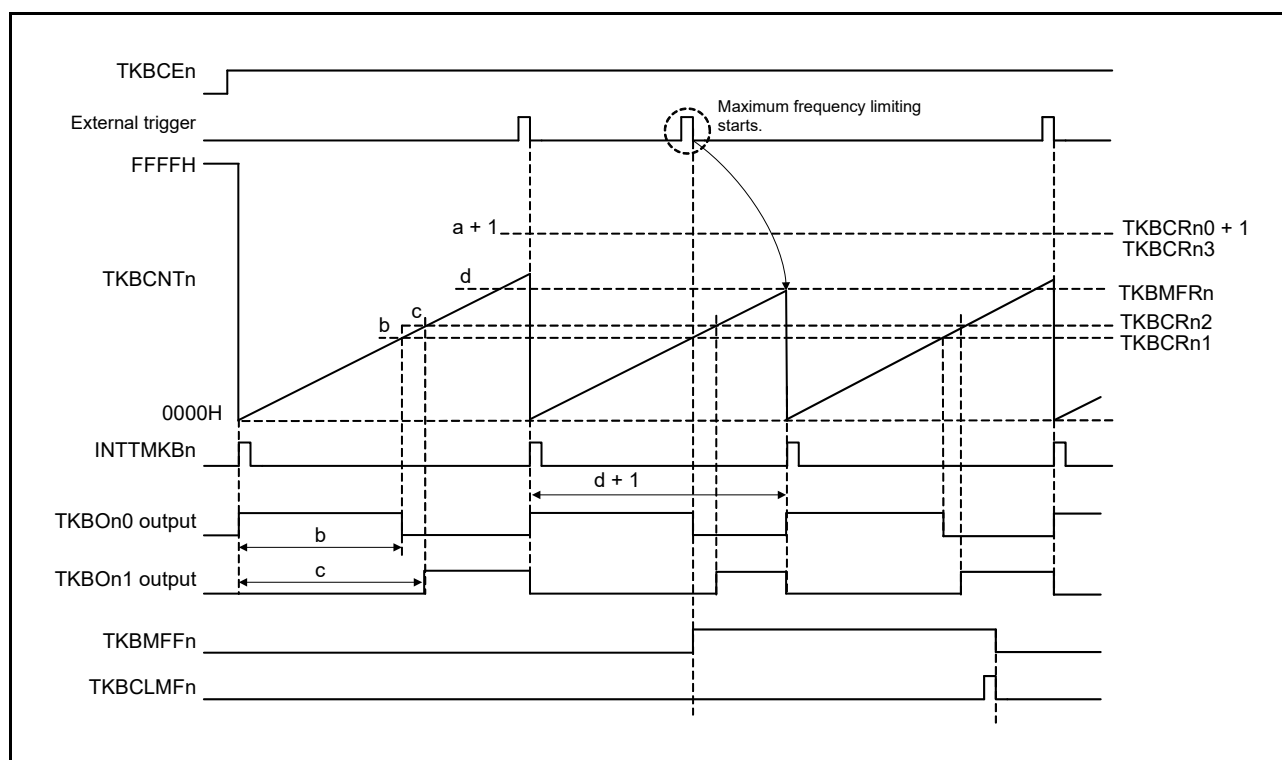
1. Formula for the maximum frequency limit (= 1/Minimum period)

$$\text{Minimum period} (= 1/\text{Maximum frequency limit}) = (\text{TKBMFRn setting} + 1) \times \text{counter clock period}$$

Caution Make sure that the TKBMFRn setting is no greater than the TKBCRn0 setting.

When the counter value is smaller than the value in the TKBMFRn register at the timing of detecting an external trigger input, the TKBMFFn flag is set to 1. The TKBMFFn flag is cleared to 0 by writing 1 to the TKBCLMFn bit.

Figure 9 - 82 Maximum Frequency Limiting under Period Control by External Trigger Input



2. Operating modes in which maximum frequency limiting can be used

The following table shows whether maximum frequency limiting can be used in each of the modes.

Operating Mode	TKBMDn[1:0]	Setting Available
Standalone mode (with the period controlled by the TKBCRn0 register)	00B	×
Standalone mode (with the period controlled by external trigger input)	00B	✓
Simultaneous start/stop mode (with the period controlled by the TKBCRn0 register)	01B	×
Simultaneous start/stop mode (with the period controlled by external trigger input)	01B	✓
Simultaneous start/clear mode (with the period controlled by the master)	10B	×
Interleaved PFC output mode	11B	✓

Remark Maximum frequency limiting can be used when the period is controlled by external trigger input. Period control for the slaves in the simultaneous start/clear mode is implemented in the same way as that for the master. When the standalone mode (with the period controlled by external trigger input) is selected for the master, maximum frequency limiting can be used in the master. Set the TKBCTLn0.TKBMFE_n bit to 0 for the slaves so that maximum frequency limiting is not applied to them at this time.

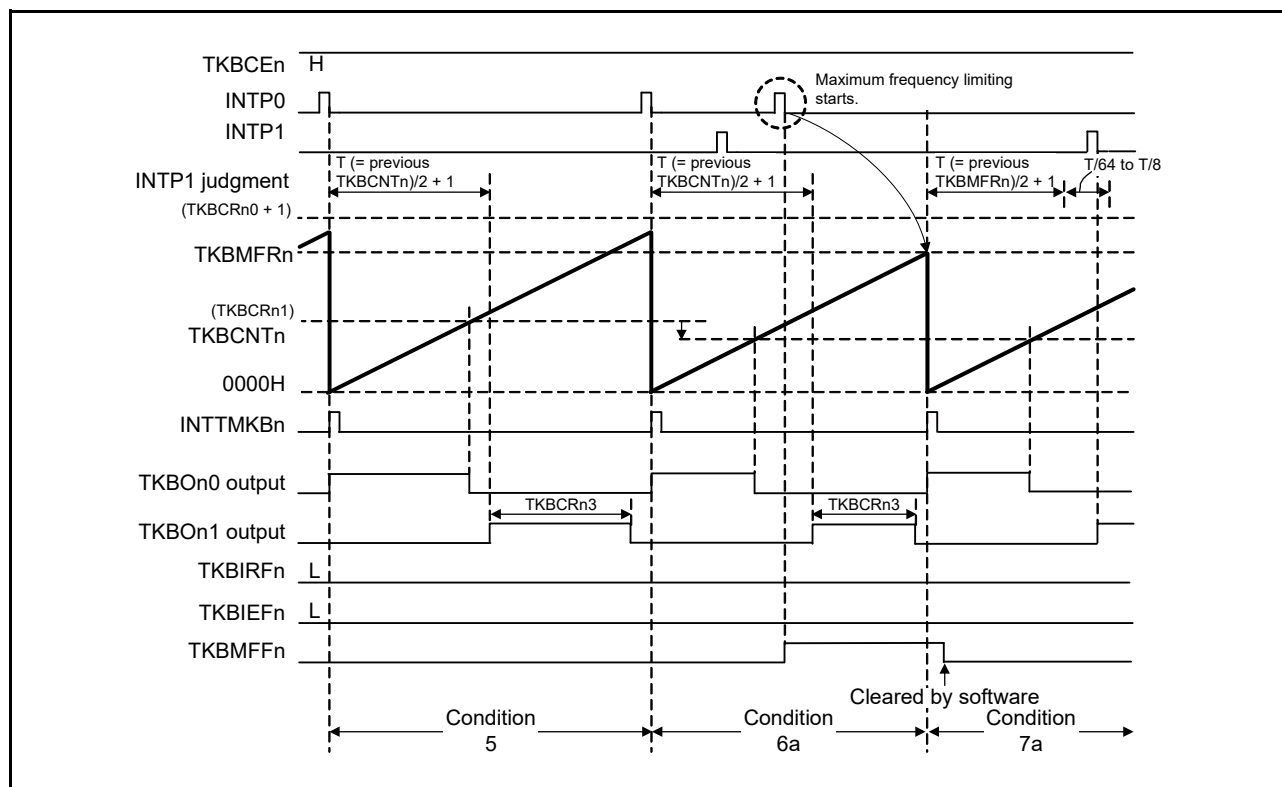
3. Usage of maximum frequency limiting in the interleaved PFC output mode

In the interleaved PFC output mode, detecting INTP0 is required for maximum frequency limiting to start. The condition for TKBOn1 output when maximum frequency limiting has started differs according to the setting of the TKBIRS_n[2] bit. The following table shows the operations depending on the setting of the TKBIRS_n[2] bit.

TKBIRS _n [2]	Control of TKBOn1 Output
0	Regardless of whether maximum frequency limiting has started, TKBOn1 is output when the timing of the detection of INTP0 and INTP1 is judged to satisfy the range for judgment specified by the TKBIRS _n [1:0] bits.
1	Regardless of whether INTP1 has been detected, TKBOn1 is output at the timing of INTP0 detection, which leads to starting of maximum frequency limiting.

Using maximum frequency limiting leads to the detection of INTP1 in the period corresponding to $T/2$ of the previous cycle in the next judgment cycle under condition 5, as shown in **Figure 9 - 61**. Here, a case in which the judgment cycle length is less than $T/2$ of the previous cycle and therefore TKBOn1 is not to be output (condition 6b) can be suppressed. **Figure 9 - 83** shows the waveforms during operation (condition 6a) when TKBOn1 is output on detecting INTP1, with maximum frequency limiting in use to avoid the state in which the judgment cycle length becomes less than $T/2$ of the previous cycle.

Figure 9 - 83 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition 6a: INTP0 Is Detected at Less Than $T/2$ of the Previous Cycle and Maximum Frequency Limiting Starts [TKBIRSn[2:0] = 0xxB]



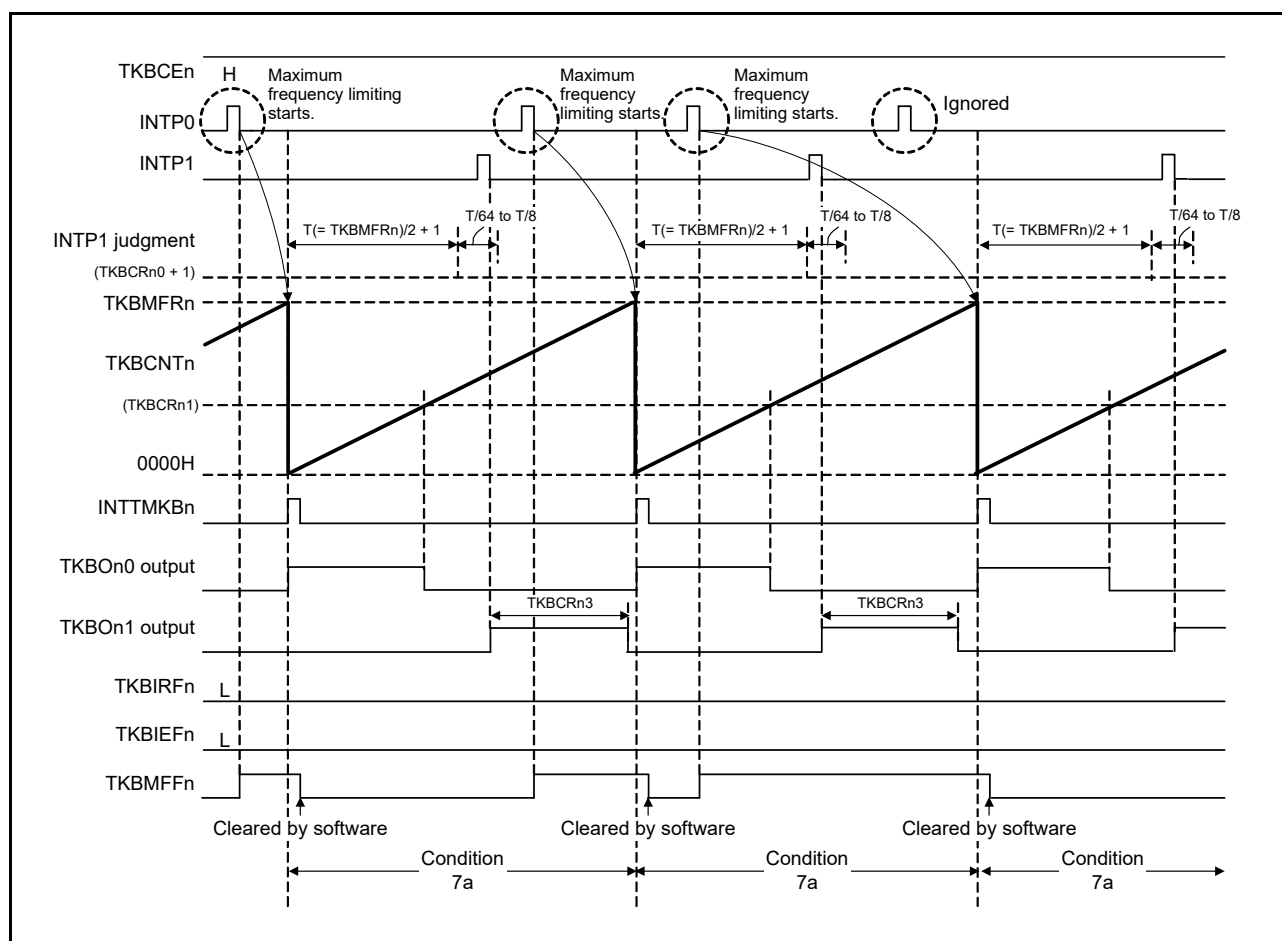
Condition 6a: When INTP1 is detected within $T/2$ of the previous cycle, a TKBOn1 pulse with the width set in the TKBCRn3 register is output at $T/2$ of the previous cycle. Starting of maximum frequency limiting makes the period be controlled by a match between the counter value and the TKBMFRn register. This changes the operation under condition 6b to that under condition 6a, and TKBOn1 is output.

4. Operation in the interleaved PFC output mode (TKBIRS_n[2:0] = 0xxB)

Using maximum frequency limiting leads to an assumption that INTP1 is detected during the counting operation starting from the detection of INTP0 and lasting until the value set by the TKBMFR_n register is reached. When TKBIRS_n[2:0] = 0xxB, the output control of TKBO_n1 depends on the timing when INTP1 is detected. Thus, the order of clearing the counter by INTP0 and the detection of INTP1 may differ. **Figure 9 - 84** shows the waveforms during operation when maximum frequency limiting starts on detection of INTP0 at less than T/2 of the previous cycle.

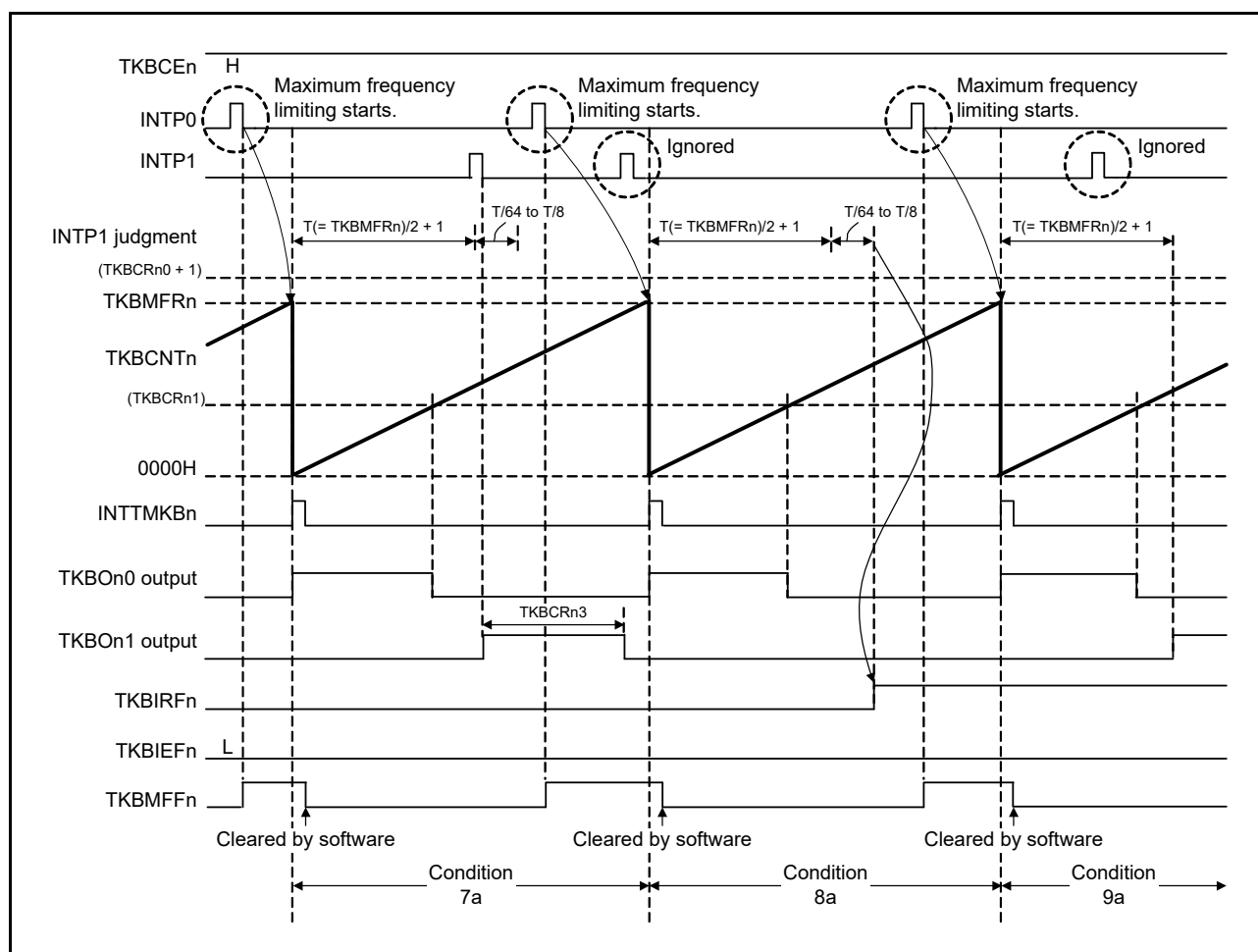
Figure 9 - 85 shows the waveforms during operation when maximum frequency limiting starts on detection of INTP0 at no less than T/2 of the previous cycle.

Figure 9 - 84 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition 7a: INTP0 Is Detected at Less Than T/2 of the Previous Cycle and Maximum Frequency Limiting Starts



Condition 7a: INTP1 is detected within the range from T/2 to T/2 + T/m (m: 8, 16, 32, or 64; set by the TKBIRS_n[1:0] bits) of the previous cycle, and a TKBO_n1 pulse with the width set in the TKBCR_n3 register is output.

Figure 9 - 85 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition 8a: INTP0 Is Detected at No Less Than $T/2 + T/m$ (m : 8, 16, 32, or 64; Set by the $TKBIRSn[1:0]$ Bits) of the Previous Cycle and Maximum Frequency Limiting Starts



Condition 7a: INTP1 is detected within the range from $T/2$ to $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $TKBIRSn[1:0]$ bits) of the previous cycle, and a $TKBOn1$ pulse with the width set in the $TKBCRn3$ register is output. Detection of the second INTP1 in the cycle with maximum frequency limiting in use is ignored.

Condition 8a: INTP1 is not detected within $T/2 + T/m$ (m : 8, 16, 32, or 64; set by the $TKBIRSn[1:0]$ bits) of the previous cycle, and the $TKBOn1$ output is retained. The $TKBIRFn$ flag is set to 1 at this time.

5. Operation in the interleaved PFC output mode (TKBIRSn[2:0] = 100B or 101B)

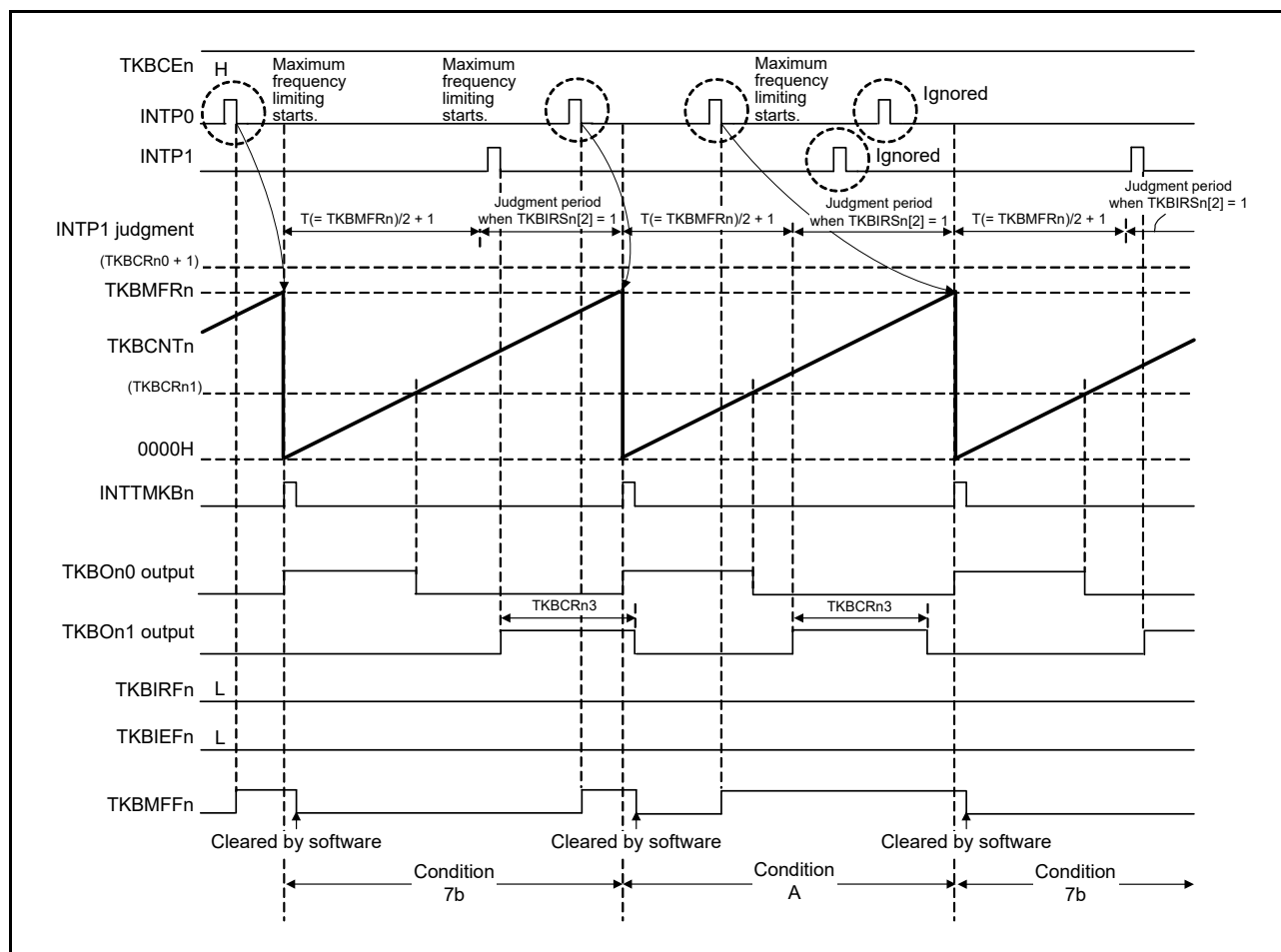
For using maximum frequency limiting when the TKBIRSn[2:0] bits are set to 100B or 101B, regardless of whether INTP1 has been detected, TKBOn1 is output at the timing of INTP0 detection, which leads to starting of maximum frequency limiting.

The TKBOn1 output conditions with maximum frequency limiting in use are shown in the following table. For the TKBOn1 output conditions with maximum frequency limiting not in use, see **9.4.10 Interleaved power factor correction (PFC) output mode**.

Condition No.	Judgment 1 (INTP0 Input)	Judgment 2 (Matching with TKBCRn0/INTP1 Input)	Judgment 3 (Cycle Width)	Output
A	INTP0 input is detected at no greater than the TKBMFRn setting and less than T/2	—	—	When TKBIRSn[2:0] = 100B: Output at T/2
B				When TKBIRSn[2:0] = 101B: Output at trigger input
C	INTP0 input is detected at no greater than the TKBMFRn setting and no less than T/2	—	—	Output at trigger input

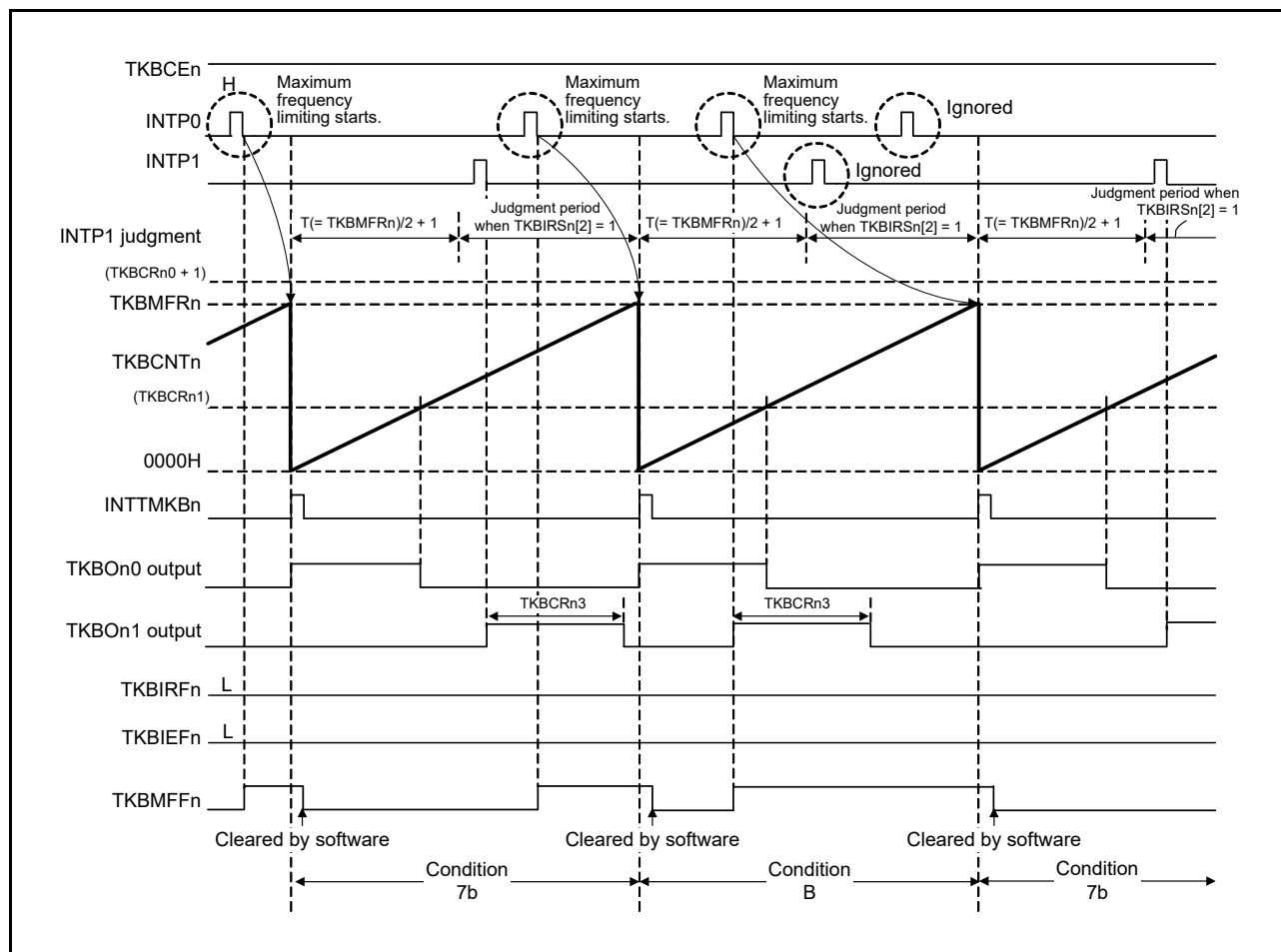
With respect to the operation when maximum frequency limiting starts on detection of INTP0 at less than $T/2$ of the previous cycle as shown in **Figure 9 - 84**, **Figure 9 - 86** shows the waveforms during operation when $TKBIRSn[2:0] = 100B$ and **Figure 9 - 87** shows the waveforms during operation when $TKBIRSn[2:0] = 101B$.

Figure 9 - 86 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition A: INTP0 Is Detected at Less Than $T/2$ of the Previous Cycle and Maximum Frequency Limiting Starts [TKBIRSn[2:0] = 100B]



Condition A: Maximum frequency limiting starts when INTP0 is detected at less than $T/2$ of the previous cycle because the counter value on detecting INTP0 is smaller than the $TKBMFRn$ setting. $TKBO1$ is output when the counter value becomes $T/2$ of the previous cycle. After maximum frequency limiting has started, INTP1 detection is ignored.

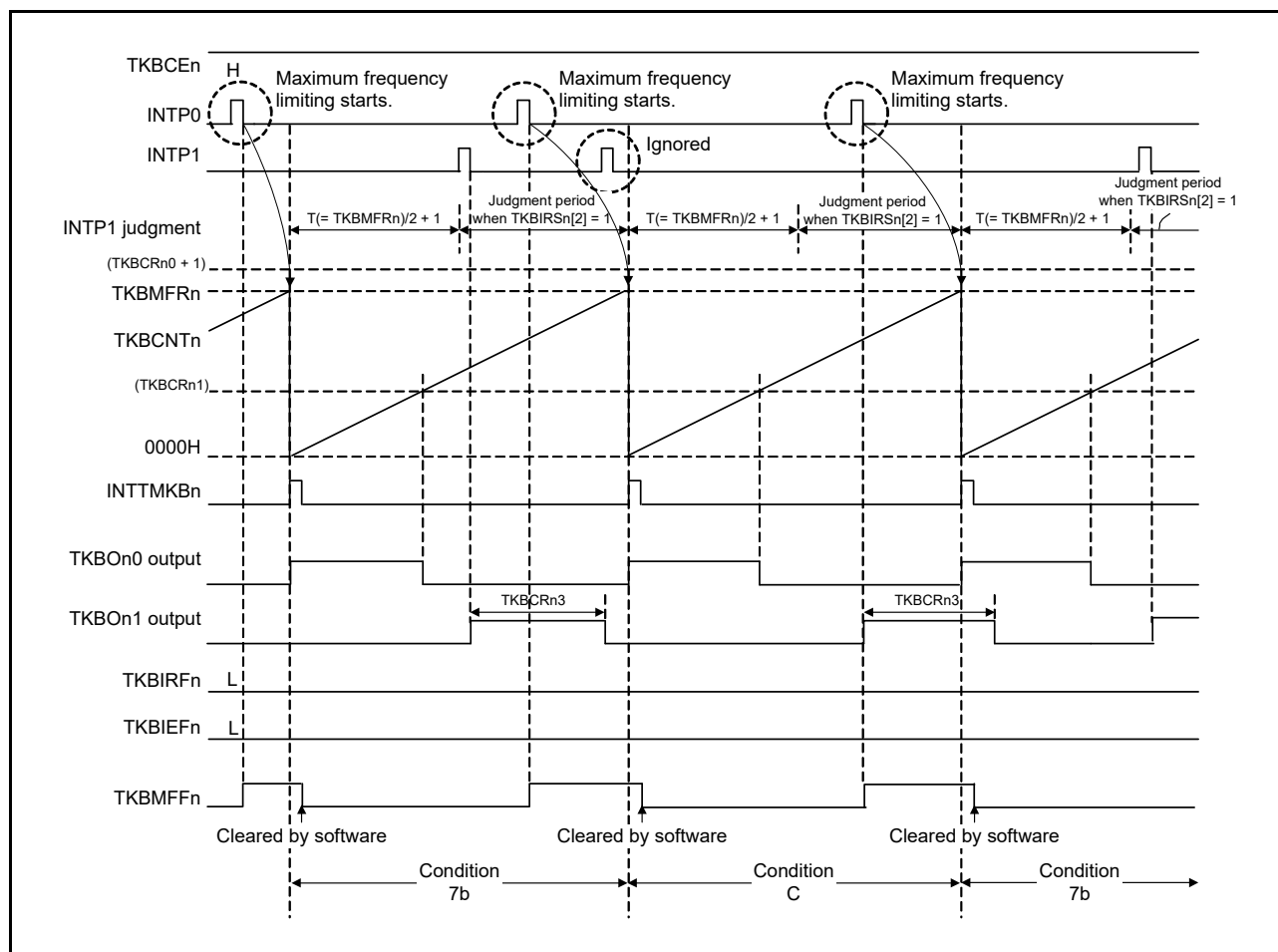
Figure 9 - 87 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition B: INTP0 Is Detected at Less Than $T/2$ of the Previous Cycle and Maximum Frequency Limiting Starts [TKBIRS_n[2:0] = 101B]



Condition B: Maximum frequency limiting starts when INTP0 is detected at less than $T/2$ of the previous cycle because the counter value on detecting INTP0 is smaller than the TKBMFR_n setting. TKBOn1 is output when INTP0 is detected. After maximum frequency limiting has started, INTP1 detection is ignored.

With respect to the operation when maximum frequency limiting starts on detection of INTP0 at no less than $T/2$ of the previous cycle as shown in **Figure 9 - 85**, **Figure 9 - 88** shows the waveforms during operation when $TKBIRSn[2:0] = 100B$ or $101B$.

Figure 9 - 88 Timing with Maximum Frequency Limiting in Use in the Interleaved PFC Output Mode under Condition C: INTP0 Is Detected at No Less Than $T/2$ of the Previous Cycle and Maximum Frequency Limiting Starts [TKBIRSn[2:0] = 100B or 101B]



Condition C: Maximum frequency limiting starts when INTP0 is detected at no less than $T/2$ of the previous cycle because the counter value on detecting INTP0 is smaller than the $TKBMFRn$ setting. $TKBOn1$ is output on detecting INTP0 because the counter value is no less than $T/2$ of the previous cycle.

9.5.5 Masking of external trigger inputs while the TKBOnp output is active

The timer has a function for suppressing the detection of essentially unnecessary external trigger inputs in response to changes in the active level of the TKBOnp output. The timer output to be used in judgment for this purpose can be selected by using the TKBTGMnp bit for controlling the masking of external trigger inputs. In masking control for the masking of external trigger inputs, the active level of the 16-bit timer KB4n output prior to the application of forced output stopping to TKBOnp is used in the judgment. When maximum frequency limiting is in use (TKBMFEn = 1), whether maximum frequency limiting has started in the cycle is judged for external trigger inputs that are detected after masking control has been applied.

Figure 9 - 89 Timing with Masking of External Trigger Inputs While the TKBOnp Output Is Active in Use [TKBTGMn1 = 0, TKBTGMn0 = 1, TKBTOLnp = 0]

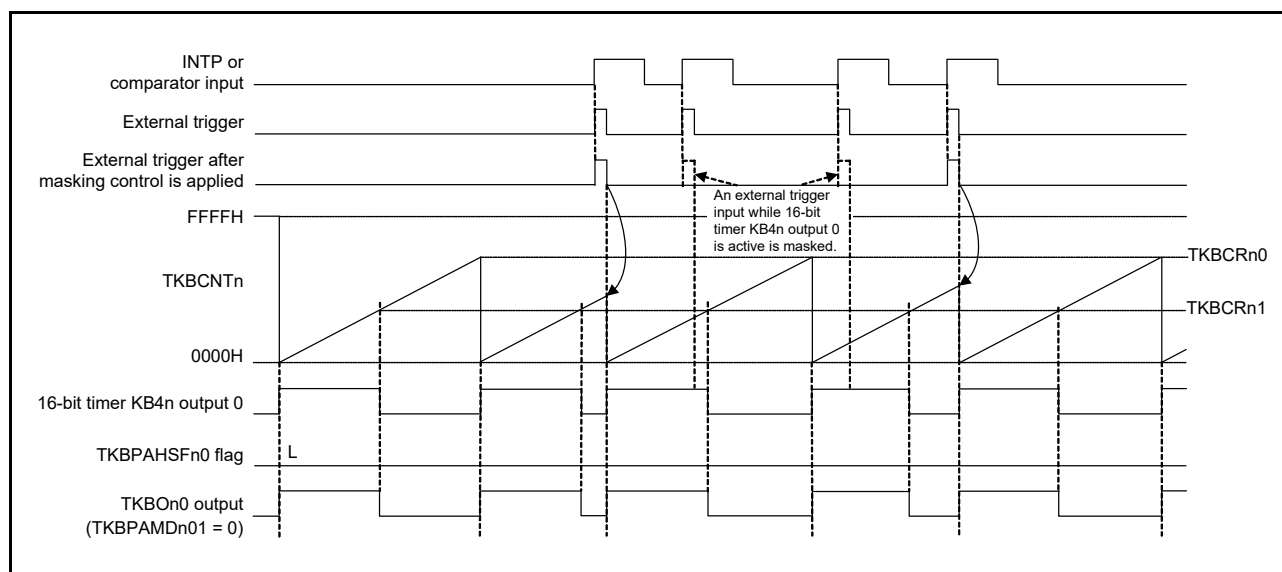
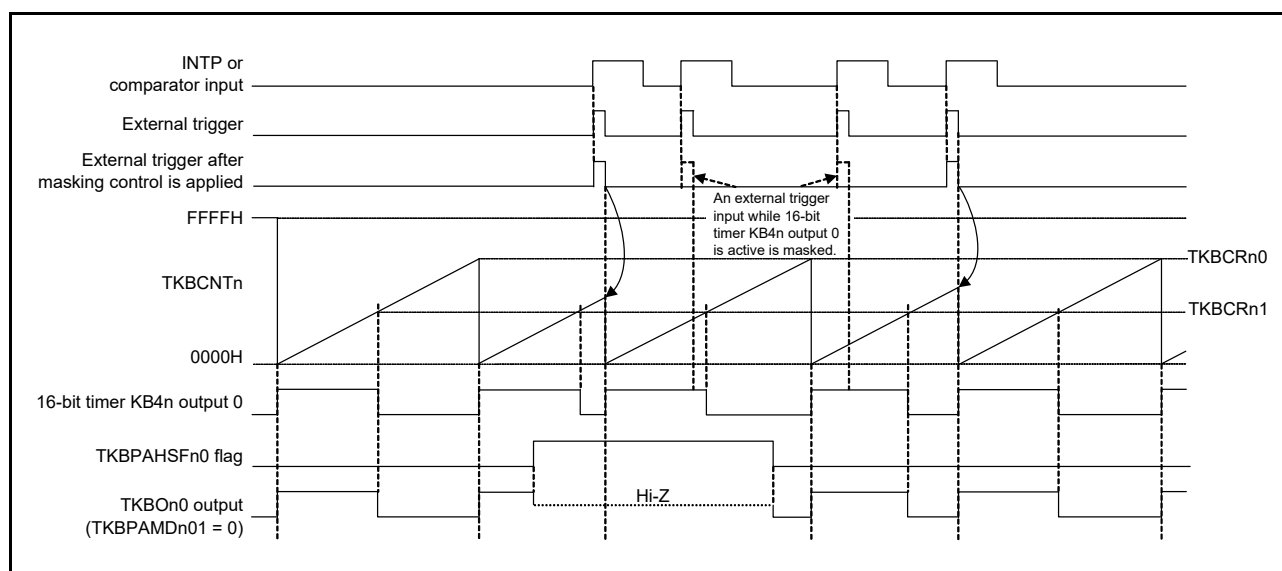


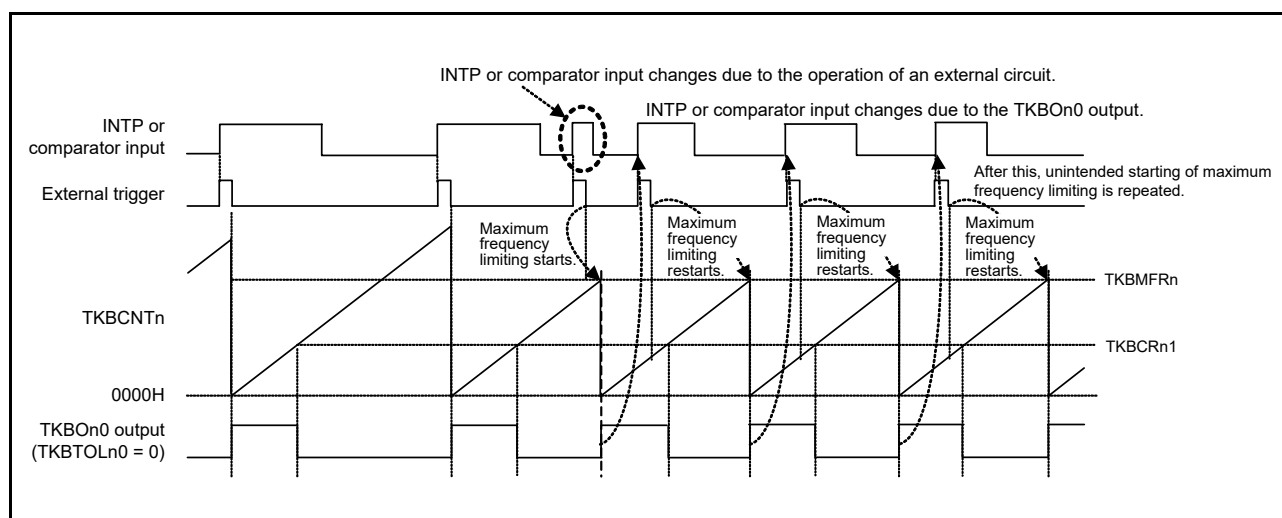
Figure 9 - 90 Timing with Masking of External Trigger Inputs While the TKBOnp Output Is Active in Use, and with the Timer Output Being in the Hi-Z State Due to Forced Output Stopping [TKBTGMn1 = 0, TKBTGMn0 = 1, TKBTOLnp = 0]



As examples of the application of masking of external trigger inputs while the TKBOnp output is active, **Figure 9 - 91** shows an example of operation when maximum frequency limiting starts consecutively due to the operation of an external circuit and **Figure 9 - 92** shows an example of operation when masking control with maximum frequency limiting is used to suppress the consecutive starting of maximum frequency limiting.

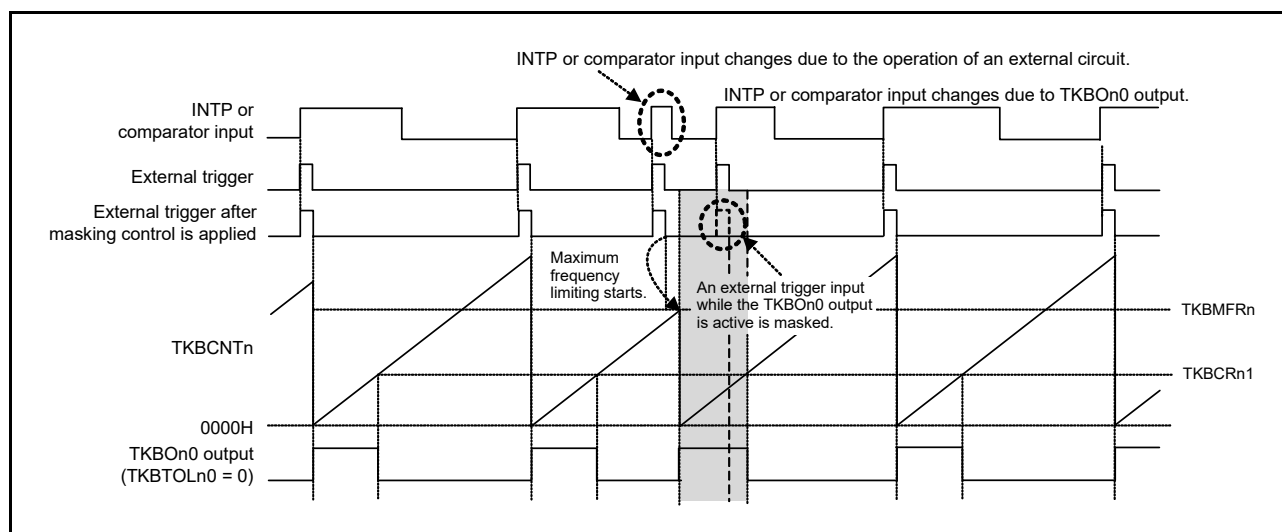
When an external trigger input is generated due to the operation of an external circuit, maximum frequency limiting starts, and the values of the TKBCNTn counter and maximum frequency limit setting register n (TKBMFRn) match, the TKBCNTn counter is cleared and the TKBOn0 output becomes active. After that, external trigger inputs repeatedly generated at changes on the TKBOn0 output may lead to continuing operation with the setting of the maximum frequency limit setting register n (TKBMFRn) used as the cycle.

Figure 9 - 91 Example of Operation When Maximum Frequency Limiting Starts Consecutively Due to the Operation of an External Circuit



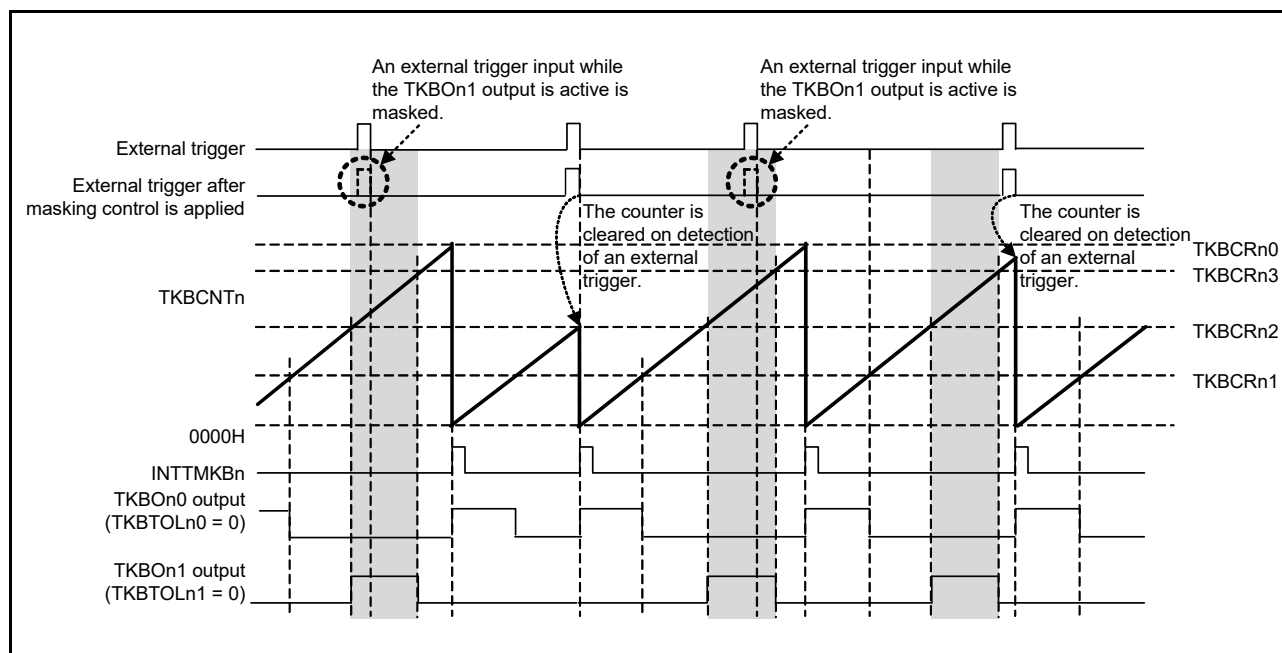
Setting the TKBTGMn0 bit to 1 masks an external trigger input while the TKBOn0 output is at the active level, which enables suppression of the consecutive starting of maximum frequency limiting as shown in **Figure 9 - 91**.

Figure 9 - 92 Example of Operation When Masking Control with Maximum Frequency Limiting Being Applied (TKBTGMn0 = 1) Is Used to Suppress the Consecutive Starting of Maximum Frequency Limiting



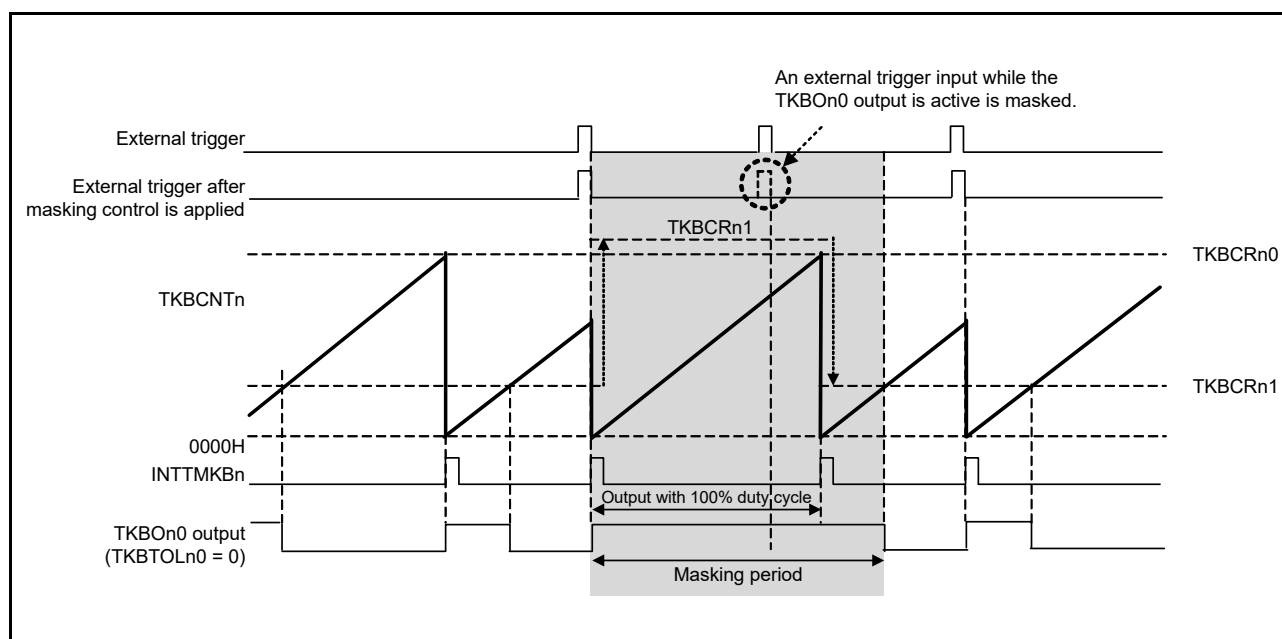
Setting the TKBTGMn1 bit to 1 enables masking of an external trigger input while the TKBOn1 output is at the active level. **Figure 9 - 93** shows an example of operation to mask an external trigger input while the TKBOn1 output is active.

Figure 9 - 93 Example of Operation to Mask an External Trigger Input While the TKBOn1 Output Is Active



When the duty cycle for the TKBOn0 output is changed to 100% during operation, the PWM cycle period is used as the masking period, and therefore the TKBCNTn counter is not cleared until the value of the TKBCNTn counter and the setting of the TKBCRn0 register match. **Figure 9 - 94** shows an example of operation under masking control for the TKBOn0 output with a duty cycle of 100%.

Figure 9 - 94 Example of Operation under Masking Control for the TKBOn0 Output with a Duty Cycle of 100% When TKBTGMn0 = 1



9.5.6 PWM output for IH control

The timer supports PWM output for inductive heating (IH) control that guarantees the width of the period over which the TKBOn1 output is at the active level. TKBOn1 is only used for PWM output for IH control. The default level specified by the TKBTODn0 bit is used for the TKBOn0 output. After the counter (TKBCNTn) starts operating, the PWM output starts when the count period specified by the TKBCRn2 register has passed, and the TKBOn1 output remains active during the count period specified by the TKBCRn3 register. The input signal selected by the TKBSTSn[1:0] and TKBINSn[2:0] bits is used as the external trigger input while PWM output for IH control is in use (TKBIHEn = 1). On detection of an external trigger input, the counter (TKBCNTn) is cleared to 0000H. The TKBTSE bit is used to enable or disable batch overwriting of the compare registers triggered by an external trigger input. Note that the period when the counter value is from 0000H to the TKBCRn3 setting is defined as the active period, and the detection of external trigger inputs is not accepted and disabled in this period. This is for ensuring the width of the period over which the TKBOn1 output is at the active level. In turn, the period when the counter value is from the TKBCRn3 setting to the TKBCRn0 setting is defined as the non-active period and the detection of external trigger inputs is accepted in this period.

[Formulas for TKBOn1 output under period control by the TKBCRn0 register]

Pulse cycle = (TKBCR0 setting + 1) × counter clock period

Duty cycle [%] = ((TKBCRn3 setting – TKBCRn2 setting)/(TKBCRn0 setting + 1)) × 100

Output with 0% duty cycle: TKBCRn3 setting = TKBCRn2 setting

Output with 100% duty cycle: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution Make sure that the TKBCRn2 setting is no greater than the TKBCRn3 setting.

[Formulas for TKBOn1 output under period control by external trigger input]

Pulse cycle = (counter value on detection of external trigger input + 1) × counter clock cycle

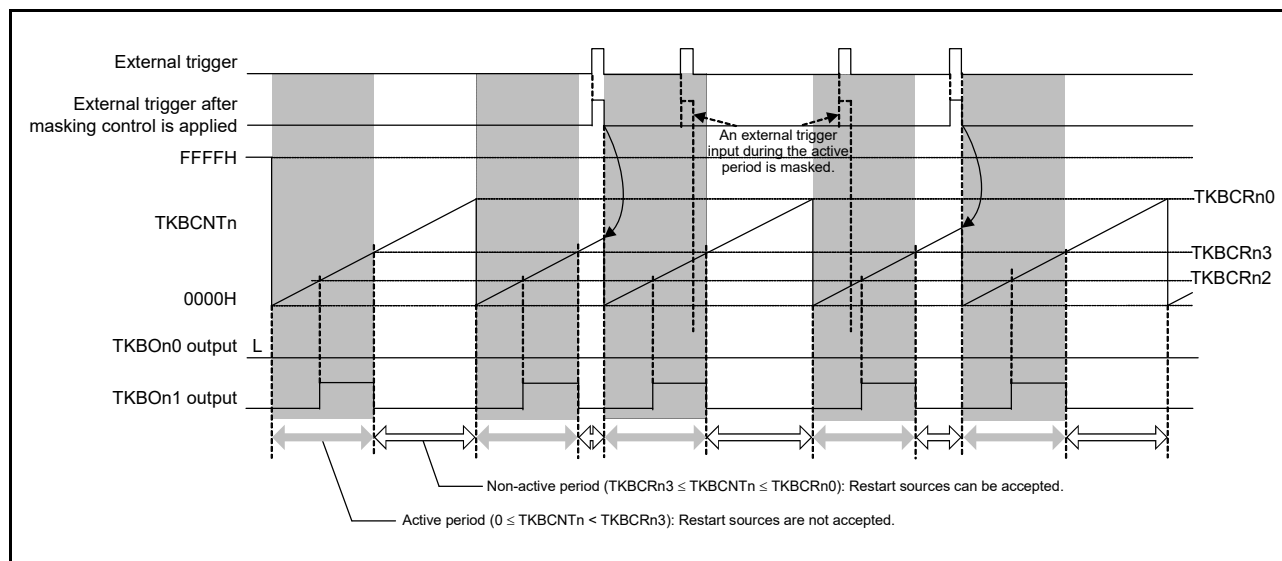
Duty cycle [%] = ((TKBCRn3 setting – TKBCRn2 setting)/(counter value on detection of external trigger input + 1)) × 100

Output with 0% duty cycle: TKBCRn3 setting = TKBCRn2 setting

Output with 100% duty cycle: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ counter value on detection of external trigger input + 1

Caution Make sure that the TKBCRn2 setting is no greater than the TKBCRn3 setting.

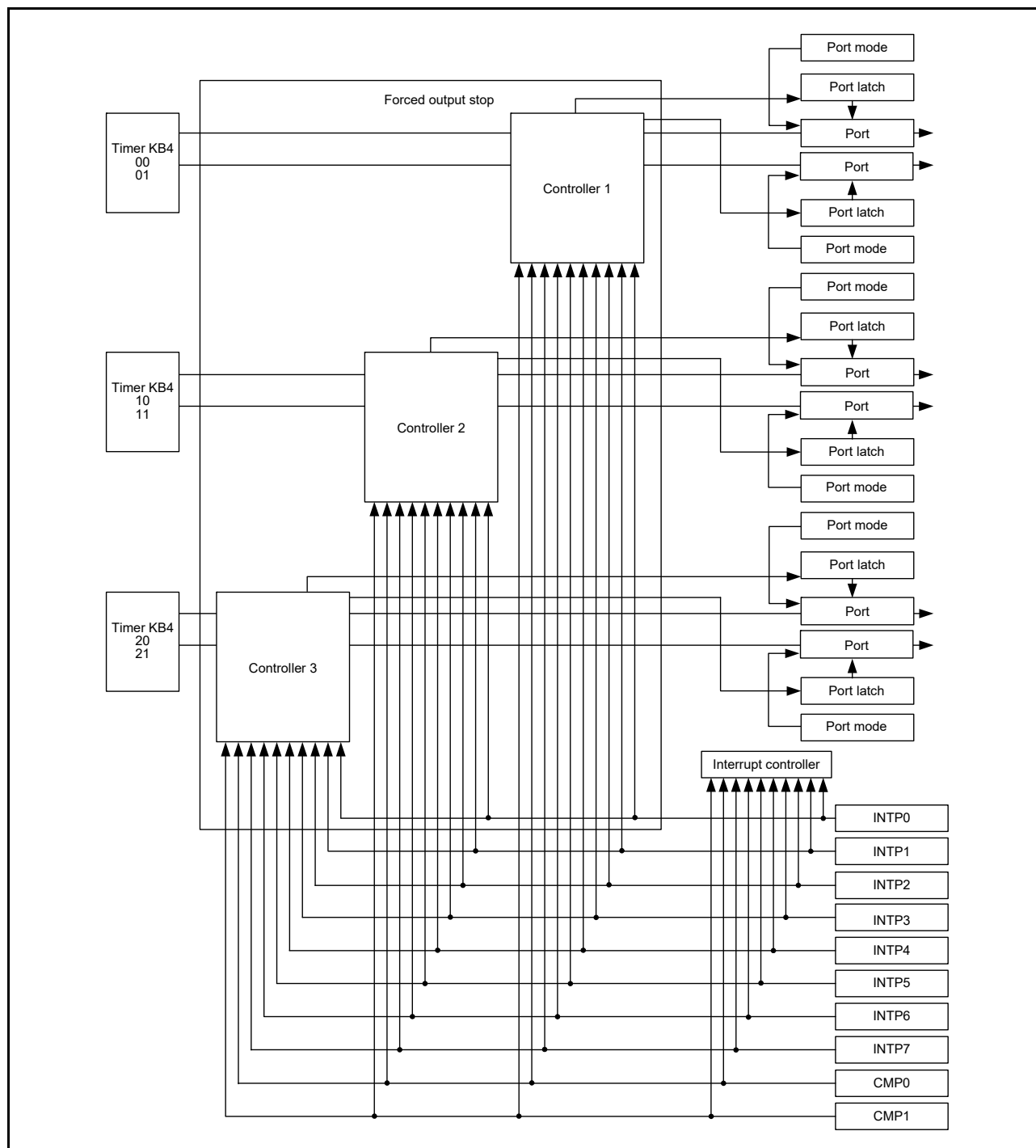
Figure 9 - 95 Waveforms during Operation with PWM Output for IH Control in Use When the Default Level of Output Is Low (TKBTODn1 = 0) and the Active Level Is High (TKBTOLn1 = 0)



9.6 Forced Output Stop

Forced output stop is used to protect power-supply circuits, etc. If any abnormal situation, such as a short circuit, that occurs in a power-supply circuit configured external to the microcontroller leads to an overvoltage or overcurrent, inputting a voltage or current sense signal through an external interrupt or to a comparator can protect the circuit by maintaining the timer output in the Hi-Z state or fixed-output state without program control by the CPU. This function starts or is canceled when an edge of the specified input signal is detected. **Figure 9 - 96** shows the system structure for forced output stop.

Figure 9 - 96 System Structure for Forced Output Stop



9.6.1 Forced output stop 1 and 2

Two methods for controlling forced output stop are supported. Forced output stop 1 is used to select fixed level output or the Hi-Z state, and forced output stop 2 is only used to set fixed level output. The following tables show the differences in control between the functions.

1. Selectable output levels for forced output stop 1 and 2

Selectable Output Levels	Forced Output Stop	
	1	2
Hi-Z	✓	×
Fixed to the low level	✓	✓
Fixed to the high level	✓	✓

2. Start or cancellation of forced output stop 1 and 2

Details of the Function and Operation (Start of Forced Output Stop)	Forced Output Stop	
	1	2
Forced output stop starts following the detection of the rising edge of a comparator output.	✓	✓
Forced output stop starts following the detection of the rising or falling edge of an external interruption input.	✓	✓
Forced output stop starts by setting the TKBPAHTSnP bit.	✓	×

Details of the Function and Operation (Cancellation of Forced Output Stop)	Forced Output Stop	
	1	2
Cancellation of forced output stop proceeds by setting the TKBPAHTTnP bit.	✓	×
Cancellation of forced output stop proceeds by synchronization with a cycle of 16-bit timer KB40, KB41, or KB42 after the TKBPAHTTnP bit is set.	✓	×
Cancellation of forced output stop proceeds at the next counter cycle after starting of forced output stop.	×	✓
Cancellation of forced output stop proceeds at the next counter cycle after the detection of a falling edge of the trigger signal for forced output stop.	×	✓
Cancellation of forced output stopping proceeds following synchronization with fKBK after the detection of a falling edge of the trigger signal for forced output stopping (fixed-off function).	×	✓
Cancellation of forced output stop proceeds by setting the TKBPAFTTnP bit.	×	✓

3. Selectable trigger signals and available trigger bits for forced output stop 1 and 2

Selectable Trigger Signals	Forced Output Stop	
	1	2
Comparators 0, 1	✓	✓
External interrupt inputs (INTP0 to INTP7)	✓	✓

(Remark is listed on the next page.)

Available Trigger Bits	Forced Output Stop	
	1	2
TKBPAHTSn _p (the trigger bit for starting forced output stop on the TKBO _n p output)	✓	×
TKBPAHTTn _p (the trigger bit for canceling forced output stop on the TKBO _n p output)	✓	×
TKBPAFTTn _p (the trigger bit for canceling forced output stop on the TKBO _n p output)	×	✓

Remark n = 0 to 2; p = 0, 1

Table 9 - 5 External Trigger Assignment List for Forced Output Stop 1

Item	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
INTP0	✓	✓	✓	✓	✓	✓
INTP1	✓	✓	✓	✓	✓	✓
INTP2	✓	✓	✓	✓	✓	✓
INTP3	✓	✓	✓	✓	✓	✓
INTP4	✓	✓	✓	✓	✓	✓
INTP5	✓	✓	✓	✓	✓	✓
INTP6	✓	✓	✓	✓	✓	✓
INTP7	✓	✓	✓	✓	✓	✓
Comparator 0	✓	✓	✓	✓	✓	✓
Comparator 1	✓	✓	✓	✓	✓	✓

Table 9 - 6 External Trigger Assignment List for Forced Output Stop 2

Item	TKBO00	TKBO01	TKBO10	TKBO11	TKBO20	TKBO21
INTP0	✓	✓	✓	✓	✓	✓
INTP1	✓	✓	✓	✓	✓	✓
INTP2	✓	✓	✓	✓	✓	✓
INTP3	✓	✓	✓	✓	✓	✓
INTP4	✓	✓	✓	✓	✓	✓
INTP5	✓	✓	✓	✓	✓	✓
INTP6	✓	✓	✓	✓	✓	✓
INTP7	✓	✓	✓	✓	✓	✓
Comparator 0	✓	✓	✓	✓	✓	✓
Comparator 1	✓	✓	✓	✓	✓	✓

Caution 1. Regarding the settings for INTP0 to INTP7, see Section 25 Interrupt Functions and 9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7).

Caution 2. Regarding the settings for the output signals from comparators 0 and 1, see Section 18 Comparator (CMP).

9.6.2 Configuration of hardware for forced output stop

Forced output stop is configured by the following hardware.

Table 9 - 7 Configuration of Hardware for Forced Output Stop

Item	Configuration
Control registers	• Forced output stop function control register np (TKBPACTLn_p) (n = 0 to 2; p = 0, 1) • Forced output stop function control register n2 (TKBPACTLn₂) (n = 0 to 2) • Forced output stop function 1 input select register np (TKBPAHZSn_p) (n = 0 to 2; p = 0, 1) • Forced output stop function 2 input select register np (TKBPAFXSn_p) (n = 0 to 2; p = 0, 1) • Forced output stop function flag register n (TKBPAFLG_n) (n = 0 to 2) • Forced output stop function start trigger register n (TKBPAHFS_n) (n = 0 to 2) • Forced output stop function cancel trigger register n (TKBPAHFT_n) (n = 0 to 2)

9.6.3 Registers for controlling forced output stop

The following registers are used to control forced output stop.

- Forced output stop function control register np (TKBPACTLnp) (n = 0 to 2; p = 0, 1)
- Forced output stop function control register n2 (TKBPACTLn2) (n = 0 to 2)
- Forced output stop function 1 input select register np (TKBPAHZSnp) (n = 0 to 2; p = 0, 1)
- Forced output stop function 2 input select register np (TKBPAFXSnp) (n = 0 to 2, p = 0, 1)
- Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2)
- Forced output stop function start trigger register n (TKBPAHFSn) (n = 0 to 2)
- Forced output stop function cancel trigger register n (TKBPAHFTn) (n = 0 to 2)

9.6.3.1 Forced output stop function control register np (TKBPACTLnp) (n = 0 to 2; p = 0, 1)

The TKBPACTLnp register is used to specify the mode for controlling forced output stop on the TKBOnp pin. The TKBPACTLnp register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 97 Format of Forced Output Stop Function Control Register np (TKBPACTLnp) (1/2)

Address: F0770H (TKBPACTL00), F0772H (TKBPACTL01), F07B0H (TKBPACTL10), F07B2H (TKBPACTL11),
F04D0H (TKBPACTL20), F04D2H (TKBPACTL21)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTLnp	0	0	0	0	0	0	TKBPAFIXnp	TKBPAFCMnp
	7	6	5	4	3	2	1	0
	0	0	0	0	TKBPAHCMnp1	TKBPAHCMnp0	TKBPAMDnp1	TKBPAMDnp0
TKBPAFIXnp	Fixed-off setting							
0	Being fixed off is disabled. The operation after forced stop 2 starts depends on the setting of the TKBPA0FCMn bit.							
1	Being fixed off is enabled (only valid when TKBPA0FCMn = 1). Forced output stop 2 starts when forced output stop input 2 is detected and the function is canceled when the reverse edge of the trigger is detected.							
TKBPAFCMnp	Operating mode selection for forced output stop 2							
0	Forced output stop 2 starts with a trigger input, and forced output stop 2 is canceled at the next counter period ^{Note} .							
1	Forced output stop 2 starts with a trigger input, and forced output stop 2 is canceled at the next counter period following detection of the reverse edge of the trigger ^{Note} .							

Figure 9 - 97 Format of Forced Output Stop Function Control Register np (TKBPACTLnp) (2/2)

TKBPAHC Mnp1	TKBPAHC Mnp0	Cancellation condition selection for forced output stop 1	
0	0	Forced output stop 1 starts in response to a trigger input, and forced output stop 1 is canceled on writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp), regardless of the input trigger's signal level.	
0	1	Forced output stop 1 starts in response to a trigger input. After that, while the trigger signal is at the high level, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) is disabled. On the other hand, while the input trigger signal is at the low level, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) cancels forced output stop 1.	
1	0	Forced output stop 1 starts in response to a trigger input. After that, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) cancels forced output stop 1 on the next cycle of the counter, regardless of the level of the input trigger signal ^{Note} .	
1	1	Forced output stop 1 starts in response to a trigger input, and while the trigger signal is in its high-level period, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) is ineffective. Forced output stop 1 is canceled on the next counter cycle after writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) when the trigger signal is in its low-level period ^{Note} .	

TKBPAMD np1	TKBPAMD np0	Output state selection on implementation of forced output stop	
		Forced output stop 1	Forced output stop 2
0	0	Hi-Z	Fixed to the low level
0	1	Hi-Z	Fixed to the high level
1	0	Fixed to the low level	Fixed to the low level
1	1	Fixed to the high level	Fixed to the high level

Note When a 16-bit timer KB4n is stopped (TKBCEn = 0) without waiting for the next counter cycle, forced output stopping remains in force until the 16-bit timer KB4n is restarted (TKBCEn = 1).

Caution 1. Overwriting the TKBPACTLnp register during timer operation is prohibited. Note, however, that the register can be refreshed (by writing the existing value) even in this case.

Caution 2. Be sure to clear bits 15 to 10 and 7 to 4 to 0.

Remark n = 0 to 2; p = 0, 1

9.6.3.2 Forced output stop function control register n2 (TKBPACTLn2) (n = 0 to 2)

The TKBPACTLn2 register is used to enable or disable forced output stop on the TKBOnp pin. The TKBPACTLn2 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 9 - 98 Format of Forced Output Stop Function Control Register n2 (TKBPACTLn2)

Address: F0777H (TKBPACTL02), F07B7H (TKBPACTL12), F04D7H (TKBPACTL22)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPACTLn ₂	0	0	0	0	0	0	TKBPACEn ₁	TKBPACEn ₀

TKBPACEn _p	Input control of trigger signal used for forced output stop on the TKBOnp pin
0	Disables forced output stop.
1	Enables forced output stop.

Caution 1. The TKBPACTLn2 register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to 0.

Remark n = 0 to 2; p = 0, 1

9.6.3.3 Forced output stop function 1 input select register np (TKBPAHZSn) (n = 0 to 2; p = 0, 1)

The TKBPAHZSn register is used to select the forced output stop input for forced output stop 1 on the TKBOnp pin. The TKBPAHZSn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 99 Format of Forced Output Stop Function 1 Input Select Register np (TKBPAHZSn) (1/2)

Address: F0778H (TKBPAHZS00), F077AH (TKBPAHZS01), F07B8H (TKBPAHZS10), F07BAH (TKBPAHZS11), F04D8H (TKBPAHZS20), F04DAH (TKBPAHZS21)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPAHZSn p	0	0	0	0	0	0	TKBPAHZSn p9	TKBPAHZSn p8
	7	6	5	4	3	2	1	0
TKBPAHZSn p7	TKBPAHZSn p6	TKBPAHZSn p5	TKBPAHZSn p4	TKBPAHZSn p3	TKBPAHZSn p2	TKBPAHZSn p1	TKBPAHZSn p0	
TKBPAHZS np9	External interrupt trigger selection for forced output stop 1							
0	Comparator 1 is not used as a trigger.							
1	Comparator 1 is used as a trigger ^{Note 1} .							
TKBPAHZS np8	External interrupt trigger selection for forced output stop 1							
0	Comparator 0 is not used as a trigger.							
1	Comparator 0 is used as a trigger ^{Note 1} .							
TKBPAHZS np7	External interrupt trigger selection for forced output stop 1							
0	INTP7 is not used as a trigger.							
1	INTP7 is used as a trigger ^{Note 2} .							
TKBPAHZS np6	External interrupt trigger selection for forced output stop 1							
0	INTP6 is not used as a trigger.							
1	INTP6 is used as a trigger ^{Note 2} .							
TKBPAHZS np5	External interrupt trigger selection for forced output stop 1							
0	INTP5 is not used as a trigger.							
1	INTP5 is used as a trigger ^{Note 2} .							

Figure 9 - 99 Format of Forced Output Stop Function 1 Input Select Register np (TKBPAHZSnp) (2/2)

TKBPAHZS np4	External interrupt trigger selection for forced output stop 1
0	INTP4 is not used as a trigger.
1	INTP4 is used as a trigger ^{Note 2} .

TKBPAHZS np3	External interrupt trigger selection for forced output stop 1
0	INTP3 is not used as a trigger.
1	INTP3 is used as a trigger ^{Note 2} .

TKBPAHZS np2	External interrupt trigger selection for forced output stop 1
0	INTP2 is not used as a trigger.
1	INTP2 is used as a trigger ^{Note 2} .

TKBPAHZS np1	External interrupt trigger selection for forced output stop 1
0	INTP1 is not used as a trigger.
1	INTP1 is used as a trigger ^{Note 2} .

TKBPAHZS np0	External interrupt trigger selection for forced output stop 1
0	INTP0 is not used as a trigger.
1	INTP0 is used as a trigger ^{Note 2} .

Note 1. For using comparator 0 as an external trigger for forced output stop 1, set the C0FCK1 and C0FCK0 bits to 00B (no filter) and the C0EDG bit to 0 (detection of one edge). For using comparator 1, set the C1FCK1 and C1FCK0 bits to 00B (no filter) and the C1EDG bit to 0 (detection of one edge). For details, see **18.3.4 Comparator filter control register (COMPFIR)**.

Note 2. For using INTPx as an external trigger for forced output stop 1, set the INTFCKx1 and INTFCKx0 bits to 00B (no filter). For details, see **9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7)**.

Caution 1. Overwriting the TKBPAHZSnp register during timer operation is prohibited.

Caution 2. Be sure to clear bits 15 to 10 to 0.

Remark n = 0 to 2; p = 0, 1; x = 0 to 7

9.6.3.4 Forced output stop function 2 input select register np (TKBPAFXSn) (n = 0 to 2, p = 0, 1)

The TKBPAFXSn register is used to select the forced output stop input for forced output stop 2 on the TKBOnp pin. The TKBPAFXSn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 9 - 100 Format of Forced Output Stop Function 2 Input Select Register np (TKBPAFXSn) (1/2)

Address: F077CH (TKBPAFXS00), F077EH (TKBPAFXS01), F07BCH (TKBPAFXS10), F07BEH (TKBPAFXS11), F04DCH (TKBPAFXS20), F04DEH (TKBPAFXS21)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TKBPAFXSn p	0	0	0	0	0	0	TKBPAFXSn p9	TKBPAFXSn p8
	7	6	5	4	3	2	1	0
TKBPAFXSn p7	TKBPAFXSn p6	TKBPAFXSn p5	TKBPAFXSn p4	TKBPAFXSn p3	TKBPAFXSn p2	TKBPAFXSn p1	TKBPAFXSn p0	
TKBPAFXS np9	External interrupt trigger selection for forced output stop 2							
0	Comparator 1 is not used as a trigger.							
1	Comparator 1 is used as a trigger ^{Note 1} .							
TKBPAFXS np8	External interrupt trigger selection for forced output stop 2							
0	Comparator 0 is not used as a trigger.							
1	Comparator 0 is used as a trigger ^{Note 1} .							
TKBPAFXS np7	External interrupt trigger selection for forced output stop 2							
0	INTP7 is not used as a trigger.							
1	INTP7 is used as a trigger ^{Note 2} .							
TKBPAFXS np6	External interrupt trigger selection for forced output stop 2							
0	INTP6 is not used as a trigger.							
1	INTP6 is used as a trigger ^{Note 2} .							
TKBPAFXS np5	External interrupt trigger selection for forced output stop 2							
0	INTP5 is not used as a trigger.							
1	INTP5 is used as a trigger ^{Note 2} .							

Figure 9 - 100 Format of Forced Output Stop Function 2 Input Select Register np (TKBPAFXSn) (2/2)

TKBPAFXS np4	External interrupt trigger selection for forced output stop 2
0	INTP4 is not used as a trigger.
1	INTP4 is used as a trigger ^{Note 2} .

TKBPAFXS np3	External interrupt trigger selection for forced output stop 2
0	INTP3 is not used as a trigger.
1	INTP3 is used as a trigger ^{Note 2} .

TKBPAFXS np2	External interrupt trigger selection for forced output stop 2
0	INTP2 is not used as a trigger.
1	INTP2 is used as a trigger ^{Note 2} .

TKBPAFXS np1	External interrupt trigger selection for forced output stop 2
0	INTP1 is not used as a trigger.
1	INTP1 is used as a trigger ^{Note 2} .

TKBPAFXS np0	External interrupt trigger selection for forced output stop 2
0	INTP0 is not used as a trigger.
1	INTP0 is used as a trigger ^{Note 2} .

Note 1. For using comparator 0 as an external trigger for forced output stop 2, set the C0FCK1 and C0FCK0 bits to 00B (no filter) and the C0EDG bit to 0 (detection of one edge). For using comparator 1, set the C1FCK1 and C1FCK0 bits to 00B (no filter) and the C1EDG bit to 0 (detection of one edge). For details, see **18.3.4 Comparator filter control register (COMPFIR)**.

Note 2. For using INTPx as an external trigger for forced output stop 2, set the INTFCKx1 and INTFCKx0 bits to 00B (no filter). For details, see **9.3.22 External interrupt control register x (INTPCTLx) (x = 0 to 7)**.

Caution 1. Overwriting the TKBPAFXSn register during timer operation is prohibited.

Caution 2. Be sure to clear bits 15 to 10 to 0.

Remark n = 0 to 2; p = 0, 1; x = 0 to 7

9.6.3.5 Forced output stop function flag register n (TKBPAFLGn) (n = 0 to 2)

The TKBPAFLGn register indicates status flags for forced output stop on the TKBOnp pin. The TKBPAFLGn 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 9 - 101 Format of Forced Output Stop Function Flag Register n (TKBPAFLGn)

Address: F0776H (TKBPAFLG0), F07B6H (TKBPAFLG1), F04D6H (TKBPAFLG2)

After reset: 00H

R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBPAFLGn	TKBPAFSFn 1	TKBPAHSFn 1	TKBPAFSFn 0	TKBPAHSFn 0	TKBPAFIFn 1	TKBPAHIFn 1	TKBPAFIFn 0	TKBPAHIFn 0
TKBPAFSFnp								
Status flag for forced output stop 2 on the TKBOnp pin								
0								
Forced output stopping is canceled.								
1								
Forced output stopping is the current state.								
TKBPAHSFnp								
Status flag for forced output stop 1 on the TKBOnp pin								
0								
Forced output stopping is canceled.								
1								
Forced output stopping is the current state.								
TKBPAFIFn								
Input monitor bit for forced output stop 2 on the TKBOnp pin								
0								
Forced output stop 2 trigger signal is at the low level (inactive).								
1								
Forced output stop 2 trigger signal is at the high level (active).								
TKBPAHIFn								
Input monitor bit for forced output stop 1 on the TKBOnp pin								
0								
Forced output stop 1 trigger signal is at the low level (inactive).								
1								
Forced output stop 1 trigger signal is at the high level (active).								

Caution The timing to cancel forced output stop 1 depends on the setting. For details, see 9.7.2 Start and cancellation by software of forced output stop 1.

Remark n = 0 to 2; p = 0, 1

9.6.3.6 Forced output stop function start trigger register n (TKBPAHFSn) (n = 0 to 2)

The TKBPAHFSn register is used to specify the start trigger for forced output stop on the TKBOnp pin. The TKBPAHFSn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 102 Format of Forced Output Stop Function Start Trigger Register n (TKBPAHFSn)

Address: F0774H (TKBPAHFS0), F07B4H (TKBPAHFS1), F04D4H (TKBPAHFS2)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPAHFSn	0	0	0	0	0	0	TKBPAHTSn 1	TKBPAHTSn 0

TKBPAHTS np	Start trigger for forced output stop 1 on the TKBOnp output
0	Invalid setting
1	Forced output stop 1 on the TKBOnp output starts.

Caution 1. The TKBPAHFSn register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to 0.

Caution 3. When TKBPAHFSn register is read, 0 is read.

Remark n = 0 to 2; p = 0, 1

9.6.3.7 Forced output stop function cancel trigger register n (TKBPAHFTn) (n = 0 to 2)

The TKBPAHFTn register is used to specify the cancel trigger for forced output stop on the TKBOnp pin. The TKBPAHFTn register can be written by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 103 Format of Forced Output Stop Function Cancel Trigger Register n (TKBPAHFTn)

Address: F0775H (TKBPAHFT0), F07B5H (TKBPAHFT1), F04D5H (TKBPAHFT2)

After reset: 00H

R/W: R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
TKBPAHFTn	0	0	TKBPAFTTn 1	TKBPAFTTn 0	0	0	TKBPAHTTn 1	TKBPAHTTn 0

TKBPAFTT np	Cancel trigger for forced output stop 2 on the TKBOnp output
0	Invalid setting
1	Forced output stop 2 on the TKBOnp output (fixed-off function) is canceled.

TKBPAHTT np	Cancel trigger for forced output stop 1 on the TKBOnp output
0	Invalid setting
1	Forced output stop 1 on the TKBOnp output is canceled.

Caution 1. The TKBPAHFTn register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7, 6, 3, and 2 to 0.

Caution 3. When TKBPAHFTn register is read, 0 is read.

Caution 4. The timing to cancel forced output stop 1 depends on the setting. For details, see 9.7.2 Start and cancellation by software of forced output stop 1.

Remark n = 0 to 2; p = 0, 1

9.7 Operation with Forced Output Stop 1

Generation of a trigger source (one among the outputs from comparators 0 and 1 and INTP0 to INTP7) enables fixing of timer output to the Hi-Z state or to the low or high level without CPU intervention, and asynchronously with respect to the fKBKC operating clock for the 16-bit timer KB4n. Once the cancel trigger of forced output stop 1 is set, the forced output stop state can be canceled in synchronization with the fKBKC operating clock for the 16-bit timer KB4n.

9.7.1 Summary of forced output stop 1

Comparator output signals, external interrupt signals, and software triggers are used as the trigger signals for forced output stop 1. The output level selectable at forced output stop is controlled by the TKBPAMDnp[1:0] bits of the TKBPACTLnp register. **Table 9 - 8** lists the relationship between the levels of the p output pins (TKBOnp) of the 16-bit timer KB4n and the level output using forced output stop 1.

Table 9 - 8 Relationship between the Levels of the TKBOnp Pins and the Level Output Using Forced Output Stop 1

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection on Using Forced Output Stop 1
0	0	Hi-Z
0	1	Hi-Z
1	0	Fixed to the low level
1	1	Fixed to the high level

The selection of a comparator output or an external interrupt signal to be used is controlled by the TKBPAHZSnp9 to TKBPAHZSnp0 bits of the forced output stop function 1 input select register np (TKBPAHZSnp). **Table 9 - 9** lists trigger selection for forced output stop 1 on the p output pins (TKBOnp) of the 16-bit timer KB4n.

Table 9 - 9 Trigger Selection for Forced Output Stop 1 of the TKBOnp Pins

Bit	Selectable Trigger Signals		
	16-bit Timer KB40	16-bit Timer KB41	16-bit Timer KB42
TKBPAHZSnp0	INTP0	INTP0	INTP0
TKBPAHZSnp1	INTP1	INTP1	INTP1
TKBPAHZSnp2	INTP2	INTP2	INTP2
TKBPAHZSnp3	INTP3	INTP3	INTP3
TKBPAHZSnp4	INTP4	INTP4	INTP4
TKBPAHZSnp5	INTP5	INTP5	INTP5
TKBPAHZSnp6	INTP6	INTP6	INTP6
TKBPAHZSnp7	INTP7	INTP7	INTP7
TKBPAHZSnp8	Comparator 0	Comparator 0	Comparator 0
TKBPAHZSnp9	Comparator 1	Comparator 1	Comparator 1

Remark n = 0 to 2; p = 0, 1

9.7.2 Start and cancellation by software of forced output stop 1

Table 9 - 10 lists the start trigger (the TKBPAHTSn bit of the TKBPAHFSn register) setting for the start of forced output stop 1.

Table 9 - 10 Operation of Start Trigger (the TKBPAHTSn Bit) for Forced Output Stop 1

TKBPAHTSn	Start of Forced Output Stop by Software
0	Invalid setting
1	Writing 1 to this bit initiates the fixed output control of Hi-Z/low-level/high-level for the TKBOnp output (the equivalent function with rising edge detection of trigger signal by forced output stop 1).

Table 9 - 11 lists the cancel trigger (the TKBPAHTTnp bit of the TKBPAHTFn register) setting for canceling forced output stop 1.

Table 9 - 11 Operation of Cancel Trigger (the TKBPAHTTnp Bit) for Forced Output Stop 1

TKBPACTLn Register		Cancellation of Forced Output Stop 1 by Software
TKBPAHCMnp1	TKBPAHCMnp0	
0	0	Forced output stop 1 starts in response to a trigger input, and forced output stop 1 is canceled on writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp), regardless of the input trigger's signal level.
0	1	Forced output stop 1 starts in response to a trigger input. After that, while the trigger signal is at the high level, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) is disabled. On the other hand, while the input trigger signal is at the low level, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) cancels forced output stop 1.
1	0	Forced output stop 1 starts in response to a trigger input. After that, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) cancels forced output stop 1 on the next cycle of the counter, regardless of the level of the input trigger signal ^{Note} .
1	1	Forced output stop 1 starts in response to a trigger input, and while the trigger signal is in its high-level period, writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) is ineffective. Forced output stop 1 is canceled on the next counter cycle after writing 1 to the forced output stop function cancel trigger bit (TKBPAHTTnp) when the trigger signal is in its low-level period ^{Note} .

Note When a 16-bit timer KB4n is stopped (TKBCEn = 0) without waiting for the next counter cycle, forced output stopping remains in force until the 16-bit timer KB4n is restarted (TKBCEn = 1).

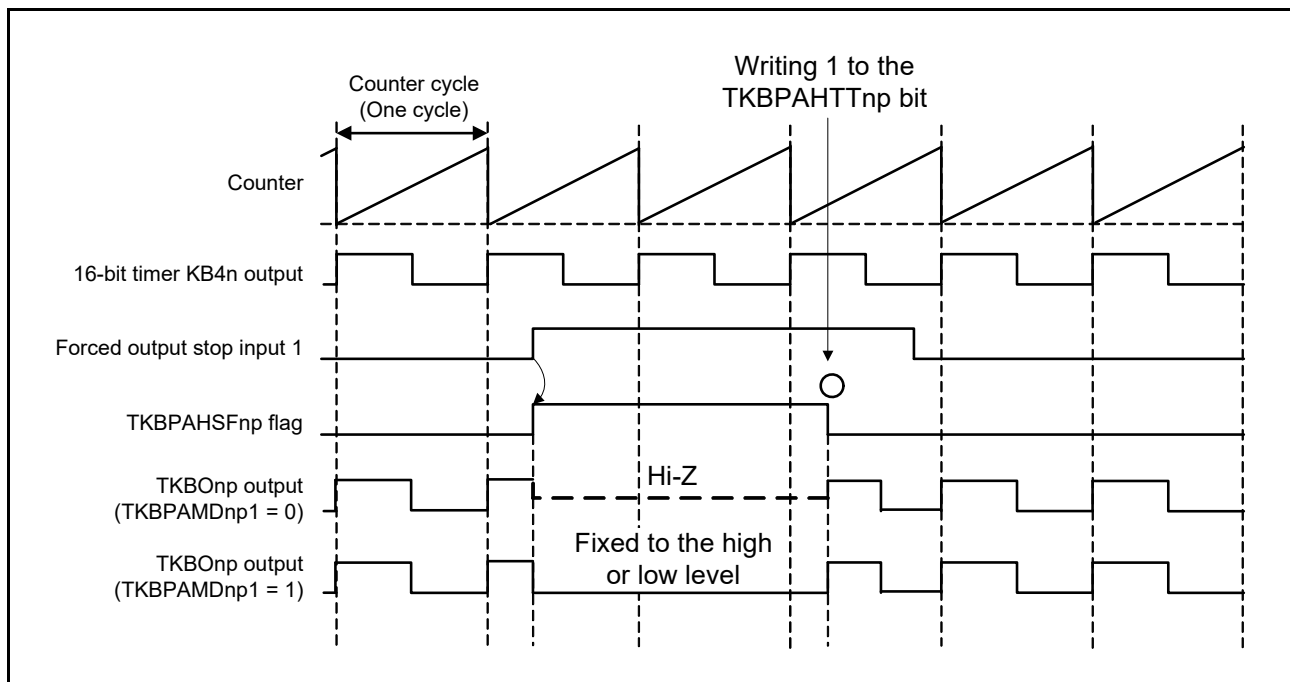
Remark n = 0 to 2; p = 0, 1

9.7.3 Basic operation with forced output stop 1

The following shows the operations with forced output stop 1, using different settings of the TKBPAHCMnp1 and TKBPAHCMnp0 bits. The trigger signal that activates forced output stop 1 (forced output stop input 1) is obtained by the logical OR of the trigger signal selected by the TKBPAHZSn9 to TKBPAHZSn0 bits of the forced output stop function 1 input select register np (TKBPAHZSn) and the setting of the TKBPAHTSn bit of the forced output stop function start trigger register n (TKBPAHFSn).

1. Forced output stop 1 when TKBPAHCMnp[1:0] = 00B

Figure 9 - 104 Forced Output Stop 1 When TKBPAHCMnp[1:0] = 00B



a) TKBPAMDnp1 = 0 (Hi-Z)

The output pins of the 16-bit timers become Hi-Z on detection of the rising edge of forced output stop input 1. Regardless of the level on forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) causes timer output to resume. The TKBPAHSFn flag being set to 1 indicates forced output stop 1 (Hi-Z).

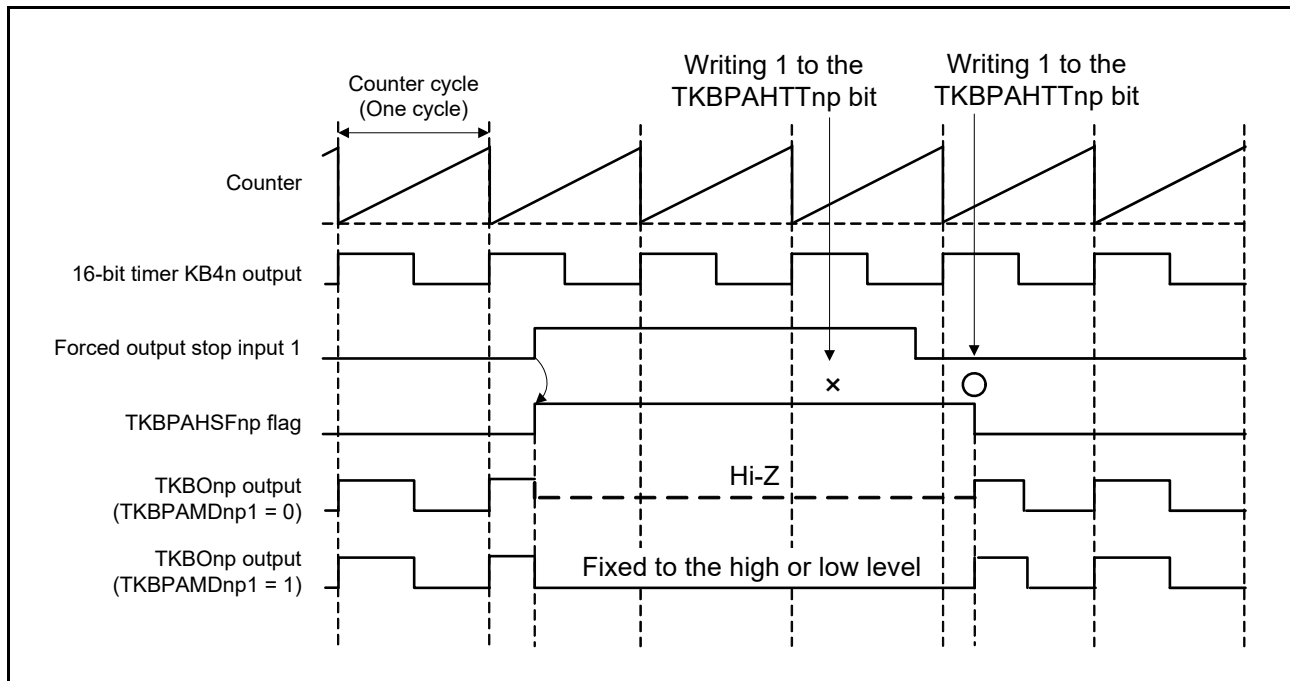
b) TKBPAMDnp1 = 1 (fixed output)

The output pins of the 16-bit timers become high or low in accord with the TKBPAMDnp0 bit setting on detection of the rising edge of forced output stop input 1. Regardless of the level on forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) stops fixed output and resumes output from the timer. The TKBPAHSFn flag being set to 1 indicates forced output stop 1 (fixed low- or high-level output).

Remark n = 0 to 2; p = 0, 1

2. Forced output stop 1 when TKBPAHCMnp[1:0] = 01B

Figure 9 - 105 Forced Output Stop 1 When TKBPAHCMnp[1:0] = 01B



a) TKBPAMDnp1 = 0 (Hi-Z)

The output pins of the 16-bit timers become Hi-Z on detection of the rising edge of forced output stop input 1. During the high-level period of forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) is invalid. Once forced output stop input 1 has turned low, writing 1 to the cancel trigger bit (TKBPAHTTnp) causes timer output to resume. The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (Hi-Z).

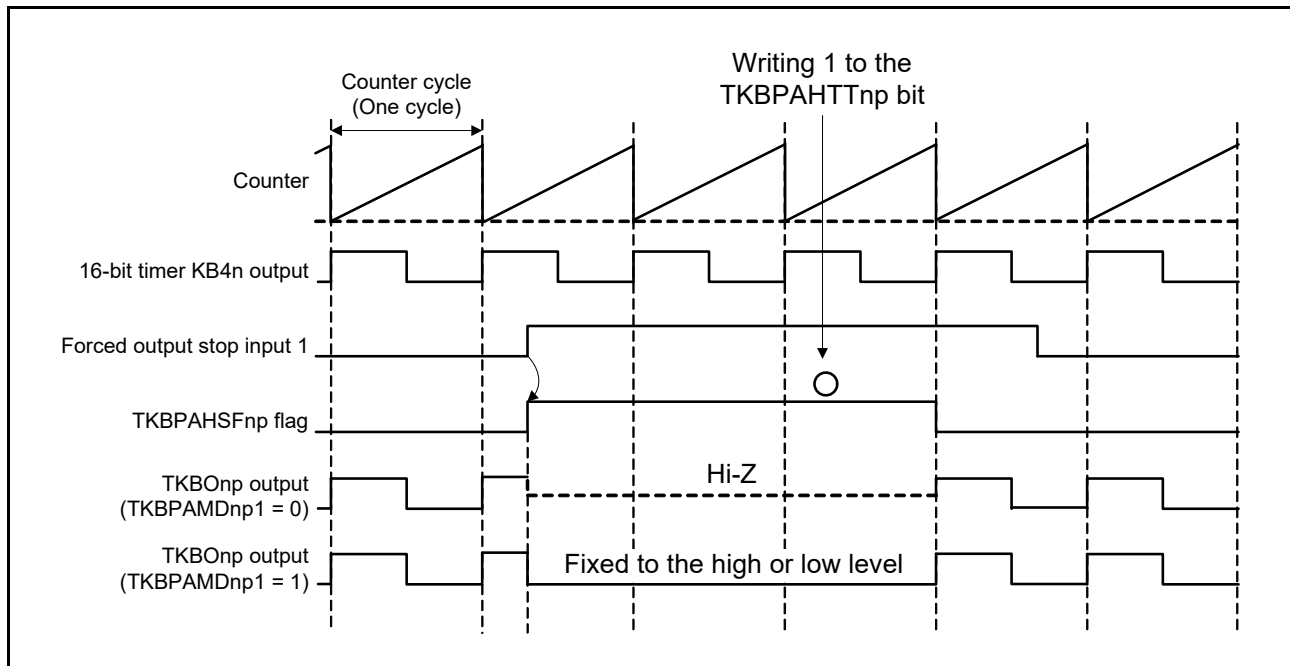
b) TKBPAMDnp1 = 1 (fixed output)

The output pins of the 16-bit timers become high or low in accord with the TKBPAMDnp0 bit setting on detection of the rising edge of forced output stop input 1. During the high-level period of forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) is invalid. Once forced output stop input 1 turned low, writing 1 to the cancel trigger bit (TKBPAHTTnp) stops fixed output and resumes output from the timer. The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (fixed low- or high-level output).

Remark n = 0 to 2; p = 0, 1

3. Forced output stop 1 when TKBPAHCMnp[1:0] = 10B

Figure 9 - 106 Forced Output Stop 1 When TKBPAHCMnp[1:0] = 10B



a) TKBPAMDnp1 = 0 (Hi-Z)

The output pins of the 16-bit timers become Hi-Z on detection of the rising edge of forced output stop input 1. Regardless of the input level of forced output stop input 1, timer output is resumed in the counter cycle immediately after the writing of 1 to the cancel trigger bit (TKBPAHTTnp). The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (Hi-Z).

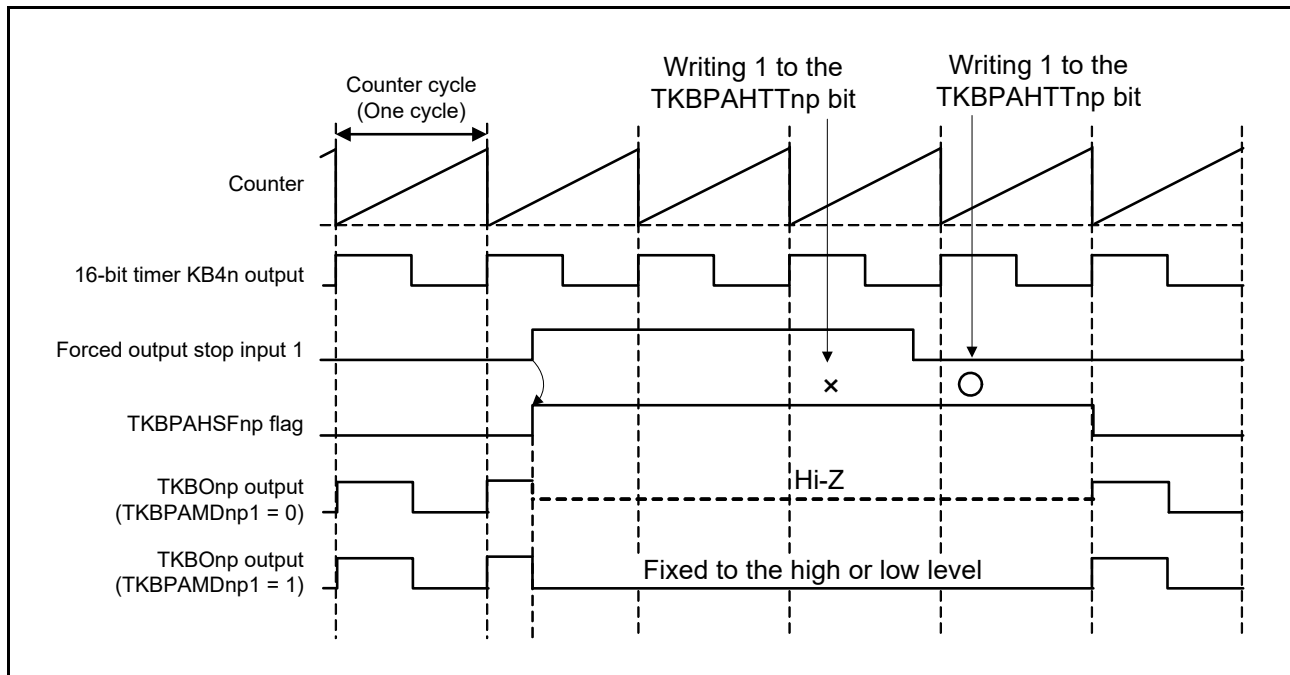
b) TKBPAMDnp1 = 1 (fixed output)

The output pins of the 16-bit timers become high or low in accord with the TKBPAMDnp0 bit setting on detection of the rising edge of forced output stop input 1. Regardless of the input level of forced output stop input 1, the timer pins are released from the fixed output state and timer output is resumed in the counter cycle immediately after the writing of 1 to the cancel trigger bit (TKBPAHTTnp). The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (fixed low- or high-level output).

Remark n = 0 to 2; p = 0, 1

4. Forced output stop 1 when TKBPAHCMnp[1:0] = 11B

Figure 9 - 107 Forced Output Stop 1 When TKBPAHCMnp[1:0] = 11B



a) TKBPAMDnp1 = 0 (Hi-Z)

The output pins of the 16-bit timers become Hi-Z on detection of the rising edge of forced output stop input 1. During the high-level period of forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) is invalid. Timer output is resumed in the counter cycle immediately after the writing of 1 to the cancel trigger bit (TKBPAHTTnp) during the low-level period of forced output stop input 1. The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (Hi-Z).

b) TKBPAMDnp1 = 1 (fixed output)

The output pins of the 16-bit timers become high or low in accord with the TKBPAMDnp0 bit setting on detection of the rising edge of forced output stop input 1. During the high-level period of forced output stop input 1, writing 1 to the cancel trigger bit (TKBPAHTTnp) is invalid. The timer pins are released from the fixed output state and timer output is resumed in the counter cycle immediately after the writing of 1 to the cancel trigger bit (TKBPAHTTnp) during the low-level period of forced output stop input 1. The TKBPAHSFnp flag being set to 1 indicates forced output stop 1 (fixed low- or high-level output).

Remark n = 0 to 2; p = 0, 1

9.8 Operation with Forced Output Stop 2

Generation of a trigger source (one among the outputs from comparators 0 and 1 and INTP0 to INTP7) enables fixing of timer output to the low or high level without CPU intervention, and asynchronously with respect to the fKBKC operating clock for the 16-bit timer KB4n. Release from the forced output stop state proceeds at the beginning of the counter cycle immediately after generation of the trigger source or a transition of the trigger source signal to the inactive level.

9.8.1 Summary of forced output stop 2

Comparator output signals and external interrupt signals are used as the trigger signals for forced output stop 2. The output level selectable at forced output stop is controlled by the TKBPAMDnp[1:0] bits of the TKBPACTLnp register.

Table 9 - 12 lists the relationship between the levels of the p output pins (TKBOnp) of the 16-bit timer KB4n and the level output using forced output stop 2.

Table 9 - 12 Relationship between the Levels of the TKBOnp Pins and the Level Output Using Forced Output Stop 2

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection on Using Forced Output Stop 2
0	0	Fixed to the low level
0	1	Fixed to the high level
1	0	Fixed to the low level
1	1	Fixed to the high level

The selection of a comparator output or an external interrupt signal to be used is controlled by the TKBPAFXSnp9 to TKBPAFXSnp0 bits of the forced output stop function 2 input select register np (TKBPAFXSnp). **Table 9 - 13** lists trigger selection for forced output stop 2 on the p output pins (TKBOnp) of the 16-bit timer KB4n.

Table 9 - 13 Trigger Selection for Forced Output Stop 2 of the TKBOnp Pins

Bit	Selectable Trigger Signals		
	16-bit Timer KB40	16-bit Timer KB41	16-bit Timer KB42
TKBPAFXSnp0	INTP0	INTP0	INTP0
TKBPAFXSnp1	INTP1	INTP1	INTP1
TKBPAFXSnp2	INTP2	INTP2	INTP2
TKBPAFXSnp3	INTP3	INTP3	INTP3
TKBPAFXSnp4	INTP4	INTP4	INTP4
TKBPAFXSnp5	INTP5	INTP5	INTP5
TKBPAFXSnp6	INTP6	INTP6	INTP6
TKBPAFXSnp7	INTP7	INTP7	INTP7
TKBPAFXSnp8	Comparator 0	Comparator 0	Comparator 0
TKBPAFXSnp9	Comparator 1	Comparator 1	Comparator 1

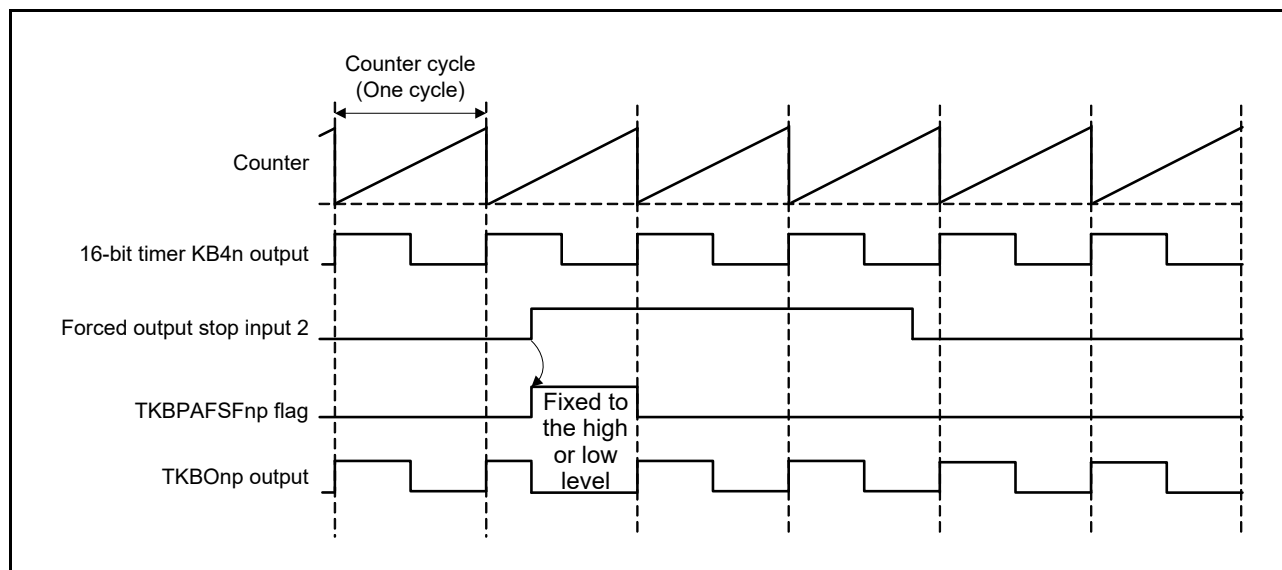
Remark n = 0 to 2; p = 0, 1

9.8.2 Basic operation with forced output stop 2

The following shows the operations with forced output stop 2, using different settings of the TKBPAFCMnp and TKBPAFIXnp bits. The trigger signal that activates forced output stop function 2 (forced output stop input 2) is selected by the TKBPAFXSn9 to TKBPAFXSn0 bits of the forced output stop function 2 input select register np (TKBPAFXSn).

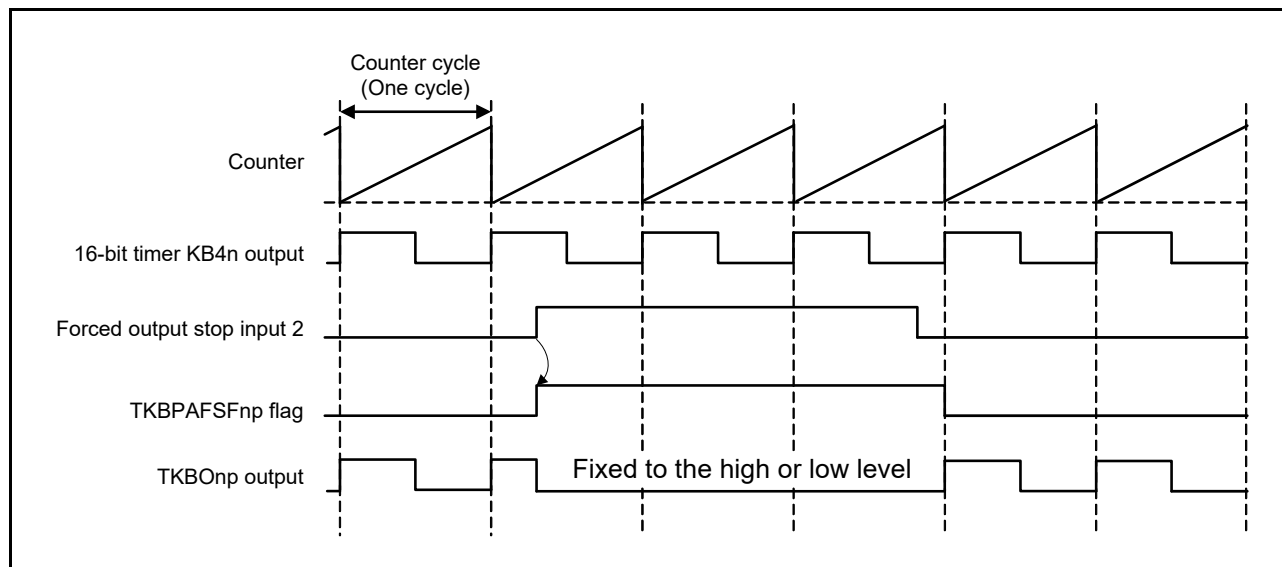
1. Forced output stop 2 when TKBPAFCMnp = 0

Figure 9 - 108 Forced Output Stop 2 When TKBPAFCMnp = 0



The output pins of the 16-bit timers become high or low in accord with the TKBPAMDnp0 bit setting on detection of the rising edge of forced output stop input 2. Regardless of the level on forced output stop input 2, the timer pins are released from the fixed output state and timer output is resumed in the next counter cycle. The TKBPAFSFn flag being set to 1 indicates forced output stop 2 (fixed low- or high-level output).

Remark n = 0 to 2; p = 0, 1

2. Forced output stop 2 when $TKBPAFCMnp = 1$ and $TKBPAFXnp = 0$ Figure 9 - 109 Forced Output Stop 2 When $TKBPAFCMnp = 1$ and $TKBPAFXnp = 0$ 

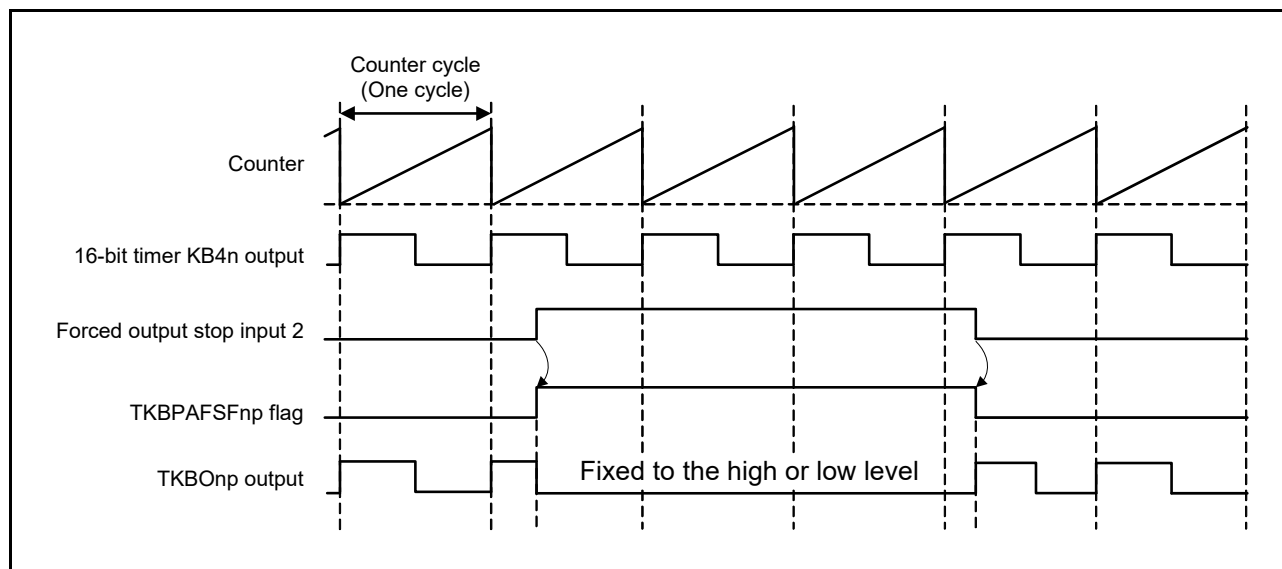
The output pins of the 16-bit timers become high or low in accord with the $TKBPAMDnp0$ bit setting on detection of the rising edge of forced output stop input 2. The timer pins are released from the fixed output state and timer output is resumed in the counter cycle immediately after the falling edge of forced output stop input 2. The $TKBPAFSFn$ flag being set to 1 indicates forced output stop 2 (fixed low- or high-level output).

Remark $n = 0$ to 2; $p = 0, 1$

3. Forced output stop 2 when $TKBPAFCMnp = 1$ and $TKBPAFIXnp = 1$ (fixed-off enabled)

Fixed off is enabled by setting the $TKBPAFIXnp$ bit to 1. This function can only be enabled while the $TKBPAFCMnp$ bit is set to 1.

Figure 9 - 110 Forced Output Stop 2 When $TKBPAFCMnp = 1$ and $TKBPAFIXnp = 1$ (Fixed-off Enabled)



The output pins of the 16-bit timers become high or low in accord with the $TKBPAMDnp0$ bit setting on detection of the rising edge of forced output stop input 2. The timer pins are released from the fixed output state in synchronization with the detected signal and timer output is resumed in the counter cycle immediately after the falling edge of forced output stop input 2. The $TKBPAFSFnp$ flag being set to 1 indicates forced output stop 2 (fixed low- or high-level output). Activation of forced stopping of the output is not synchronized with f_{KBKC} so it proceeds on a rising edge of forced output stop input 2. Termination of forced stopping of the output is synchronized with f_{KBKC} , so it proceeds when two or three cycles of f_{KBKC} have elapsed after a falling edge of forced output stop input 2. If a further rising edge of forced output stop input 2 is detected while synchronization processing is in progress, release from the forced output stop state does not proceed so the fixed output state is retained.

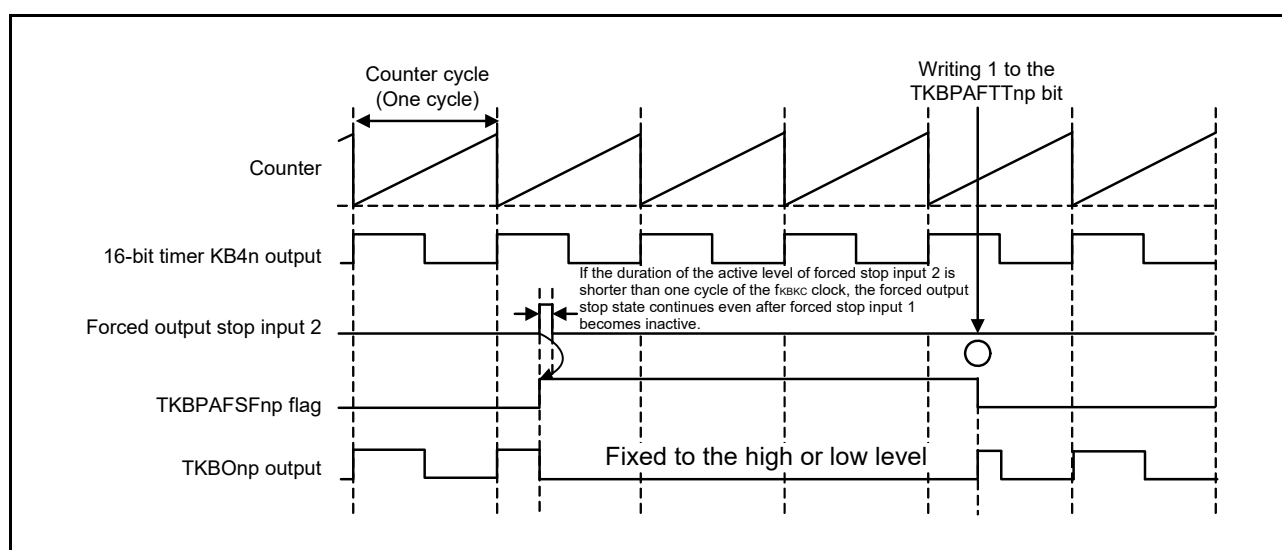
Remark $n = 0$ to 2; $p = 0, 1$

With fixed off, the start of forced output stopping is not synchronized with the clock and the output is stopped as soon as forced output stop input 2 becomes active. However, the cancellation of stopping is synchronized with the clock. Therefore, if the duration of the active level of forced output stop input 2 is shorter than one cycle of the f_{KBK} clock, forced output stop 2 may not be canceled and the output may continue to be in the forced output stop state in some cases. In such cases, use the following procedure to cancel forced output stop.

1. Compare the values of the $TKBPAFIFnp$ and $TKBPAFSFnp$ flags. If the value of the $TKBPAFIFnp$ flag is 0 (forced output stop input 2 is at the low level) and that of the $TKBPAFSFnp$ is 1 (the timer pin is in the forced output stop state), proceed to step 2.
2. Set the $TKBPAFTTnp$ bit to 1.
3. Check that the value of the $TKBPAFSFnp$ flag is 0 (the timer pin is released from the forced output stop state).

Figure 9 - 111 shows the operation with fixed off when a short pulse is input.

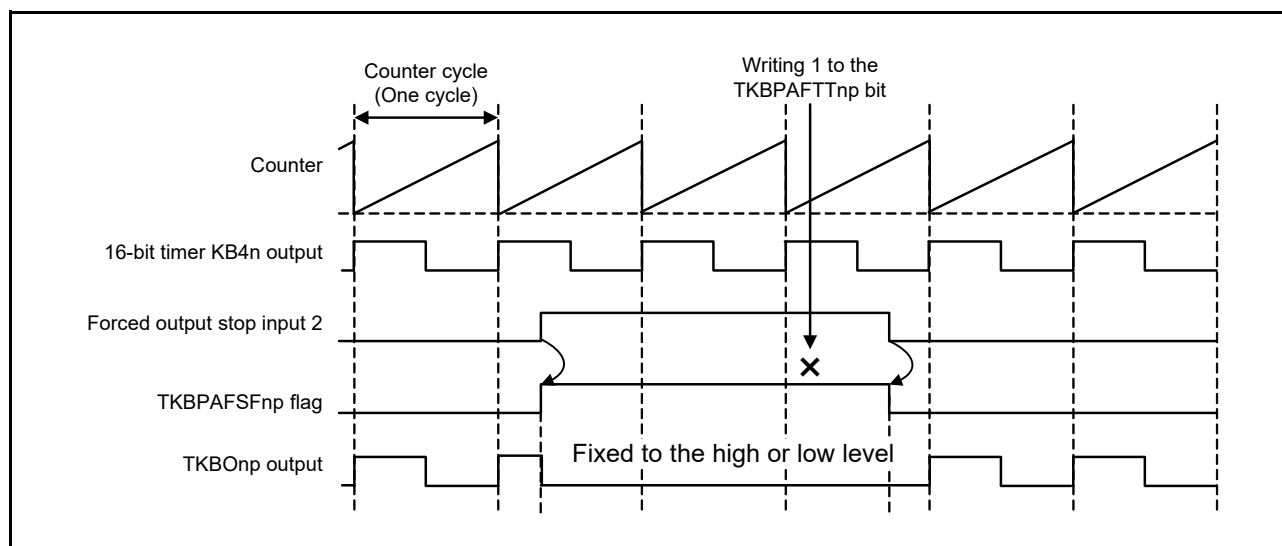
Figure 9 - 111 Operation with Fixed Off When a Short Pulse Is Input



Writing 1 to the cancel trigger bit ($TKBPAFTTnp$) is ignored while forced output stop input 2 is active (high level).

Figure 9 - 112 shows the operation of writing to the cancel trigger bit during the active-level period of forced output stop input 2.

Figure 9 - 112 Writing to the Cancel Trigger Bit During the Active-level Period of Forced Output Stop Input 2



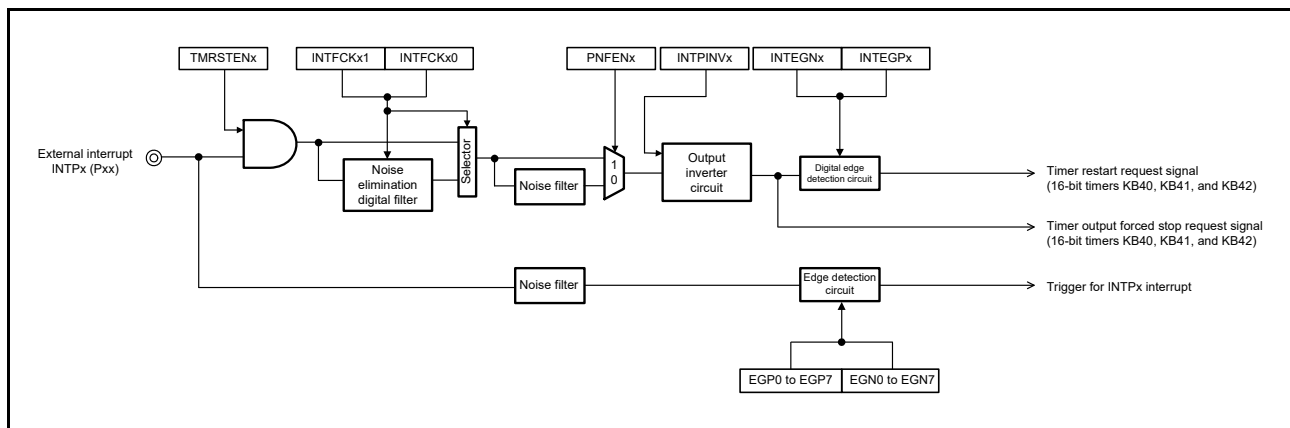
9.9 Notes on Using Interlocking of External Interrupts INTPx with 16-bit Timers KB40, KB41, and KB42

The INTPx external interrupts can be used as triggers for the INTPx interrupts, signals for requesting restart of 16-bit timers KB40, KB41, and KB42, and signals for requesting forced stop of the output from the timers. The signals for requesting restart of 16-bit timers KB40, KB41, and KB42 as well as the input signal edges detected on the INTPx pins can be selected as triggers for the INTPx interrupts. Thus, the INTPx interrupts can be generated in synchronization with generation of the restart signals for 16-bit timers KB40, KB41, and KB42.

According to the function to be used, set the following registers.

- External interrupt control register x (INTPCTLx)
- External interrupt rising edge enable register 0 (EGP0)
- External interrupt falling edge enable register 0 (EGN0)

Figure 9 - 113 Block Diagram of Hardware for External Interrupts INTPx



Remark 1. EGP0 to EGP7 bits: Bits in the external interrupt rising edge enable register (EGP0: bits 0 to 7)
EGN0 to EGN7 bits: Bits in the external interrupt falling edge enable register (EGN0: bits 0 to 7)

Remark 2. x = 0 to 7

Section 10 Realtime Clock (RTC)

10.1 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 HALT or STOP mode, to trigger transitions of the A/D converter to the SNOOZE mode, or as the trigger for alternation of the display on the LCD. A signal with a 0.5-second cycle (signal name: sub500ms) is also capable of driving alternation of the display on the LCD.

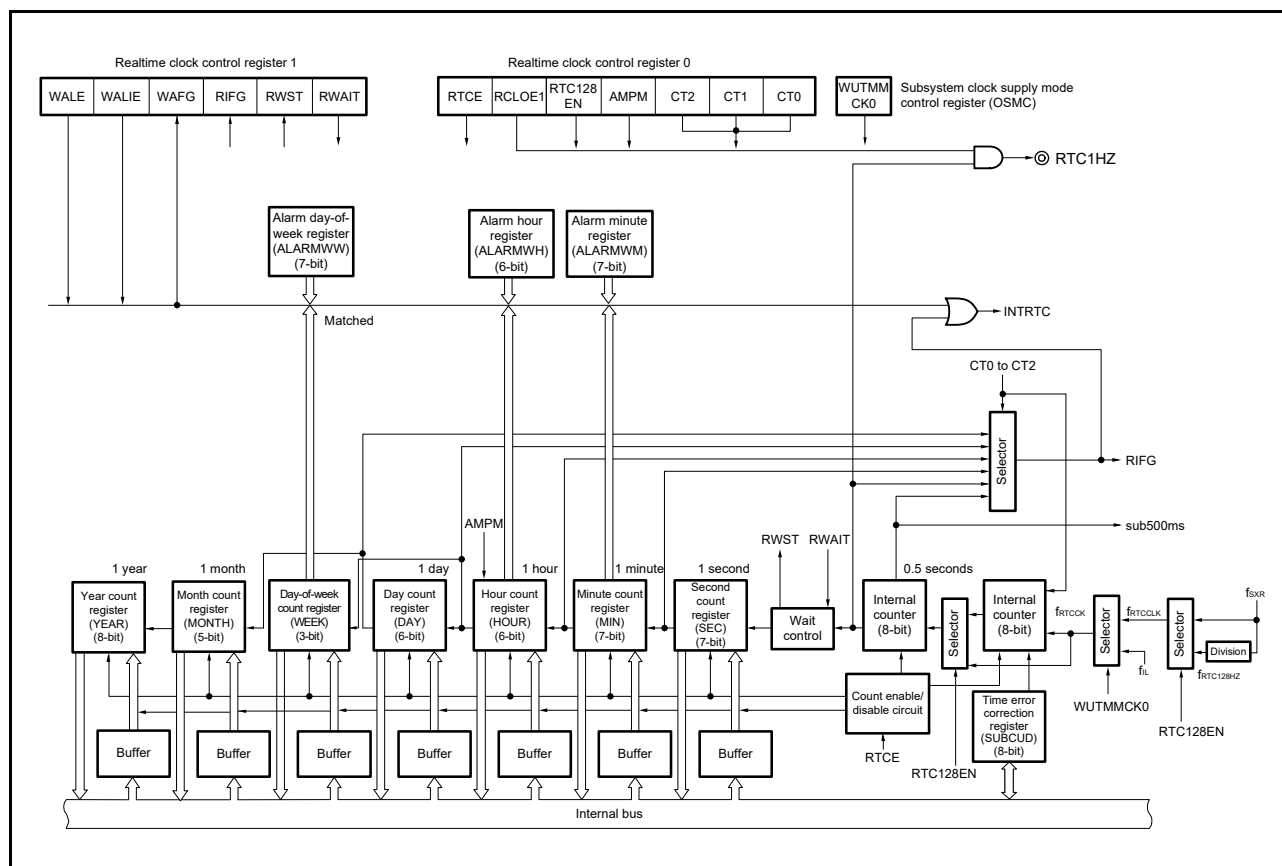
10.2 Configuration of the Realtime Clock

The realtime clock includes the following hardware blocks.

Table 10 - 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 10 - 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 (fsXR = 32.768 kHz) is selected as the operating clock 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.

10.3 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 registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

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.

10.3.1 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 10 - 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	IICA1EN	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. - The realtime clock can operate.						
	1	- The SFRs used by the realtime clock can be read and written. - The realtime clock can operate.						

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 (f_{RTCLK}) has become stable. If RTCWEN is 0, writing to the registers for controlling the realtime clock is ignored, and the value read is 00H.

- 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)

Note, however, that writing to the following registers is valid.

- Subsystem clock supply mode control register (OSMC)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

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.

Caution 3. Be sure to clear bit 1 to 0.

10.3.2 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 a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined^{Note}.

Note The value of the RTCLPC and WUTMMCK0 bits following a reset is 0 and that of the HIPREC bit is 1.

Figure 10 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
 After reset: Undefined^{Note 1}
 R/W: R/W^{Note 2}

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 3, 4}							

Note 1. The value of the RTCLPC and WUTMMCK0 bits following a reset is 0 and that of the HIPREC bit is 1.

Note 2. Be sure to clear bits 6, 5, and 1 to 0. Bits 3, 2, and 0 are read-only bits and writing to them is ignored.

Note 3. Setting the WUTMMCK0 bit to 1 is prohibited while the subsystem clock X is oscillating.

Note 4. 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, 8-bit interval timer, serial interfaces UARTA0 to UARTA3, external signal sampler, clock output/buzzer output controller, and LCD controller/driver 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.

10.3.3 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 10 - 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 WUTMMCK0 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 10 - 2 Displayed Time Digits** shows the time (hour) digits indicated according to the setting of this bit.

Figure 10 - 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 second (same time as second count up)
0	1	1	Once per minute (second 00 of every minute)
1	0	0	Once an hour (minute 00 and second 00 of every hour)
1	0	1	Once a day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once a 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 HALT or STOP mode immediately after setting the RTCE bit to 1, follow the procedure in **Figure 10 - 18 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

10.3.4 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 10 - 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 _{RTCK} 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 state 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.	

RWST	Wait status flag of realtime clock ^{Note 1}
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.	

Figure 10 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (2/2)

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. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. After setting this bit to 1, it takes up to one cycle of fRTCK until the counter value can be actually read or written (RWST = 1). Notes 2, 3 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. Bit 1 is read-only.

Note 2. When the RWAIT bit is set to 1 within one cycle of fRTCK 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 (fRTCK).

Note 3. When the RWAIT bit is set to 1 within one cycle of fRTCK 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 (fRTCK).

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 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.

10.3.5 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 `frtclk` 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 10 - 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 10.4.3 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.

10.3.6 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 `frtclk` 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 10 - 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 10.4.3 Reading from and writing to the counters of the realtime clock.

10.3.7 Hour count register (HOUR)

The HOUR is an 8-bit register that takes a value of 0 to 23 or 1 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 f_{RTCK} 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 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 10 - 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) or PM (1) when AMPM = 0 (that is, when 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 10.4.3 Reading from and writing to the counters of the realtime clock.

Table 10 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 10 - 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.

10.3.8 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 `frtcck` 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 10 - 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 10.4.3 Reading from and writing to the counters of the realtime clock.

10.3.9 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 10 - 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 release from the reset state, 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 10.4.3 Reading from and writing to the counters of the realtime clock.

10.3.10 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 `frtclk` 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 10 - 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 10.4.3 Reading from and writing to the counters of the realtime clock.

10.3.11 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 `frtclk` 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 10 - 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 10.4.3 Reading from and writing to the counters of the realtime clock.

10.3.12 Time error correction register (SUBCUD)

This register is used to correct the time with high accuracy when it is running slow or fast by adjusting the value that is considered an overflow from the internal counter (16 bits) to the second count register (SEC) (reference value: 7FFFH). The SUBCUD register can be set by an 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 10 - 13 Format of Time Error Correction Register (SUBCUD)

Address: F0227H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of time error correction timing
0	Corrects time error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects time error only when the second digits are at 00 (every 60 seconds).

Writing to the SUBCUD register at the following timing is prohibited.

- When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H
- When DEV = 1 is set: For a period of SEC = 00H

F6	Setting of time error correction value
0	Increases by $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$.
1	Decreases by $\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$.

When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the time error is not corrected. * is 0 or 1.
/F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).

Range of correction value: (when F6 = 0) 2, 4, 6, 8, ... , 120, 122, 124
(when F6 = 1) -2, -4, -6, -8, ... , -120, -122, -124

The range of value that can be corrected by using the time error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	-189.2 ppm to 189.2 ppm	-63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Caution Time error correction cannot be used in the 128-Hz operating mode (RTC128EN = 1); it can only proceed if the setting of RTC128EN is 0.

Remark If a correctable range is -63.1 ppm or lower and 63.1 ppm or higher, set DEV to 0.

10.3.13 Alarm minute register (ALARMWWM)

This register is used to set minutes of alarm. The ALARMWWM 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 10 - 14 Format of Alarm Minute Register (ALARMWWM)

Address: F0228H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

10.3.14 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 10 - 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) or PM (1) when AMPM = 0 (that is, when the 12-hour system is selected).

10.3.15 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 10 - 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	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour	Hour	Minute	Minute	Hour	Hour	Minute	Minute
	W 0	W 1	W 2	W 3	W 4	W 5	W 6	10	1	10	1	10	1	10	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

10.3.16 Registers for controlling the port functions multiplexed with the realtime clock output

Set the following registers for controlling the port functions multiplexed with the realtime clock output.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

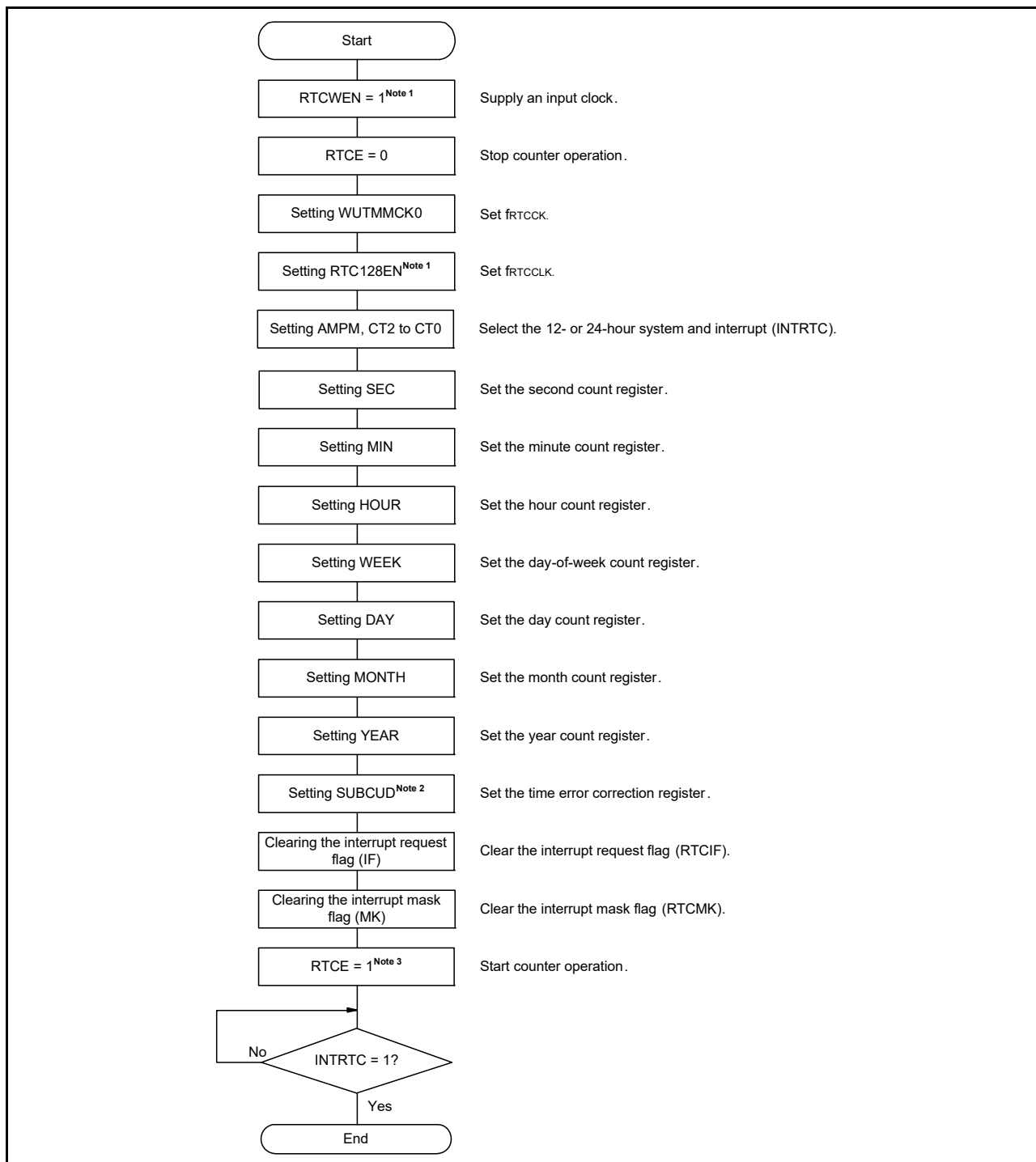
- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

10.4 Operations of the Realtime Clock

10.4.1 Starting the Realtime Clock Operation

Figure 10 - 17 Procedure for Starting the Realtime Clock Operation



Note 1. First set the RTCWEN bit to 1 and set the RTC128EN bit as desired, while oscillation of the count clock (f_{RTCCK}) 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 **10.4.6 Example of time error correction by the realtime clock**. Time error correction cannot be used while the setting of the RTC128EN bit is 1.

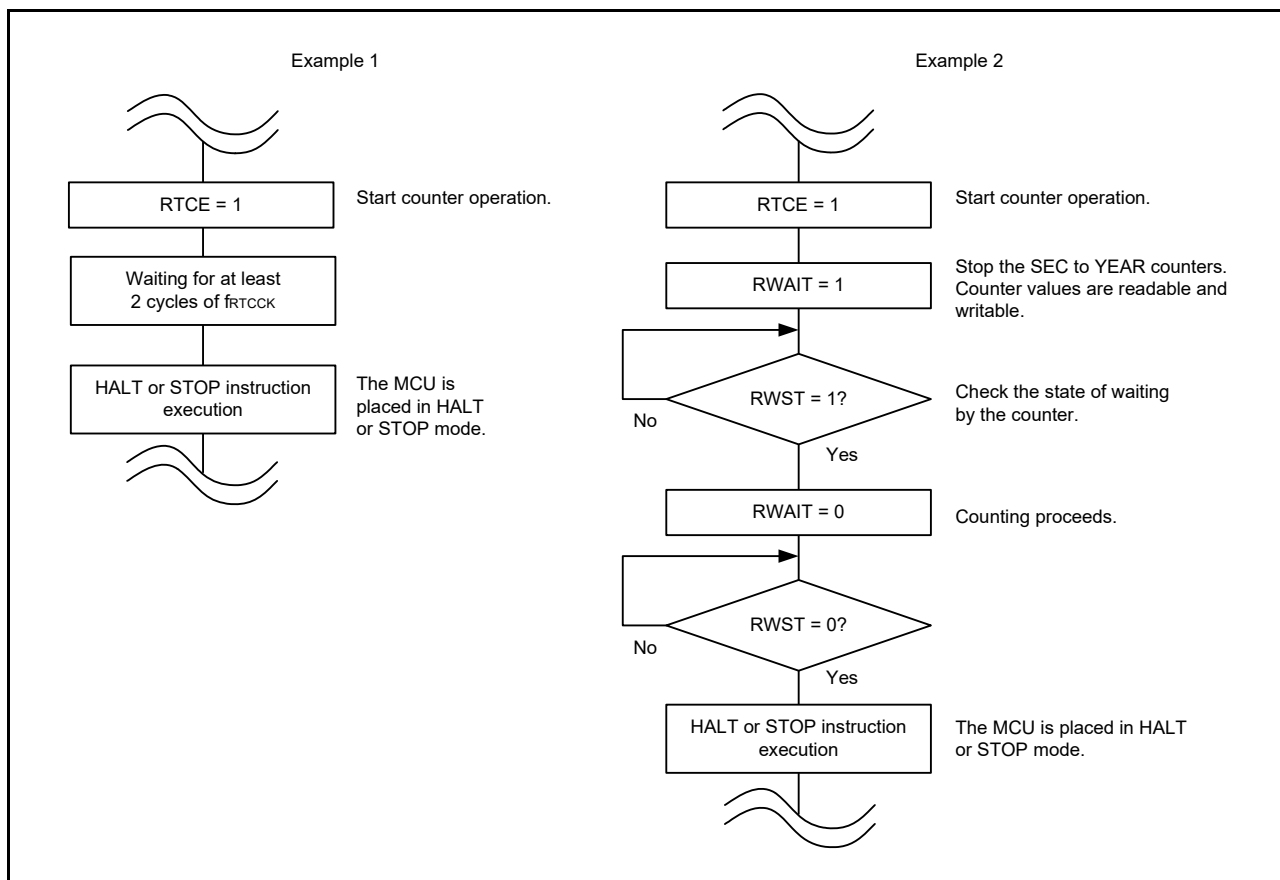
Note 3. Confirm the procedure described in **10.4.2 Shifting to HALT or STOP mode after starting operation** when shifting to HALT or STOP mode without waiting for INTRTC = 1 after RTCE = 1.

10.4.2 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 HALT or STOP mode after the INTRTC interrupt has occurred.

- Make a transition to HALT or STOP mode when at least two counter clock cycles (f_{RTCK}) have elapsed after setting the RTCE bit to 1 (see **Figure 10 - 18, 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 or STOP mode will proceed (see **Figure 10 - 18, Example 2**).

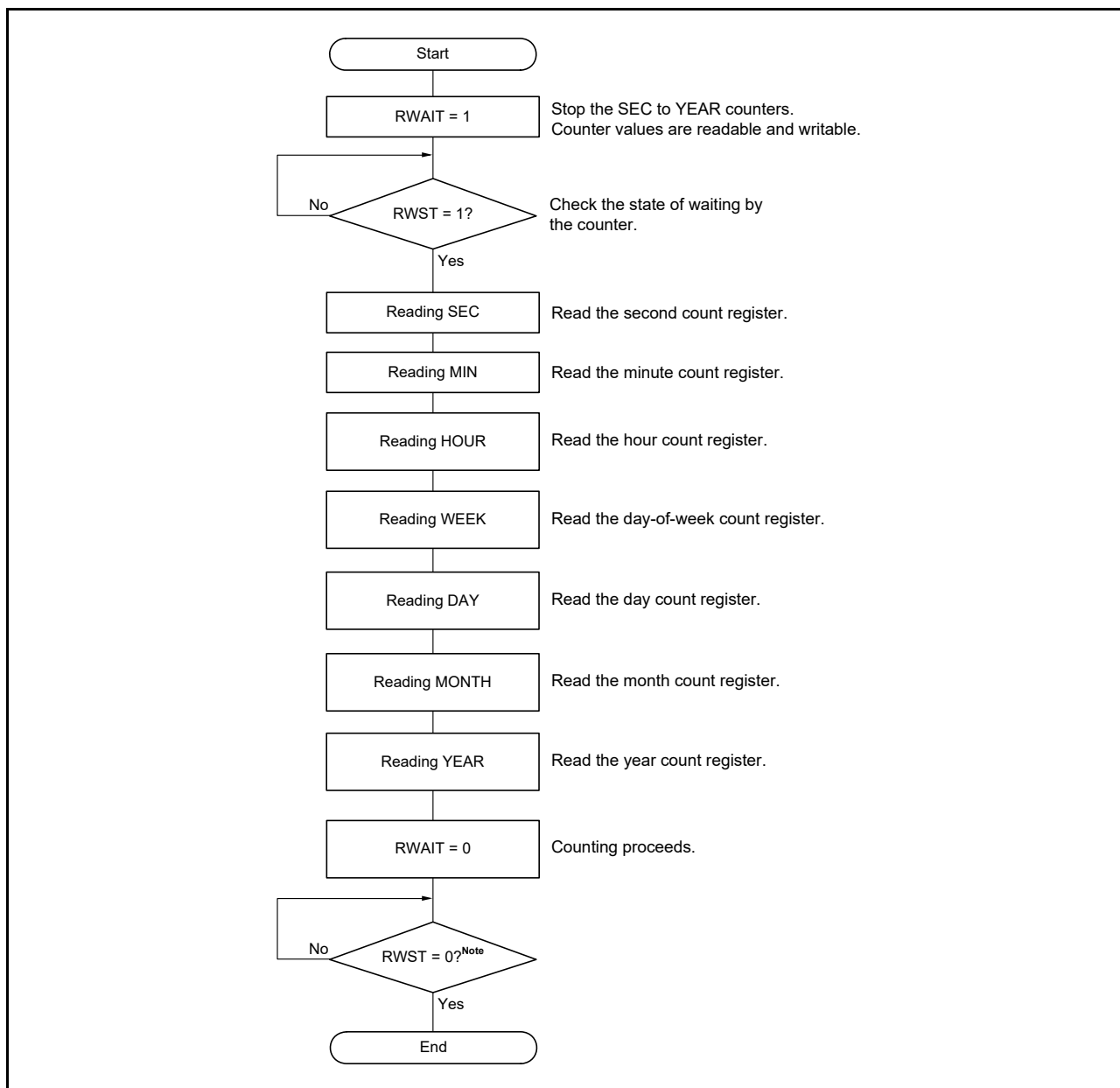
Figure 10 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1



10.4.3 Reading from and writing to the counters of the realtime clock

Read or write to the counter after setting RWAIT to 1 first. Set RWAIT to 0 after completion of reading or writing to the counter.

Figure 10 - 19 Procedure for Reading the Realtime Clock

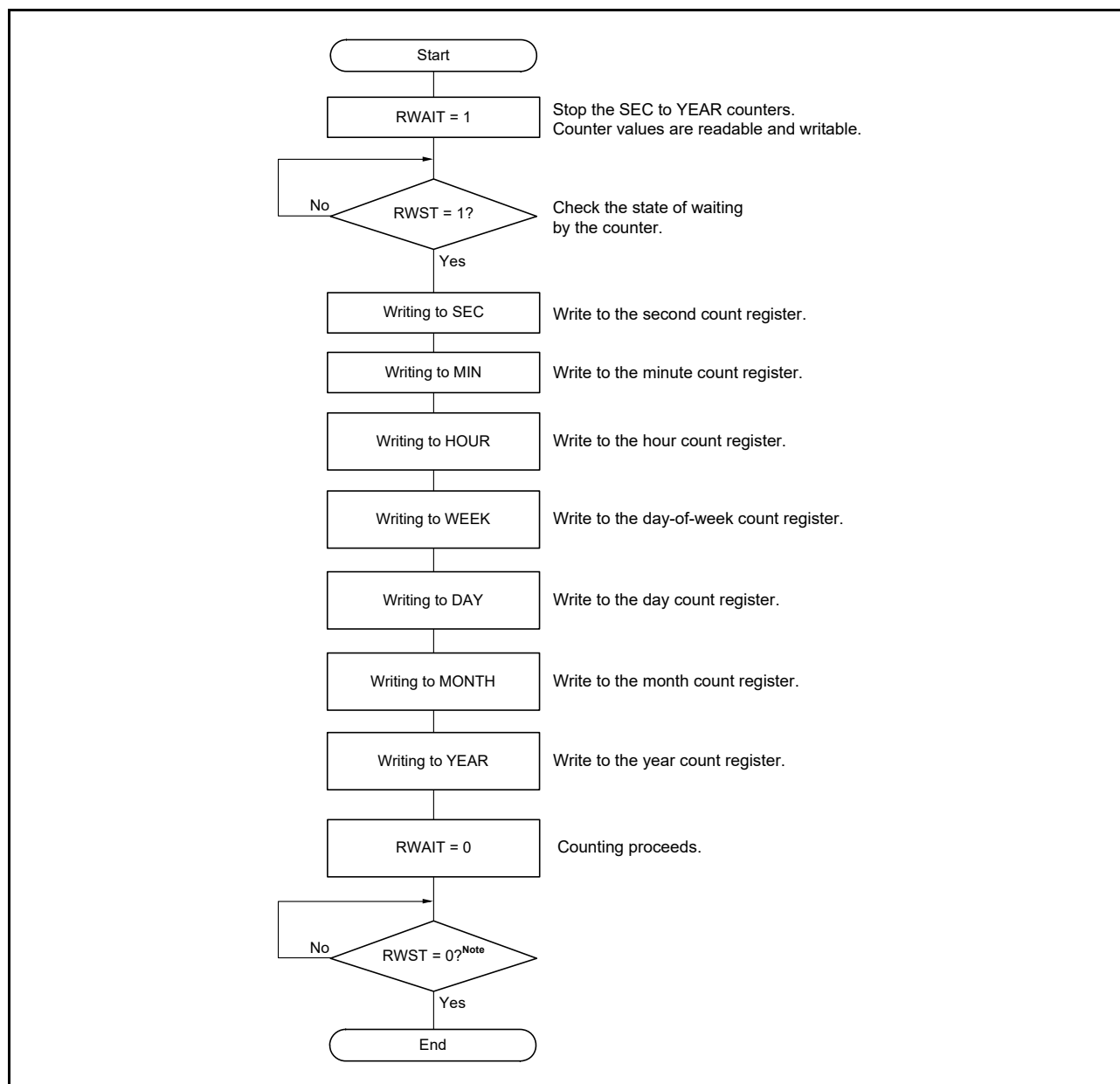


Note Be sure to confirm that RWST = 0 before setting HALT or STOP mode.

Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

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 be read and only some registers may be read.

Figure 10 - 20 Procedure for Writing to the Realtime Clock



Note Be sure to confirm that RWST = 0 before setting HALT or 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. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

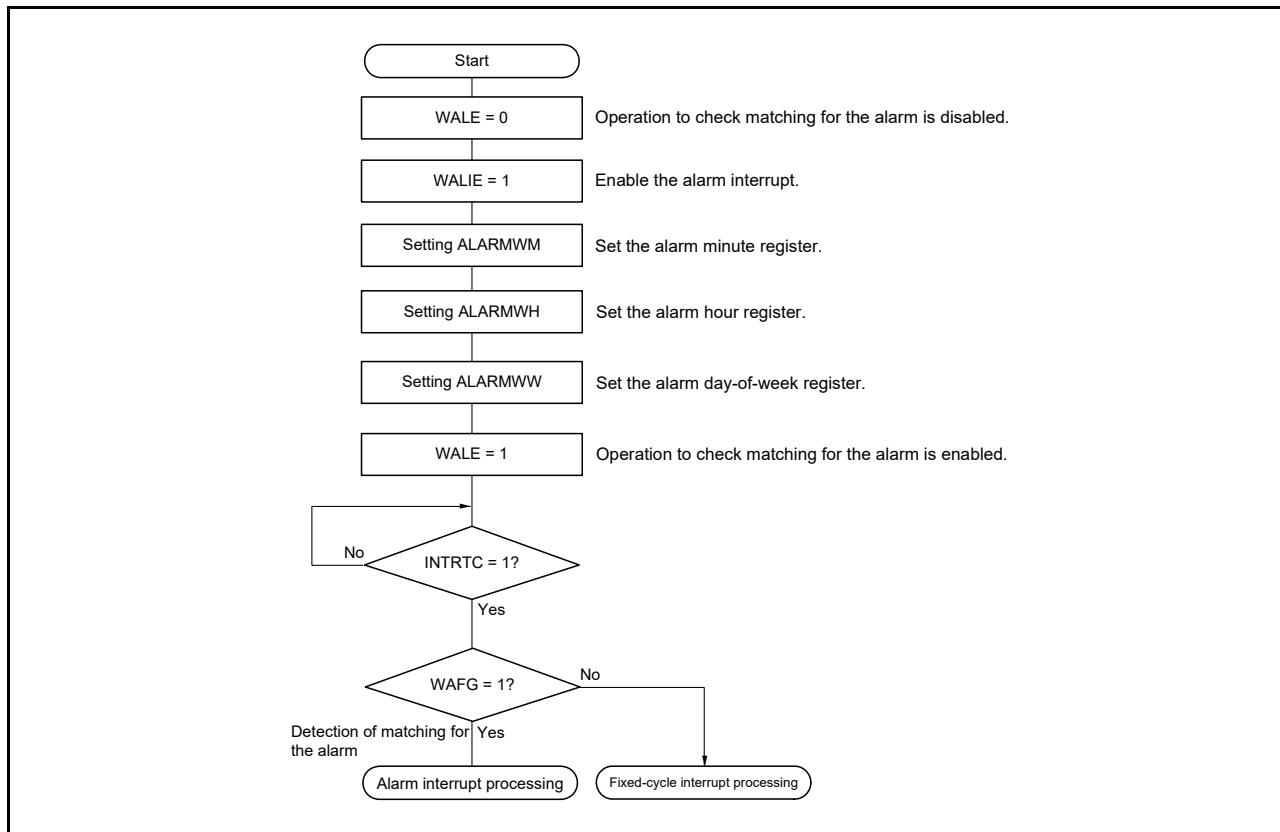
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

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.

10.4.4 Setting alarm by the realtime clock

Set time of alarm after setting WALE to 0 (alarm operation invalid) first.

Figure 10 - 21 Alarm Processing Procedure

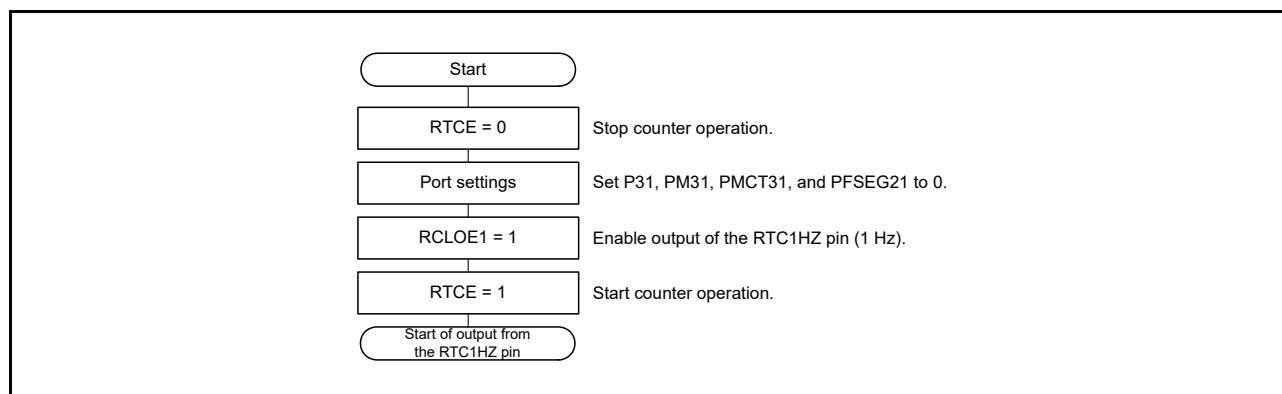


Remark 1. The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm day-of-week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm 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.

10.4.5 1 Hz output by the realtime clock

Figure 10 - 22 1 Hz Output Setting Procedure



Caution First set the RTCWEN bit to 1, while oscillation of the count clock (f_{RTCCK}) is stable.

Remark 1. The output pin for 1 Hz in the above figure is that when PIORx = 0 in a 64- to 100-pin product.

Remark 2. When the 1-Hz output is to be used in a 44- to 52-pin product, allocate the RTC1HZ pin by setting the corresponding bits in the relevant peripheral I/O redirection register (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

10.4.6 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)

$$\begin{aligned}\text{Correction value}^{\text{Note}} &= \text{Number of correction counts in 1 minute} \div 3 \\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3\end{aligned}$$

(When DEV = 1)

$$\begin{aligned}\text{Correction value}^{\text{Note}} &= \text{Number of correction counts in 1 minute} \\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60\end{aligned}$$

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 = $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$

(When F6 = 1) Correction value = $-\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$

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 (f_{RTCCK}). It can be calculated from the output frequency of the RTC1HZ pin $\times$ 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 **10.4.5 1 Hz output by the realtime clock** for the setting procedure of the RTC1Hz output, and see **14.4 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:

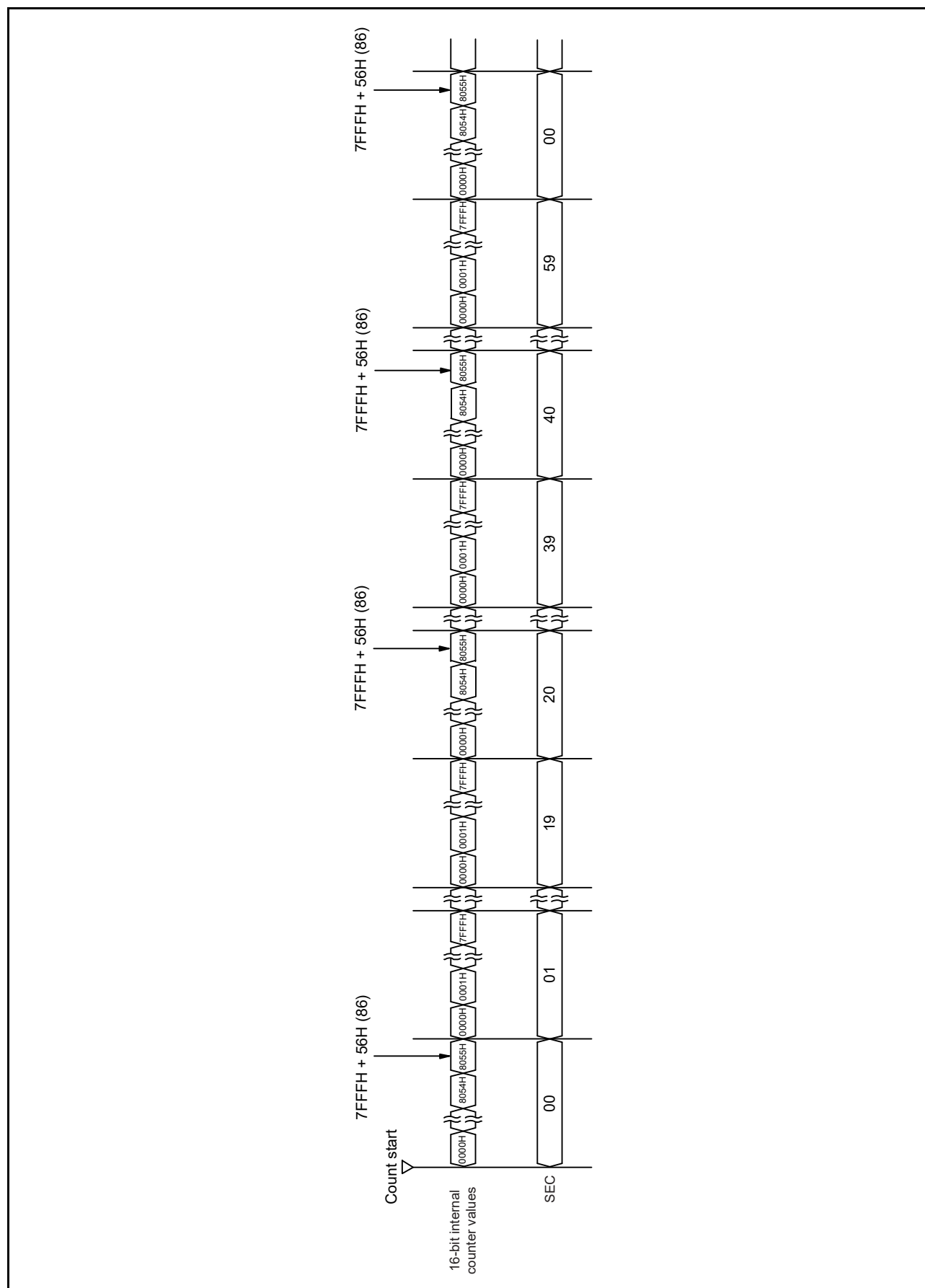
If the correction value is 0 or larger (the clock is running slow), set F6 to 0. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}
 \{ (F5, F4, F3, F2, F1, F0) - 1 \} \times 2 &= 86 \\
 (F5, F4, F3, F2, F1, F0) &= 44 \\
 (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)
 \end{aligned}$$

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 10 - 23 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 10 - 23 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 **10.4.5 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 × 0.9999817 ≈ 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:

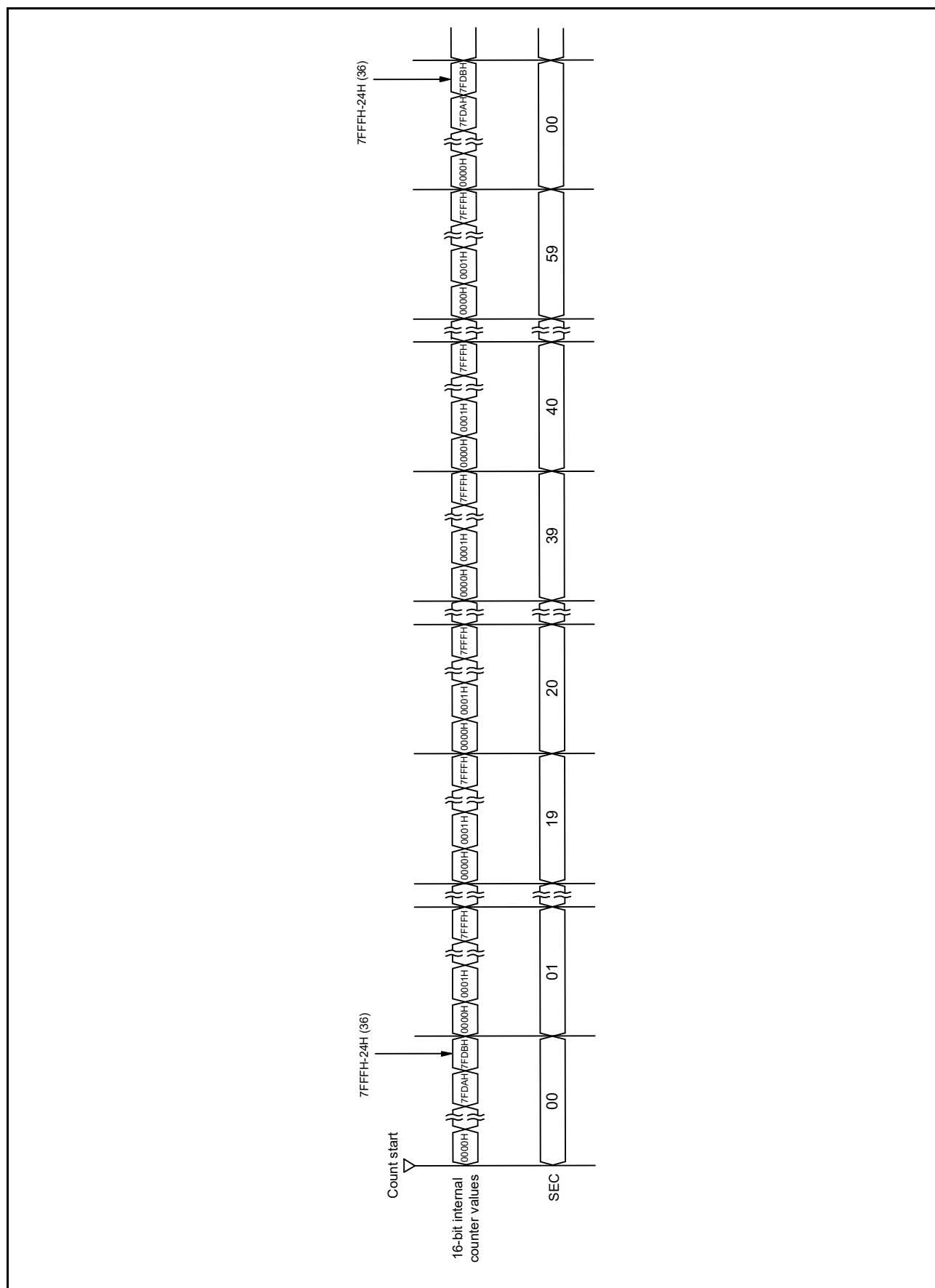
If the correction value is 0 or less (the clock is running fast), set F6 to 1. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}-\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2 &= -36 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1) \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0)\end{aligned}$$

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 10 - 24 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).

Figure 10 - 24 Operation for Correction when the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



Section 11 8-bit Interval Timers

An 8-bit interval timer unit has two 8-bit interval timers (referred to as channels 0 and 1), and each is capable of running independently. The two 8-bit interval timers of a unit can also be cascade-connected to run as a single 16-bit interval timer. The number of units and channels supported by the RL78/L23 depends on the product as indicated in the list below.

Number of Pins		44 to 64		80 and 100
Flash Memory		64 to 128 Kbytes	256 to 512 Kbytes	128 to 512 Kbytes
Unit 0	Channel 0	✓	✓	✓
	Channel 1	✓	✓	✓
Unit 1	Channel 0	✓	✓	✓
	Channel 1	✓	✓	✓
Unit 2	Channel 0	✓	✓	✓
	Channel 1	✓	✓	✓
Unit 3	Channel 0	✓	✓	✓
	Channel 1	✓	✓	✓
Unit 4	Channel 0	—	✓	✓
	Channel 1	—	✓	✓
Unit 5	Channel 0	—	✓	✓
	Channel 1	—	✓	✓
Unit 6	Channel 0	—	✓	✓
	Channel 1	—	✓	✓
Unit 7	Channel 0	—	✓	✓
	Channel 1	—	✓	✓

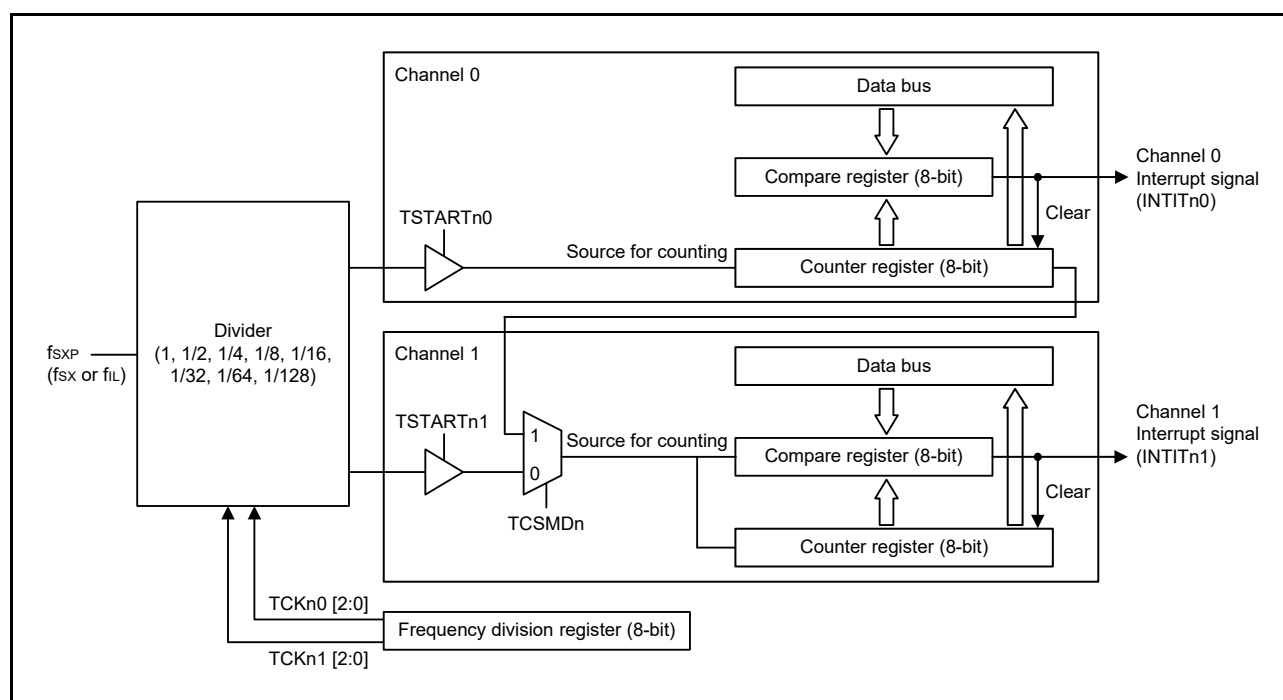
11.1 Overview

An 8-bit interval timer is driven by the fsXP clock, which is asynchronous with respect to the CPU. **Table 11 - 1** lists the specifications of and **Figure 11 - 1** is a block diagram of the 8-bit interval timer unit.

Table 11 - 1 Specifications of the 8-bit Interval Timer Unit

Item	Description
Source for counting (operating clock)	- fsXP, fsXP/2, fsXP/4, fsXP/8, fsXP/16, fsXP/32, fsXP/64, fsXP/128 - fsXP indicates the fsX or fIL clock.
Operating mode	8-bit interval timer mode Each of channels 0 and 1 runs as an independent 8-bit counter. 16-bit interval timer mode Channels 0 and 1 are cascade-connected to run as a 16-bit counter.
Interrupt	Output when the counter value matches the value for comparison.

Figure 11 - 1 Block Diagram of the 8-bit Interval Timer Unit



Remark 1. TSTARTn_i (i = 0, 1), TCSMD_n, TCLKEN_n: Bits in the TRTCR_n register
 TCKn_i [2:0]: Bits in the TRTMD_n register

Remark 2. n: Unit number (n = 0 to 7)

11.2 I/O Pins

The 8-bit interval timers do not have I/O pins.

11.3 Registers for Controlling the 8-bit Interval Timers

The following registers are used to control the 8-bit interval timers.

- 8-bit interval timer counter registers ni (TRT ni)**Note 1**
- 8-bit interval timer counter register n (TRT n)**Note 2**
- 8-bit interval timer compare registers ni (TRTCMP ni)**Note 1**
- 8-bit interval timer compare register n (TRTCMP n)**Note 2**
- 8-bit interval timer control register n (TRTCR n)
- 8-bit interval timer frequency division register n (TRTMD n)

Note 1. These registers are only accessible when the TCSMD n bit in the TRTCR n register = 0.

Note 2. This register is only accessible when the TCSMD n bit in the TRTCR n register = 1.

Remark n : Unit number ($n = 0$ to 7); i : Channel number ($i = 0, 1$)

11.3.1 8-bit interval timer counter registers ni (TRTni)

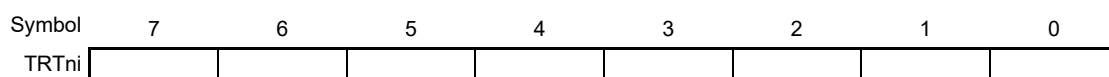
The TRTni registers are 8-bit interval timer counter registers that count up cycles of the counter clock. The TRTni registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 11 - 2 Format of 8-bit Interval Timer Counter Registers ni (TRTni)

Address: F0550H (TRT00), F0551H (TRT01), F0558H (TRT10), F0559H (TRT11), F0560H (TRT20),
F0561H (TRT21), F0568H (TRT30), F0569H (TRT31), F0570H (TRT40), F0571H (TRT41),
F0578H (TRT50), F0579H (TRT51), F0580H (TRT60), F0581H (TRT61), F0588H (TRT70), F0589H (TRT71)

After reset: 00H

R/W: RNotes 1, 2



Note 1. Each TRTni register is set to 00H two cycles of the counter clock after write access to the corresponding TRTCMPni compare register. See **11.4.4 Timing for updating compare register values**.

Note 2. These registers are only accessible when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

Remark n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

11.3.2 8-bit interval timer counter register n (TRTn)

The TRTni registers operate as a single 16-bit counter register (TRTn) when the 8-bit interval timer unit is used in the 16-bit interval timer mode. The TRTn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

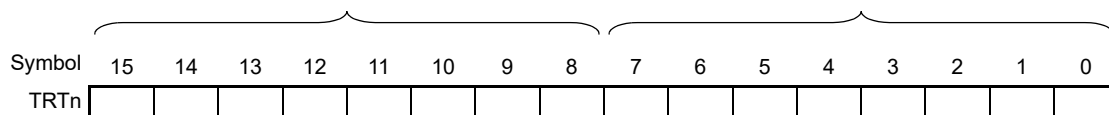
Figure 11 - 3 Format of 8-bit Interval Timer Counter Register n (TRTn)

Address: F0550H (TRT0), F0558H (TRT1), F0560H (TRT2), F0568H (TRT3), F0570H (TRT4), F0578H (TRT5),
F0580H (TRT6), F0588H (TRT7)

After reset: 0000H

R/W: RNotes 1, 2

F0551H (TRT01), F0559H (TRT11),	F0550H (TRT00), F0558H (TRT10),
F0561H (TRT21), F0569H (TRT31),	F0560H (TRT20), F0568H (TRT30),
F0571H (TRT41), F0579H (TRT51),	F0570H (TRT40), F0578H (TRT50),
F0581H (TRT61), F0589H (TRT71)	F0580H (TRT60), F0588H (TRT70)



Note 1. The TRTn register is set to 0000H two cycles of the counter clock after write access to the TRTCMPn compare register. See **11.4.4 Timing for updating compare register values**.

Note 2. This register is only accessible when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 1.

Remark n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

11.3.3 8-bit interval timer compare registers ni (TRTCMPni)

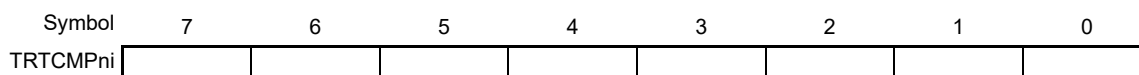
The TRTCMPni registers hold the values for comparison with the counters of the 8-bit interval timers. The TRTCMPni registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is FFH. A value from 01H to FFH can be specified^{Note 1}. These registers hold values to be compared with the values of the TRTn0 and TRTn1 registers (counters). Write access clears the counter values (TRTn0 and TRTn1) to 00H. See **11.4.4 Timing for updating compare register values** for the timing of rewriting the value for comparison.

Figure 11 - 4 Format of 8-bit Interval Timer Compare Registers ni (TRTCMPni)

Address: F0440H (TRTCMP00), F0441H (TRTCMP01), F0448H (TRTCMP10), F0449H (TRTCMP11),
F0450H (TRTCMP20), F0451H (TRTCMP21), F0458H (TRTCMP30), F0459H (TRTCMP31),
F0460H (TRTCMP40), F0461H (TRTCMP41), F0468H (TRTCMP50), F0469H (TRTCMP51),
F0470H (TRTCMP60), F0471H (TRTCMP61), F0478H (TRTCMP70), F0479H (TRTCMP71)

After reset: FFH

R/W: R/W^{Notes 1, 2}



Note 1. Setting a TRTCMPni register to 00H is prohibited.

Note 2. These registers are only accessible when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

Remark n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

11.3.4 8-bit interval timer compare register n (TRTCMPn)

The TRTCMPni registers serve as a single 16-bit comparison value register (TRTCMPn) when the 8-bit interval timer unit is used in the 16-bit interval timer mode. The TRTCMPn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is FFFFH. A value from 0001H to FFFFH can be specified^{Note 1}. This register holds a value to be compared with the value of the TRTn register (counter). Write access clears the counter value (TRTn) to 0000H. See **11.4.4 Timing for updating compare register values** for the timing of rewriting the value for comparison.

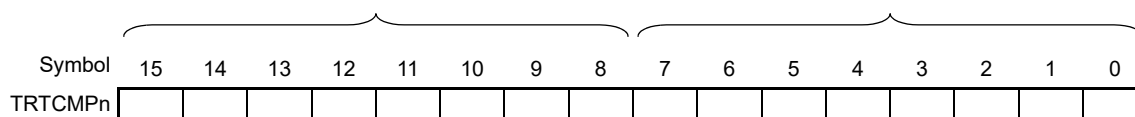
Figure 11 - 5 Format of 8-bit Interval Timer Compare Register n (TRTCMPn)

Address: F0440H (TRTCMP0), F0448H (TRTCMP1), F0450H (TRTCMP2), F0458H (TRTCMP3),
F0460H (TRTCMP4), F0468H (TRTCMP5), F0470H (TRTCMP6), F0478H (TRTCMP7)

After reset: FFFFH

R/W: R/W^{Note 2}

F0441H (TRTCMP01), F0449H (TRTCMP11),	F0440H (TRTCMP00), F0448H (TRTCMP10),
F0451H (TRTCMP21), F0459H (TRTCMP31),	F0450H (TRTCMP20), F0458H (TRTCMP30),
F0461H (TRTCMP41), F0469H (TRTCMP51),	F0460H (TRTCMP40), F0468H (TRTCMP50),
F0471H (TRTCMP61), F0479H (TRTCMP71)	F0470H (TRTCMP60), F0478H (TRTCMP70)



Note 1. Setting the TRTCMPn register to 0000H is prohibited.

Note 2. This register is only accessible when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 1.

Remark n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

11.3.5 8-bit interval timer control register n (TRTCRn)

The TRTCRn register is used to start or stop counting by the 8-bit interval timers and to select the 8-bit or 16-bit interval timer mode. The TRTCRn register can be set by a 1-bit or 8-bit manipulation instruction. The value of this register following a reset is 00H.

Figure 11 - 6 Format of 8-bit Interval Timer Control Register n (TRTCRn)

Address: F0442H (TRTCR0), F044AH (TRTCR1), F0452H (TRTCR2), F045AH (TRTCR3), F0462H (TRTCR4), F046AH (TRTCR5), F0472H (TRTCR6), F047AH (TRTCR7)

After reset: 00H

R/W: R/W^{Note 1}

Symbol	7	6	5	4	3	<2>	1	<0>
TRTCRn	TCSMDn	0	0	TCLKENn	0	TSTARTn1	0	TSTARTn0

TCSMDn	Mode select
0	Each 8-bit interval timer runs in the 8-bit interval timer mode.
1	The 8-bit interval timers run in the 16-bit interval timer mode (channels 0 and 1 are cascade-connected).
See 11.4 Operation for details.	

TCLKENn	8-bit interval timer clock enable ^{Note 2}
0	Supply of the clock signal is stopped.
1	The clock signal is supplied.

TSTARTn1	8-bit interval timer 1 count start ^{Notes 2, 3}
0	Counting stops.
1	Counting starts.
In the 8-bit interval timer mode, writing 1 to the TSTARTn1 bit starts the TRTn1 count and writing 0 stops the count. In the 16-bit interval timer mode, this bit is not used and is ineffective. See 11.4 Operation for details.	

TSTARTn0	8-bit interval timer 0 count start ^{Notes 2, 3}
0	Counting stops.
1	Counting starts.
In the 8-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn0 count and writing 0 stops the count. In the 16-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn count and writing 0 stops the count. See 11.4 Operation for details.	

Note 1. Bits 6, 5, 3, and 1 are read-only. When writing to these bits, write 0. When reading from these bits, 0 is read.

Note 2. Be sure to set the TCLKENn bit to 1 before setting an 8-bit interval timer. To stop the clock, set TSTARTn0 and TSTARTn1 to 0 and then set the TCLKENn bit to 0 after one or more cycles of the operating clock (fsx or fil) have elapsed. See **11.5.3 Setting procedure for the 8-bit interval timers** for details.

Note 3. See **11.5.1 Changing settings of operating modes** for the notes on using bits TSTARTn0, TSTARTn1, and TCSMDn.

Remark n: Unit number (n = 0 to 7)

11.3.6 8-bit interval timer frequency division register n (TRTMDn)

The TRTMDn register is used to select the frequency divisors of the source for counting by the 8-bit interval timers. The TRTMDn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 11 - 7 Format of 8-bit Interval Timer Frequency Division Register n (TRTMDn)

Address: F0443H (TRTMD0), F044BH (TRTMD1), F0453H (TRTMD2), F045BH (TRTMD3), F0463H (TRTMD4), F046BH (TRTMD5), F0473H (TRTMD6), F047BH (TRTMD7)

After reset: 00H

R/W: R/W^{Note 1}

Symbol	7	6	5	4	3	2	1	0
TRTMDn	—	TCKn1			—	TCKn0		

TCKn1			8-bit interval timer 1 frequency divisor select ^{Notes 2, 3, 4}
Bit 6	Bit 5	Bit 4	
0	0	0	f _{SP} (f _{SP} or f _L)
0	0	1	f _{SP} /2 (f _{SP} /2 or f _L /2)
0	1	0	f _{SP} /4 (f _{SP} /4 or f _L /4)
0	1	1	f _{SP} /8 (f _{SP} /8 or f _L /8)
1	0	0	f _{SP} /16 (f _{SP} /16 or f _L /16)
1	0	1	f _{SP} /32 (f _{SP} /32 or f _L /32)
1	1	0	f _{SP} /64 (f _{SP} /64 or f _L /64)
1	1	1	f _{SP} /128 (f _{SP} /128 or f _L /128)
In the 8-bit interval timer mode, TRTn1 counts using the source for counting set in TCKn1. In the 16-bit interval timer mode, these bits are not used; write 000 to them. See 11.4 Operation for details.			

TCKn0			8-bit interval timer 0 frequency divisor select ^{Notes 2, 3, 4}
Bit 2	Bit 1	Bit 0	
0	0	0	f _{SP} (f _{SP} or f _L)
0	0	1	f _{SP} /2 (f _{SP} /2 or f _L /2)
0	1	0	f _{SP} /4 (f _{SP} /4 or f _L /4)
0	1	1	f _{SP} /8 (f _{SP} /8 or f _L /8)
1	0	0	f _{SP} /16 (f _{SP} /16 or f _L /16)
1	0	1	f _{SP} /32 (f _{SP} /32 or f _L /32)
1	1	0	f _{SP} /64 (f _{SP} /64 or f _L /64)
1	1	1	f _{SP} /128 (f _{SP} /128 or f _L /128)
In the 8-bit interval timer mode, TRTn0 counts using the source for counting set in TCKn0. In the 16-bit interval timer mode, TRTn counts using the source for counting set in TCKn0. See 11.4 Operation for details.			

Note 1. Bits 7 and 3 are read-only. When writing to these bits, write 0. When reading from these bits, 0 is read.

Note 2. Do not switch the source for counting while counting is in progress. When switching the source for counting, set these bits while the TSTARTn bit in the TRTCRn register is 0 (counting stops).

Note 3. Set TCKni of the unused channel to 000B.

Note 4. Be sure to set the TCKni (i = 0, 1) bit before setting the TRTCMPni register.

Remark n: Unit number (n = 0 to 7)

11.4 Operation

11.4.1 Interval timer modes

The 8-bit interval timer unit has two interval timer modes: the 8-bit interval timer mode and 16-bit interval timer mode.

Table 11 - 2 and **Table 11 - 3** respectively list the registers and settings for use in the 8-bit interval timer mode and 16-bit interval timer mode.

Table 11 - 2 Registers and Settings Used in the 8-bit Interval Timer Mode

Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n0 (TRTn0)	b7 to b0	Used as an 8-bit counter of channel 0. The counter value can be read.
8-bit interval timer counter register n1 (TRTn1)	b7 to b0	Used as an 8-bit counter of channel 1. The counter value can be read.
8-bit interval timer compare register n0 (TRTCMPn0)	b7 to b0	Used to specify the 8-bit value for use in comparison by channel 0. Specify the value.
8-bit interval timer compare register n1 (TRTCMPn1)	b7 to b0	Used to specify the 8-bit value for use in comparison by channel 1. Specify the value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Specify whether to start or stop counting in channel 0.
	TSTARTn1	Specify whether to start or stop counting in channel 1.
	TCLKENn	Set to 1.
	TCSMDn	Set to 0.
8-bit interval timer frequency division register n (TRTMDn)	TCKn0	Select the counter clock for channel 0.
	TCKn1	Select the counter clock for channel 1.

Remark n: Unit number (n = 0 to 7)

Table 11 - 3 Registers and Settings Used in the 16-bit Interval Timer Mode

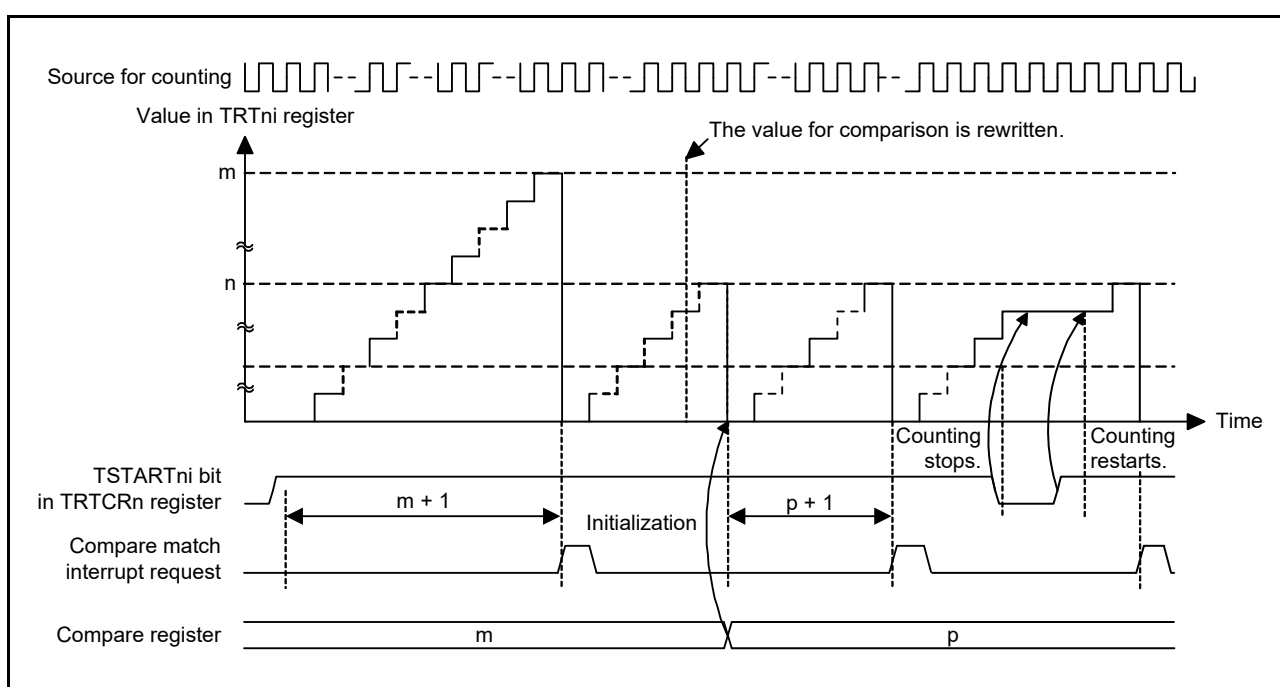
Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n (TRTn)	b15 to b0	Used as a 16-bit counter. The counter value can be read.
8-bit interval timer compare register n (TRTCMPn)	b15 to b0	Used to specify the 16-bit value for use in comparison. Specify the value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Specify whether to start or stop counting.
	TSTARTn1	Set to 0.
	TCLKENn	Set to 1.
	TCSMDn	Set to 1.
8-bit interval timer frequency division register n (TRTMDn)	TCKn0	Select the counter clock.
	TCKn1	Set to 000B.

Remark n: Unit number (n = 0 to 7)

11.4.2 Timer operation

The counter is incremented by cycles of the source selected by the TCKni ($n = 0$ to 7 ; $i = 0, 1$) bits in the frequency division register (TRTMDn). The counter value is incremented by one each time a cycle from the source is input. After the counter value reaches the value for comparison, matching of the values is detected on the input of the next cycle from the source, after which an interrupt is generated. The interrupt request takes the form of a single pulse synchronized with the source for counting. Note that, when the TSTARTni bit in the TRTCRn register is set to 0 and counting is stopped at 00H, the interrupt signal (INTITnm) will be fixed to the high level. If the interrupt signal (INTITnm) is in use as an activating source for the data transfer controller or a source of events for the ELCL in such cases, the activating source or event signal is continually generated. When operation is stopped, the counter retains the value counted immediately before it stopped. To clear the counter value, set a value for comparison in the TRTCMPni register again. After writing to the TRTCMPni register, the counter value is cleared after two cycles of the source for counting.

Figure 11 - 8 Example of Timer Operation



Remark n: Unit number ($n = 0$ to 7); i: Channel number ($i = 0, 1$); m, p: Values set in TRTCMPni register

However, the initial 00H count interval when starting count varies as follows according to the timing 1 is written in the TSTARTni ($i = 0, 1$) bit of the TRTCRn register.

- When the fsx or fil clock is selected as the source for counting:
Maximum: Two cycles of the source for counting
Minimum: One cycle of the source for counting
- When the fsx/2^m or fil/2^m clock is selected as the source for counting:
Maximum: One cycle of the source for counting
Minimum: One cycle of the selected clock (fsx or fil)

When the counter value matches the value for comparison, the counter value is cleared by the next source for counting. When the value for comparison in the TRTCMPn register is rewritten, the counter value is cleared after two cycles of the source for counting. **Table 11 - 4** lists the interrupt sources in the 8-bit and 16-bit interval timer modes.

Table 11 - 4 Interrupt Sources in the 8-bit and 16-bit Interval Timer Modes

Interrupt Name	8-bit Interval Timer Mode Source	16-bit Interval Timer Mode Source
INTITn0	Rising edge of the next source for counting after compare match of channel 0	Rising edge of the next source for counting after compare match
INTITn1	Rising edge of the next source for counting after compare match of channel 1	Not generated

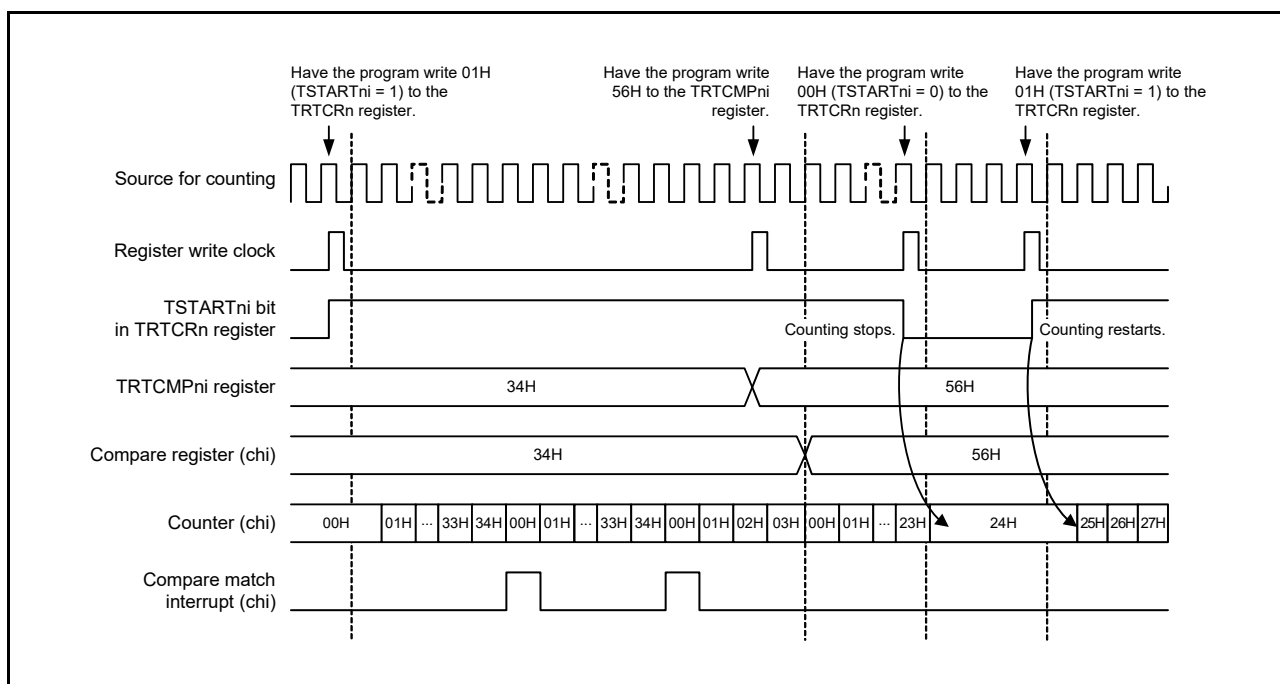
Remark n: Unit number (n = 0 to 7)

11.4.3 Start/stop timing

11.4.3.1 Timing when the fsx clock is selected as the source for counting

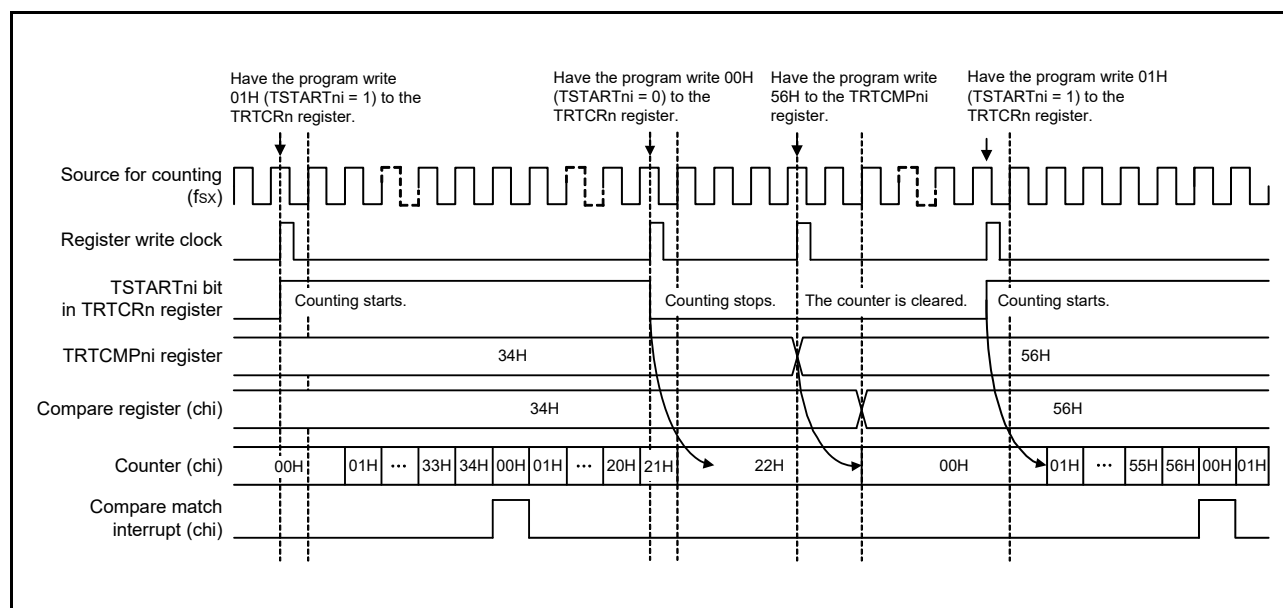
After 1 is written to the TSTART_{ni} ($n = 0$ to 7 ; $i = 0, 1$) bit in the TRTCR_n register, the count is started by the next subsystem clock X (fsx), and then the counter is incremented from 00H to 01H by the next source for counting (fsx). Likewise, after 0 is written to the TSTART_{ni} bit, the count is stopped after the counter is incremented by the subsystem clock X (fsx). **Figure 11 - 9** shows the timing for starting and stopping counting, and **Figure 11 - 10** shows the timing of a sequence of stopping counting, setting the compare register (counter clearing), and then starting counting. **Figure 11 - 9** and **Figure 11 - 10** show the timing of updating in the 8-bit interval timer mode. This timing also applies to the 16-bit interval timer mode.

Figure 11 - 9 Example of Starting and Stopping Counting When the fsx Clock is Selected



Remark n: Unit number ($n = 0$ to 7); i: Channel number ($i = 0, 1$)

Figure 11 - 10 Example of a Sequence of Stopping Counting, Setting the Compare Register (Counter Clearing), and Then Starting Counting When the fsx Clock is Selected



Remark n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

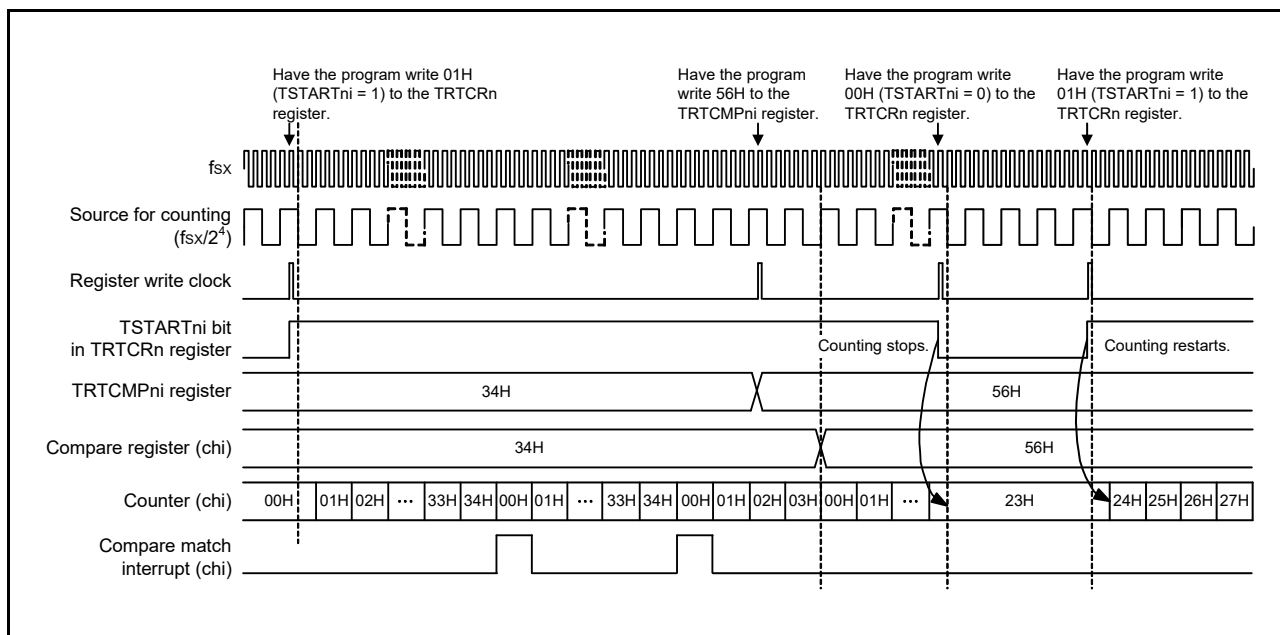
11.4.3.2 Timing when the $fsx/2^m$ clock is selected as the source for counting

After 1 is written to the TSTARTNi ($n = 0$ to 7 ; $i = 0, 1$) bit in the TRTCRn register, the count is started with the next subsystem clock X (fsx), and then the counter is incremented from 00H to 01H by the next source for counting ($fsx/2^m$). Likewise, after 0 is written to the TSTARTNi bit, the count is stopped with the subsystem clock X (fsx). However, the first period for counting from 00H when a timer starts counting is shorter than one cycle of the source for counting as shown below, to an extent that depends on the timing of writing to the TSTARTNi bit and the timing of the next cycle of the source for counting.

- Minimum: One cycle of the subsystem clock X (fsx)
- Maximum: One cycle of the source for counting

Figure 11 - 11 shows the timing for starting and stopping counting, and **Figure 11 - 12** shows the timing of a sequence of stopping counting, setting the compare register (counter clearing), and then starting counting. **Figure 11 - 11** and **Figure 11 - 12** show the timing of updating in the 8-bit interval timer mode. This timing also applies to the 16-bit interval timer mode.

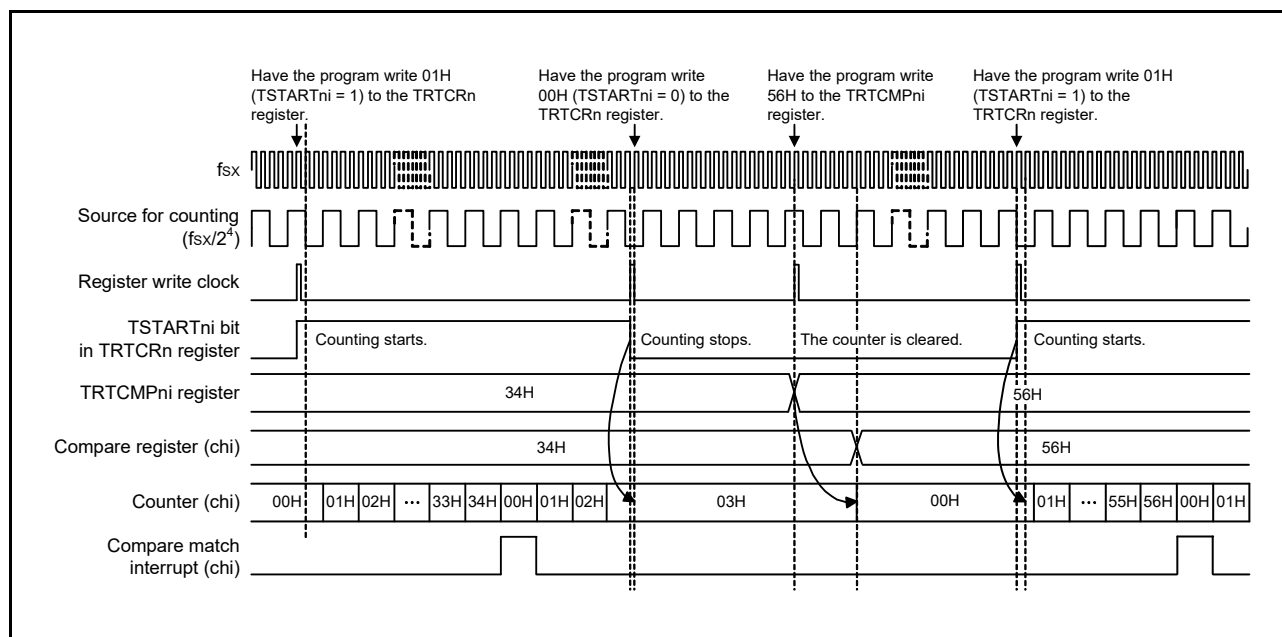
Figure 11 - 11 Example of Starting and Stopping Counting When the $fsx/2^m$ Clock is Selected



Remark 1. n: Unit number ($n = 0$ to 7); i: Channel number ($i = 0, 1$)

Remark 2. $m \neq 0$

Figure 11 - 12 Timing of a Sequence of Stopping Counting, Setting the Compare Register (Counter Clearing), and Then Starting Counting When the $fsx/2^m$ Clock is Selected



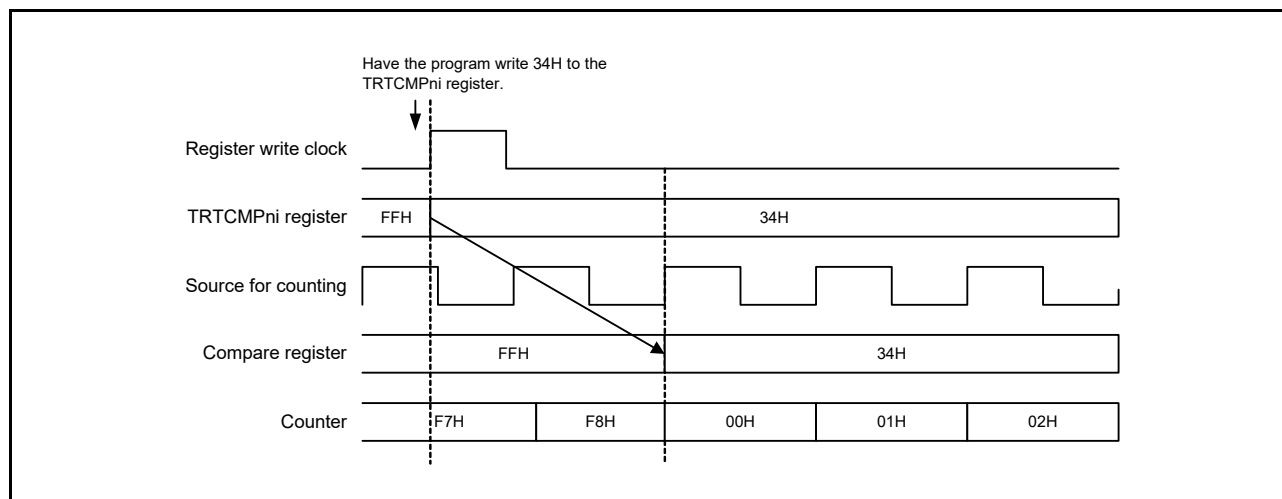
Remark 1. n: Unit number (n = 0 to 7); i: Channel number (i = 0, 1)

Remark 2. m ≠ 0

11.4.4 Timing for updating compare register values

The timing for updating the value of the TRTCMPn*i* (*n* = 0 to 7; *i* = 0, 1) register is the same, regardless of the value of the TSTARTn*i* bit in the TRTCRn register. After write access to a TRTCMPn*i* register, the value is stored in that register after two cycles of the source for counting. When stored in the compare register, the counter value is cleared and set (the 8-bit interval timer mode: 00H, the 16-bit interval timer mode: 0000H). **Figure 11 - 13** shows the timing of rewriting. This figure shows the timing of updating in the 8-bit interval timer mode. This timing also applies to the 16-bit interval timer mode.

Figure 11 - 13 Timing of Rewriting the Value for Comparison

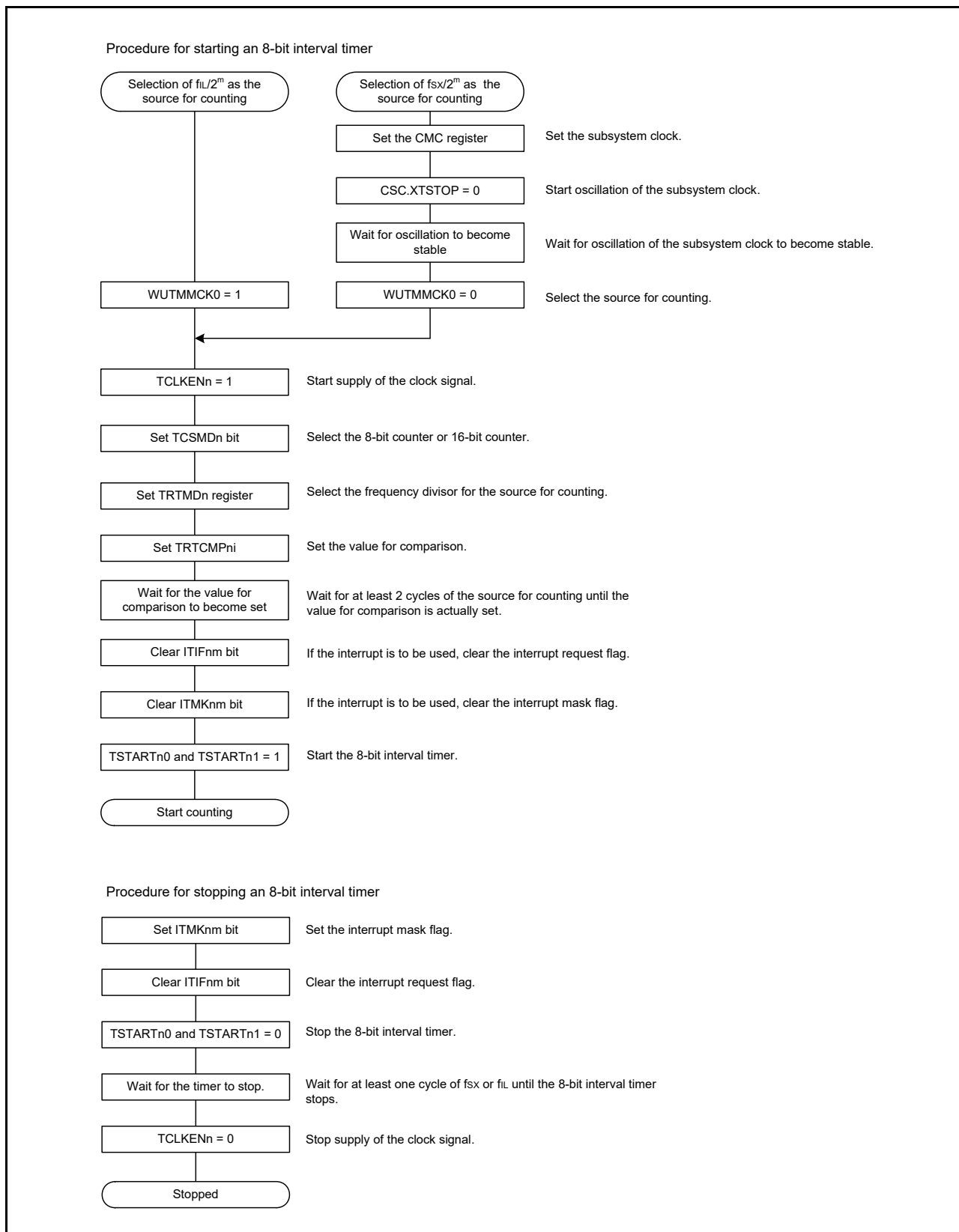


Remark n: Unit number (*n* = 0 to 7); *i*: Channel number (*i* = 0, 1)

11.4.5 Procedures for Setting an 8-bit Interval Timer

The procedures for starting and stopping an 8-bit interval timer are shown below.

Figure 11 - 14 Procedures for Starting and Stopping an 8-bit Interval Timer



11.5 Notes on 8-bit Interval Timers

11.5.1 Changing settings of operating modes

The settings of bits TCSMDn and TCKni ($n = 0$ to 7 ; $i = 0, 1$) must be changed while the TSTARTni bit in the TRTCRn register is 0 (counting stops). After the value of the TSTARTni bit is rewritten from 1 to 0 (counting stops), allow at least one cycle of fsx or fil to elapse before accessing the registers (TRTCRn and TRTMDn) associated with the 8-bit interval timers.

11.5.2 Accessing compare registers

Do not write to the compare registers (TRTCMPn0, TRTCMPn1, and TRTCMPn) successively without intervals. When writing to them successively, allow at least two cycles of the source for counting between one write access and the next. Writing to a compare register (TRTCMPn0, TRTCMPn1, or TRTCMPn) must proceed while the source to drive counting is made to oscillate by setting the 8-bit interval timer clock enable bit (TCLKENn) to 1.

11.5.3 Setting procedure for the 8-bit interval timers

To supply the clock, set the 8-bit interval timer clock enable bit (TCLKENn) in the 8-bit interval timer control register (TRTCRn) to 1 and then set the TSTARTni bit. Do not set bits TCLKENn and TSTARTni at the same time. To stop the clock, set the TSTARTni bit to 0 and then allow at least one cycle of fsx or fil to elapse before setting the TCLKENn bit to 0.

Remark n: Unit number ($n = 0$ to 7); i: Channel number ($i = 0, 1$)

Section 12 32-bit Interval Timer (TML32)

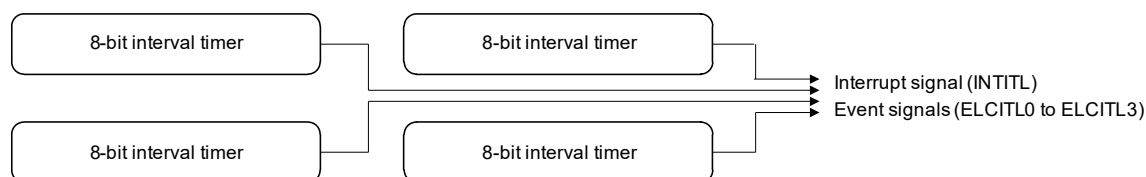
The 32-bit interval timer has four 8-bit interval timers (referred to as channels 0 to 3). Each is capable of running independently and in that case they all have the same functions. Two 8-bit interval timers can also be cascade-connected to run as a single 16-bit interval timer. In addition, all four 8-bit interval timers can be cascade-connected to run as a single 32-bit interval timer.

12.1 Overview

The 32-bit interval timer is driven by the fMXP, fSXP, fIHP, or fIMP clock signal or event input from the ELCL, all of which are asynchronous with respect to the CPU. The 32-bit interval timer has the following functions.

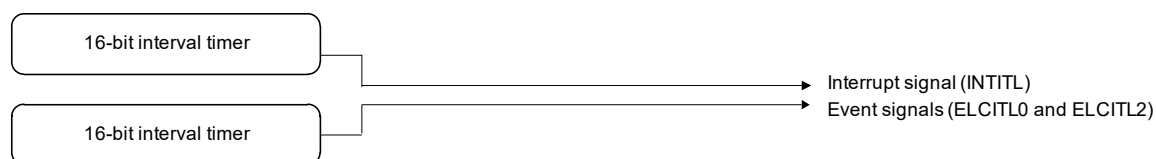
1. 8-bit counter mode

The 8-bit timers are usable as four 8-bit interval timers that generate interrupts (INTITL) and event signal (ELCITL_n) at fixed intervals.



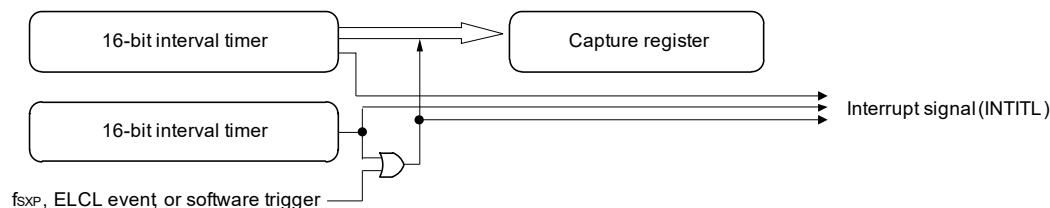
2. 16-bit counter mode

The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (INTITL) and event signal (ELCITL_n) at fixed intervals.



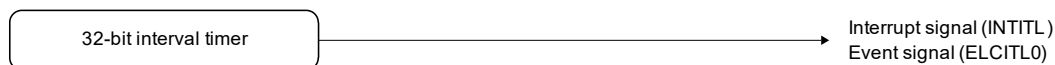
3. 16-bit capture mode

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 (ELCITL_n) at fixed intervals.



Remark n: Channel number (0 to 3)

Table 12 - 1 lists the specifications of and **Figure 12 - 1** is a block diagram of the 32-bit interval timer.

Table 12 - 1 Specifications of the 32-bit Interval Timer

Item	Description
Source for counting (operating clock)	• fMXP • fSXP • fIHP • fIMP • Event input from the ELCL
Capture clock (Selectable sources for counting by the timer which can generate a capture trigger)	• fMXP • fSXP • fIHP • fIMP • Event input from the ELCL
Frequency divisor	• 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128
Operating mode	• 8-bit counter mode Each of channels 0 to 3 runs as an independent 8-bit counter. • 16-bit counter mode The counters of channels 0 and 1 are cascade-connected, and the counters of channels 2 and 3 are cascade-connected, to run as two 16-bit counters. • 32-bit counter mode Channels 0 to 3 are all cascade-connected to run as a 32-bit counter. • 16-bit capture mode Channels 0 and 1 are cascade-connected to run as a 16-bit counter driven by the source for counting, channels 2 and 3 are cascade-connected to run as a 16-bit counter driven by the capture clock, and the cascade-connected counters are used for capture operations.
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 value for comparison. – Output when the capturing of the counter value is completed in the capture mode.
Event signals	• Four trigger signals ELCITL0 to ELCITL3 for the ELCL, A/D converter, and LCD controller/driver are output. – Output when the counter value in any of channels 0 to 3 matches the value for comparison.

Remark fMXP: High-speed peripheral clock frequency

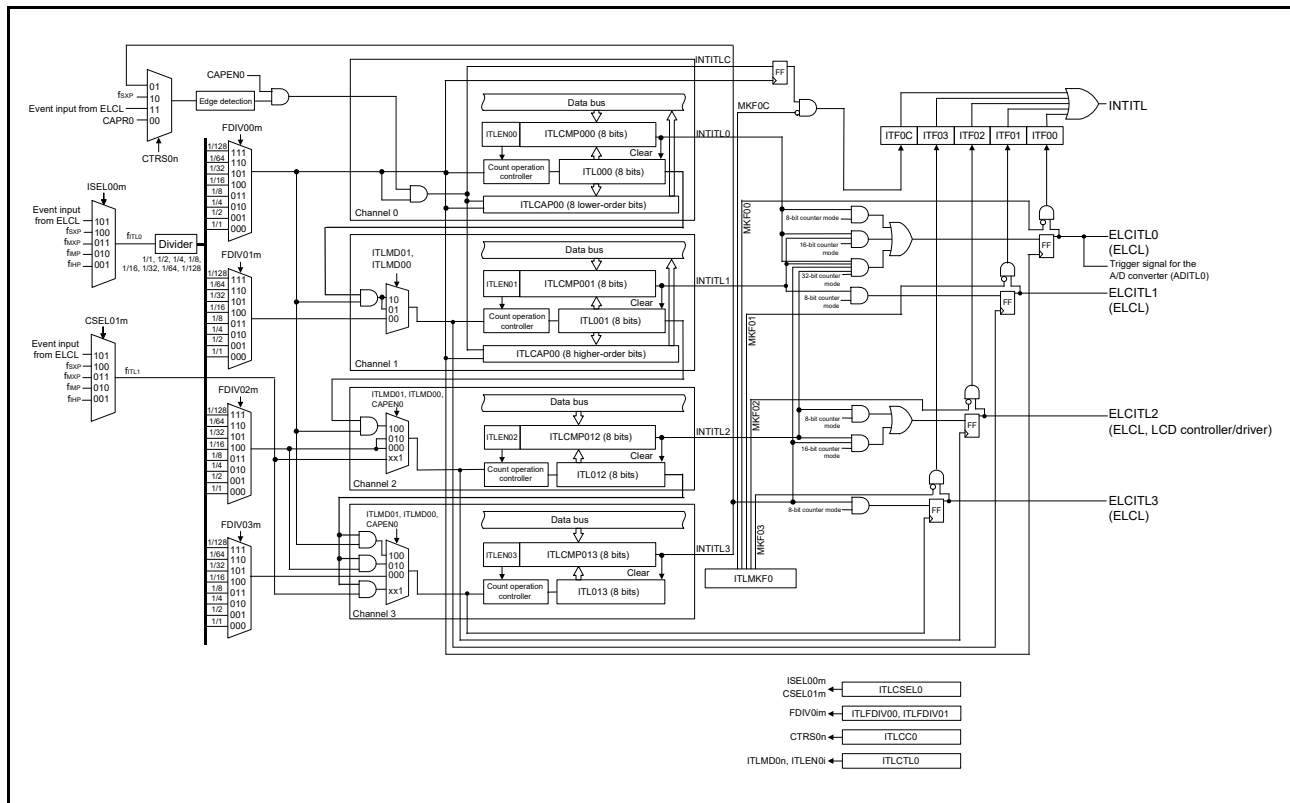
fSXP: Low-speed peripheral clock frequency

fIHP: High-speed on-chip oscillator peripheral clock frequency

fIMP: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)

Event input from the ELCL: The ELCL 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 24 Logic and Event Link Controller (ELCL)**.

Figure 12 - 1 Block Diagram of the 32-bit Interval Timer

**Remark 1.** ITL000, ITL001, ITL012, ITL013: 8-bit counters

In the 16-bit counter mode, the counters in channels 0 and 1 are cascade-connected (ITL000 + ITL001) and the counters in channels 2 and 3 are cascade-connected (ITL012 + ITL013). In the 32-bit counter mode, the counters in channels 0 to 3 are cascade-connected (ITL000 + ITL001 + ITL012 + ITL013).

ISEL00m, CSEL01m: Bits in the interval timer clock select register (ITLCSEL0)

FDIV00m: Bits in the interval timer frequency division register (ITLFDIV0n)

CTR00n: Bits in the interval timer capture control register (ITLCC0)

ITLMD0n, ITLEN0i: Bits in the interval timer control register (ITLCTL0)

Remark 2. n = 0, 1; m = 0 to 2; i = 0 to 3

12.2 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 register 0n (ITLCMP0n) (n = 0, 1)
- Interval timer capture register 00 (ITLCAP00)
- 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)

12.2.1 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 following a reset is 00H.

Figure 12 - 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	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	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 Be sure to clear bit 1 to 0.

12.2.2 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) 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 following a reset is 00H.

Figure 12 - 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	DACRES	SMSRES	CMPRES	TML32RES	0	0	0	CTSURES
TML32RES		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.						

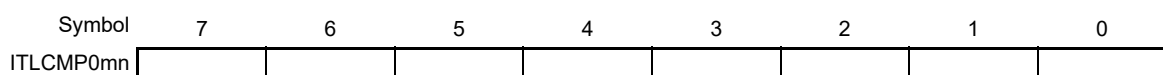
Caution Be sure to clear bits 3 to 1 to 0.

12.2.3 Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13)

The ITLCMP0mn registers hold the values for comparison when the interval timer is used in the 8-bit counter mode. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is FFH. A value from 01H to FFH can be specified. Setting a register to 00H is prohibited. These registers hold values to be compared with the values of the ITL000 to ITL013 counters.

Figure 12 - 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}



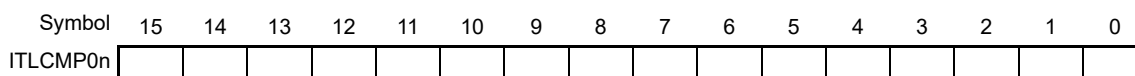
Note Write to the ITLCMP000 to ITLCMP013 registers while the settings of the ITLEN00 to ITLEN03 bits in the ITLCTL0 register are 0, respectively.

12.2.4 Interval timer compare register 0n (ITLCMP0n) (n = 0, 1)

The ITLCMP0mn registers serve as a single 16-bit comparison value register (ITLCMP0n) when the interval timer is used in the 16-bit or 32-bit counter mode. This register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is FFFFH. A value from 0001H to FFFFH can be specified. Setting this register to 0000H is prohibited. This register holds a value to be compared with the value of the ITL0n counter. When the settings of the ITLMD01 and ITLMD00 bits in the ITLCTL0 register are 1 and 0, respectively, this register is used as a compare register in the 32-bit counter mode; specify the 16 higher-order bits of the value for comparison in the ITLCMP01 register and the 16 lower-order bits of the value for comparison in the ITLCMP00 register.

Figure 12 - 5 Format of Interval Timer Compare Register 0n (ITLCMP0n)

Address: F0360H, F0361H (ITLCMP00), F0362H, F0363H (ITLCMP01)
 After reset: FFFFH
 R/W: R/W^{Note}



Note Write to the ITLCMP00 register while the ITLEN00 bit in the ITLCTL0 register is 0. Write to the ITLCMP01 register while the ITLEN02 bit in the ITLCTL0 register is 0 in the 16-bit counter mode or while the ITLEN00 bit in the ITLCTL0 register is 0 in the 32-bit counter mode.

12.2.5 Interval timer capture register 00 (ITLCAP00)

This register holds 16-bit captured values when the interval timers are running in the 16-bit counter mode. The ITLCAP00 register can be read by a 16-bit manipulation instruction. The value of this register following a reset is 0000H. 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 value for comparison in the ITLCMP01 register.

Figure 12 - 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																

12.2.6 Interval timer control register (ITLCTL0)

The ITLCTL0 register is used to start or stop counting by the interval timer and to select the 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 following a reset is 00H.

Figure 12 - 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 runs in the 8-bit counter mode.
0	1	The interval timer runs in the 16-bit counter mode (channel 0 is cascade-connected with channel 1 and channel 2 is cascade-connected with channel 3).
1	0	The interval timer runs in the 32-bit counter mode (channels 0 to 3 are all cascade-connected).
1	1	Setting prohibited.

ITLEN03	8-bit counter mode: ITL013 count enable ^{Note 2}
0	Counting stops.
1	Counting starts.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL013 counter and writing 0 stops it. In the 16-bit counter mode, set this bit to 0. In the 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 starts.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 counter and writing 0 stops it. In the 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 + ITL013 counter and writing 0 stops it. In the 32-bit counter mode, set this bit to 0.	

ITLEN01	8-bit counter mode: ITL001 count enable ^{Note 2}
0	Counting stops.
1	Counting starts.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL001 counter and writing 0 stops it. In the 16-bit counter mode, set this bit to 0. In the 32-bit counter mode, set this bit to 0.	

Figure 12 - 7 Format of Interval Timer Control Register (ITLCTL0) (2/2)

ITLEN00	8-bit counter mode: ITL000 count enable ^{Note 2} 16-bit counter mode: ITL000 + ITL001 count enable ^{Note 2} 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enable ^{Note 2}
0	Counting stops.
1	Counting starts.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 counter and writing 0 stops it. In the 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 counter and writing 0 stops it. In the 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 counter mode, be sure to only write to the ITLMD00 and ITLMD01 bits while the ITLEN03 to ITLEN00 bits are all 0.

Note 2. When any 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 counter mode ^{Note 1}	0	0				✓	ITL000
					✓		ITL001
				✓			ITL012
			✓				ITL013
16-bit counter mode ^{Note 2}	0	1	Always set to 0.		Always set to 0.	✓	ITL000 + ITL001
			Always set to 0.	✓	Always set to 0.		ITL012 + ITL013
32-bit counter mode	1	0	Always set to 0.	Always set to 0.	Always set to 0.	✓	ITL000 + ITL001 + ITL012 + ITL013

Note 1. Two or more bits of ITLEN03 to ITLEN00 can be set to 1 or 0 at the same time.

Note 2. The ITLEN02 and ITLEN00 bits can be set to 1 or 0 at the same time.

Remark ✓: Enables counting in the target counter.

12.2.7 Interval timer clock select register 0 (ITLCSEL0)

The ITLCSEL0 register is used to select the source for counting by the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 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 1}
0	0	0	Counting stops.
0	0	1	fIHP ^{Note 2}
0	1	0	fIMP
0	1	1	fMXP
1	0	0	fSXP
1	0	1	Event input from the ELCL
Others			Setting prohibited.

ISEL002	ISEL001	ISEL000	Selection of interval timer count clock (fITL0) ^{Note 1}
0	0	0	Counting stops.
0	0	1	fIHP ^{Note 3}
0	1	0	fIMP
0	1	1	fMXP
1	0	0	fSXP
1	0	1	Event input from the ELCL
Others			Setting prohibited.

Note 1. Be sure to only write to the CSEL012 to CSEL010 bits and ISEL002 to ISEL000 bits while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.

Note 2. If the FRQSEL4 bit in the option bytes is set to 1, this cannot be used.

Note 3. If the FRQSEL4 bit in the option bytes is set to 1, set the interval timer counter clock to a frequency no greater than fITL0/2 by setting a value to obtain a frequency in this range in the given interval timer frequency division register n (ITLFDIV0n).

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
Event input from the ELCL: The ELCL 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 24 Logic and Event Link Controller (ELCL)**.

12.2.8 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 following a reset is 00H.

Figure 12 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00)

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 the 8-bit counter mode, ITL001 counts cycles of the counter clock specified in the FDIV012 to FDIV010 bits. In the 16-bit counter mode, set these bits to 000B. In the 32-bit counter mode, set these bits to 000B.

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 the 8-bit counter mode, ITL000 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In the 16-bit counter mode, ITL000 + ITL001 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In the 32-bit counter mode, ITL000 + ITL001 + ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits.

Note 1. In the 8-bit counter mode, be sure to only write to the FDIV012 to FDIV010 bits while the ITLEN01 bit in the ITLCTL0 register is 0.

Note 2. Be sure to only write to the FDIV002 to FDIV000 bits while the ITLEN00 bit in the ITLCTL0 register is 0.

12.2.9 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 following a reset is 00H.

Figure 12 - 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 the 8-bit counter mode, ITL013 counts cycles of the counter clock specified in the FDIV032 to FDIV030 bits. In the 16-bit counter mode, set these bits to 000B. In the 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 the 8-bit counter mode, ITL012 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In the 16-bit counter mode, ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In the 32-bit counter mode, these bits are not used; write 000B to them.			

Note 1. In the 8-bit counter mode, be sure to only write to the FDIV032 to FDIV030 bits while the ITLEN03 bit in the ITLCTL0 register is 0.

Note 2. In the 8-bit or 16-bit counter mode, be sure to only write to the FDIV022 to FDIV020 bits while the ITLEN02 bit in the ITLCTL0 register is 0.

12.2.10 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 following a reset is 00H.

Figure 12 - 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 generation of the capture trigger selected in the CTRS01 and CTRS00 bits and capture of the data in the ITLCAP00 register. Writing 1 to the CAPF0CR bit clears this flag to 0.								
CAPR0								
Software capture trigger ^{Notes 4, 5}								
0								
Trigger operation does not proceed.								
1								
A software trigger for capturing is generated.								
CAPC0CR								
Selection of the 16-bit counter (ITL000 + ITL001) clearing after capturing ^{Note 6}								
0								
The 16-bit counter (ITL000 + ITL001) is retained after the completion of capturing.								
1								
The 16-bit counter (ITL000 + ITL001) is cleared after the completion of capturing.								
CTRS01	CTRS00	Selection of capture trigger ^{Notes 5, 7}						
0	0	Software trigger						
0	1	Interrupt on compare match with ITLCMP01 ^{Note 8}						
1	0	fsXP (rising edge)						
1	1	Event input from ELCL (rising edge)						

Note 1. Be sure to only write to the CAPEN0 bit while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register 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. In the capture operation, the interval at which the capture trigger is generated should be two or more cycles of the counter clock.

- Note 6.** Be sure to only write to the CAPC0CR bit while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.
- Note 7.** Be sure to only write to the CTRS01 and CTRS00 bits while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.
- Note 8.** When the interrupt on compare match with ITLCMP01 is selected as a capture trigger, the compare match detection flag for channel 2 (ITF02) and capture detection flag (ITF0C) are set on capture of the counter value. When only using the capture detection flag, set the ITLMKF0 register to mask the compare match detection flag for channel 2.

12.2.11 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 following a reset is 00H. 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 12 - 2** lists the conditions for setting the status flags in each counter mode selected by the ITLMD01 and ITLMD00 bits.

Figure 12 - 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 to 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 12 - 2 Conditions for Setting the Status Flags in Each Counter Mode

Mode	ITLMD01	ITLMD00	CAPEN0	Status Flag	Conditions for the Setting Status Flag
8-bit counter mode	0	0	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP000 and ITL000 values
			×	ITF01	The next rising edge of the counter clock following a match between the ITLCMP001 and ITL001 values
			×	ITF02	The next rising edge of the counter clock following a match between the ITLCMP012 and ITL012 values
			×	ITF03	The next rising edge of the counter clock following a match between the ITLCMP013 and ITL013 values
16-bit counter mode	0	1	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 and ITL000 + ITL001 values
			×	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 counter 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

Remark ×: Don't care

12.2.12 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 following a reset is 00H. Setting an MKF0C or MKF0i (i = 0 to 3) bit to 1 masks the corresponding status flag among ITLF0C and ITLF0i (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 12 - 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.

12.3 Operation

12.3.1 Settings for counter modes

The 32-bit interval timer has three counter modes: the 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode. **Table 12 - 3** to **Table 12 - 5** respectively list the registers and settings for use in the 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode.

Table 12 - 3 Registers and Settings Used in the 8-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare registers 0mn (ITLCMP0mn)	Bits 7 to 0	Specify respective 8-bit values for comparison with the counter values of 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 register n (ITLFDIV0n)	FDIV000 to FDIV002	Select the counter clock for channel 0.
	FDIV010 to FDIV012	Select the counter clock for channel 1.
	FDIV020 to FDIV022	Select the counter clock for channel 2.
	FDIV030 to FDIV032	Select the counter clock for channel 3.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the counter 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 12 - 4 Registers and Settings Used in the 16-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare register 0n (ITLCMP0n)	Bits 15 to 0	Specify respective 16-bit values for comparison with the counter value of channels 0 and 1, and with that of 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 register n (ITLFDIV0n)	FDIV000 to FDIV002	Select the counter clock for channels 0 and 1.
	FDIV010 to FDIV012	Set to 000B.
	FDIV020 to FDIV022	Select the counter clock for channels 2 and 3.
	FDIV030 to FDIV032	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL000 to ISEL002	Select the counter 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 12 - 5 Registers and Settings Used in the 32-bit Counter Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare register 0n (ITLCMP0n)	Bits 15 to 0	Specify a value for comparison in the 32-bit counter mode. Specify the 16 lower-order bits of the value for comparison in the register for channels 0 and 1 (ITLCMP00) and the 16 higher-order bits of the value for comparison in the register for 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 register n (ITLFDIV0n)	FDIV000 to FDIV002	Select the counter 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 counter 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

12.3.2 Settings for the capture mode

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 12 - 6 Registers and Settings Used in the 16-bit Capture Mode

Register Name (Symbol)	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	Bits 15 to 0	Specify the 8-bit value for comparison with the counter value of channels 0 and 1.
Interval timer compare register 01 (ITLCMP01) ^{Note}	Bits 15 to 0	Specify the 8-bit value for comparison with the counter value of 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 register n (ITLFDIV0n)	FDIV000 to FDIV002	Select the counter 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 counter clock for the interval timer in channels 0 and 1.
	CSEL010 to CSEL012	Select the counter 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.

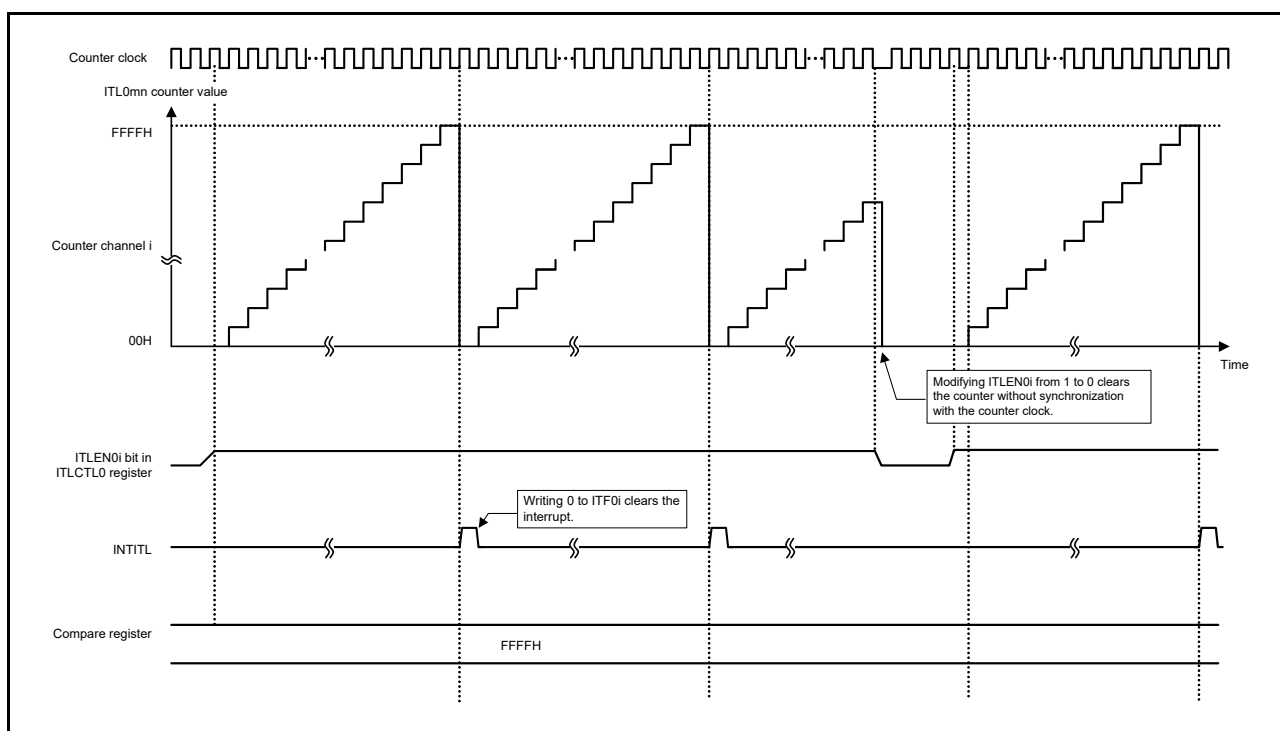
Note Channels 2 and 3 can only be used in the 16-bit counter mode when an interrupt on compare match with ITLCMP01 is not to be used as the capture trigger.

Remark n = 0, 1

12.3.3 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 value for comparison. 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 12 - 14 Example of Timer Operation



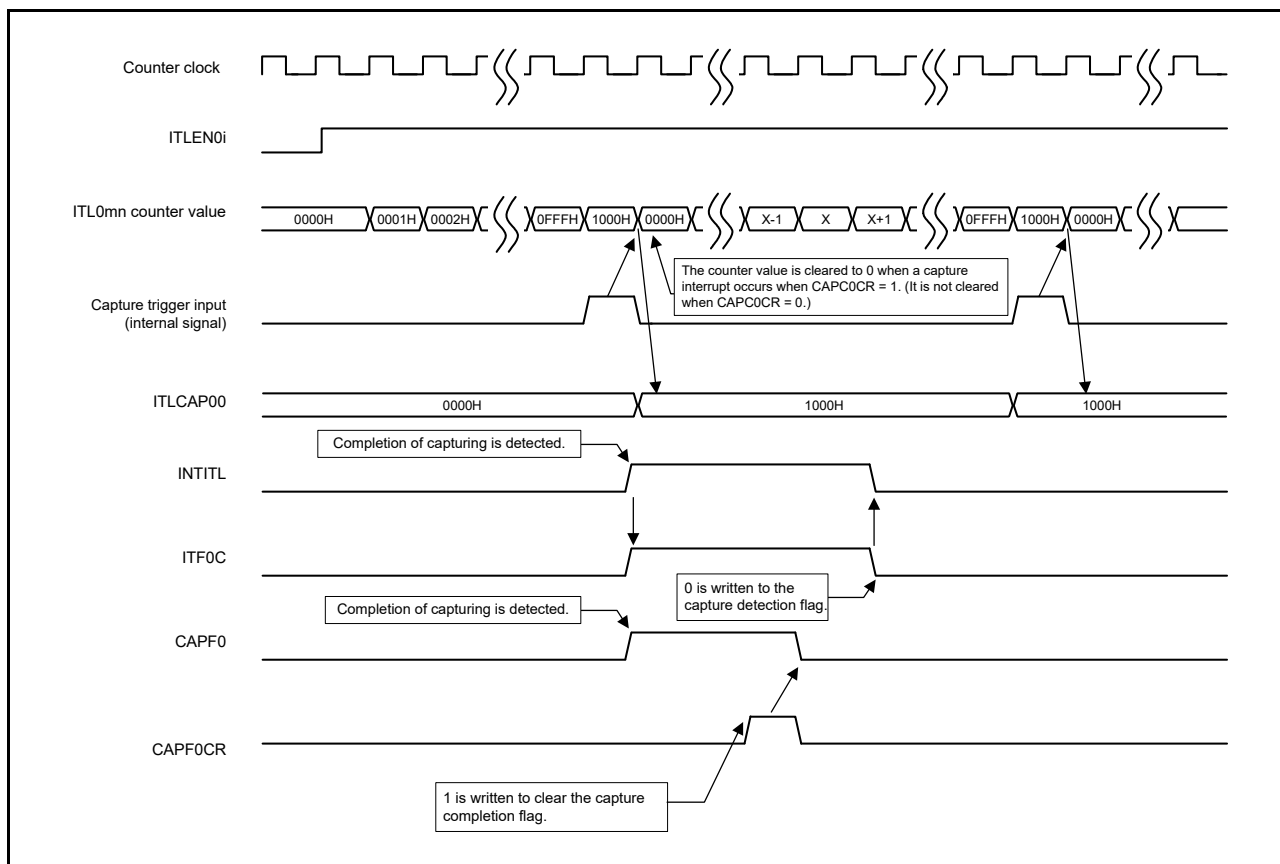
Remark mn = 00, 01, 12, 13; i = 0 to 3

12.3.4 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 ELCL, 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 (ITLSEL0) to select the clock for counting, and set interval timer compare register 01 (ITLCMP01) to specify the value for comparison. When using fsXP, an event signal input from the ELCL, or a software trigger (setting the CAPR0 bit to 1) as a capture trigger, channels 2 and 3 can be used in the 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 12 - 15 Example of Capture Operation



Remark mn = 00, 01, 12, 13; i = 0 to 3

When the counter value matches the value for comparison while the CAPC0CR bit in the ITLCC0 register is set to 1 (the mode where the 16-bit counter (ITL000 + ITL001) is cleared following the completion of capture), counting of the next clock cycle clears the counter value. Note that the ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the value for comparison before a capture trigger is input. The counter value is not cleared in this way if the CAPC0CR bit is set to 0 (the mode where the 16-bit counter (ITL000 + ITL001) retains its value after the completion of capture). The ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the value for comparison.

12.3.5 Interrupt

Table 12 - 7 lists the interrupt sources in the 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 signal (INTITL) is output.

Table 12 - 7 Interrupt Sources in the 8-bit, 16-bit, and 32-bit Counter Modes

Interrupt Source	Interrupt Condition in the 8-bit Counter Mode	Interrupt Condition in the 16-bit Counter Mode	Interrupt Condition in the 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 12 - 16** shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

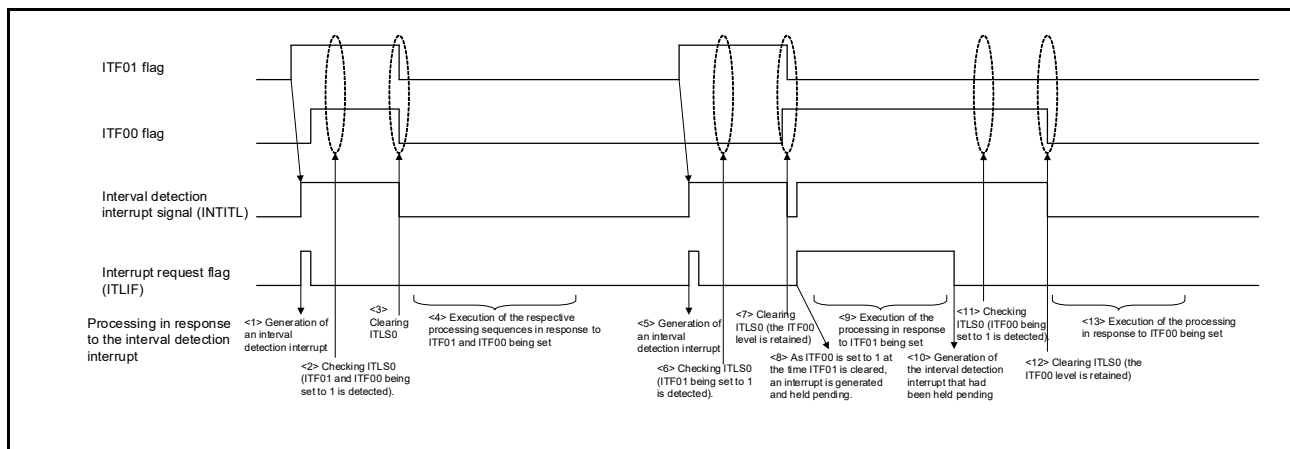
The following describes the operation shown in **Figure 12 - 16**. When a compare match in channel 1 is detected while the value of the ITLS0 register is 00H, the ITF01 flag is set to 1 and the interval detection interrupt signal (INTITL) is driven high. While the interval detection interrupt signal (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 to 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 12 - 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 12 - 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 12 - 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 12 - 16 Example of Clearing the Detected Flags



12.3.6 Procedures for Setting the Interval Timer

The procedures for setting the 32-bit interval timer are shown below.

Figure 12 - 17 Procedure for Starting the 32-bit Interval Timer

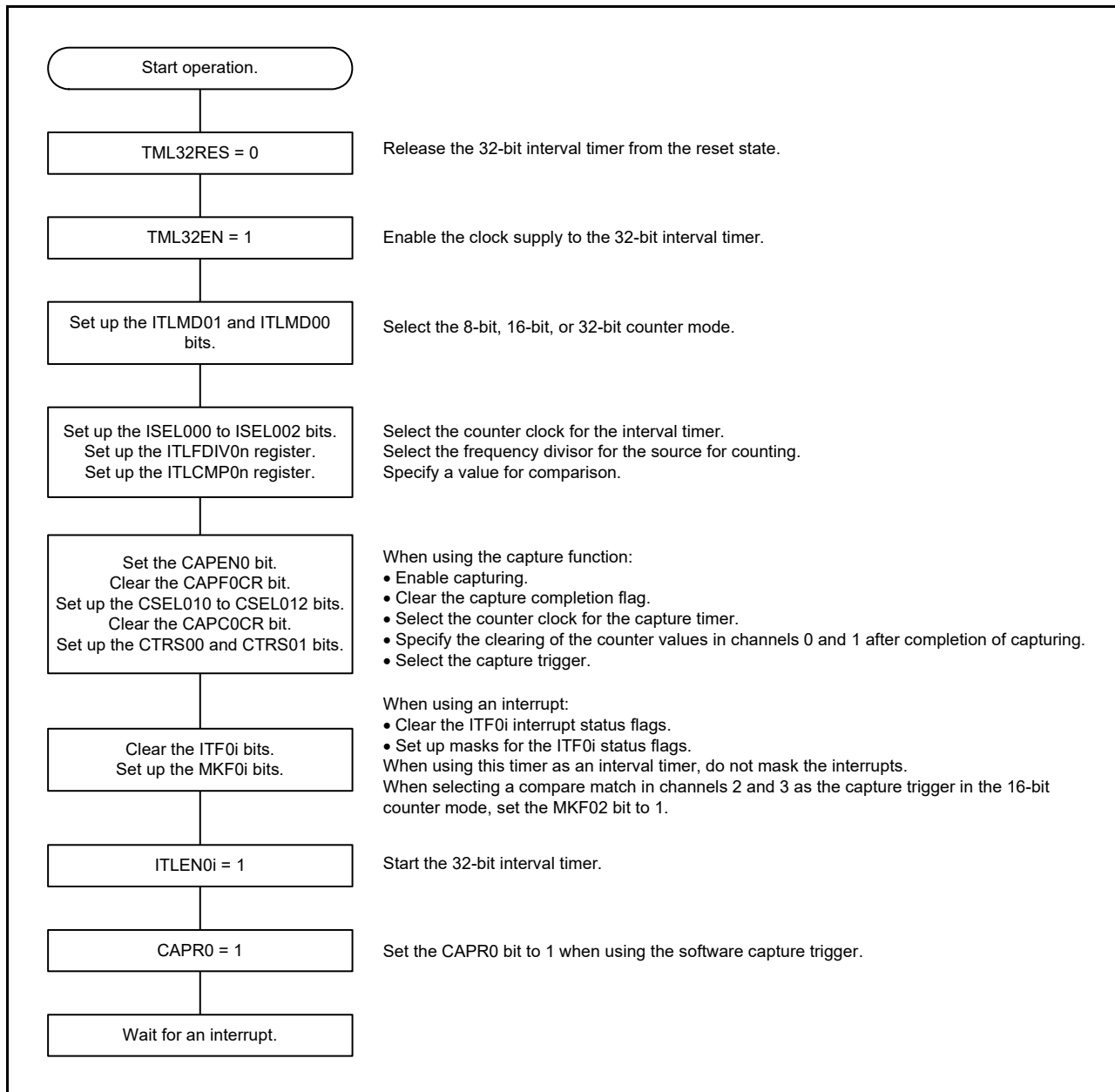


Figure 12 - 18 Procedure for Stopping the 32-bit Interval Timer

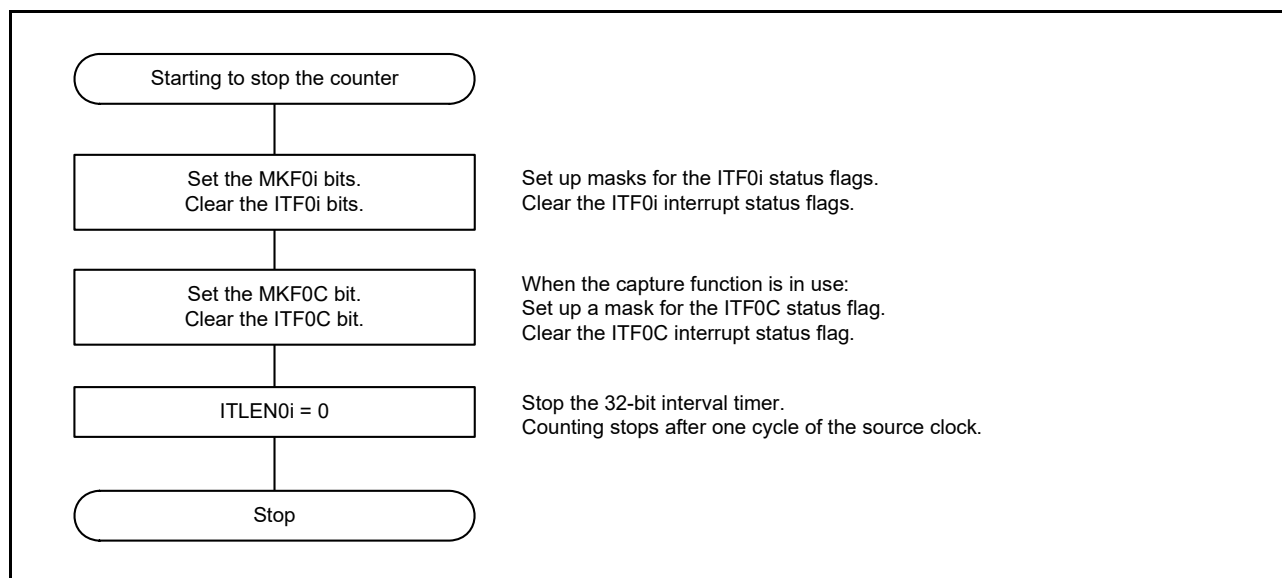


Figure 12 - 19 Procedure for Changing the Operating Mode of the 32-bit Interval Timer

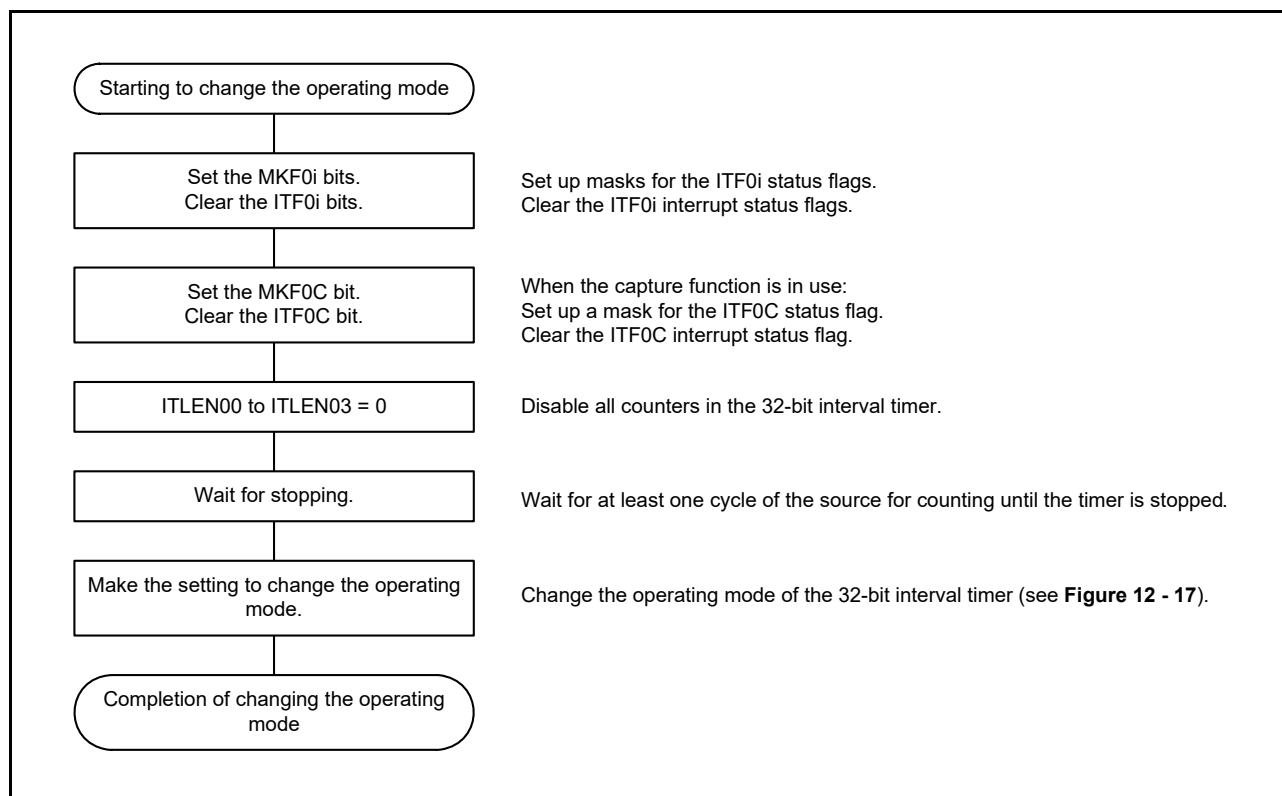


Figure 12 - 20 Procedure for Resetting the 32-bit Interval Timer

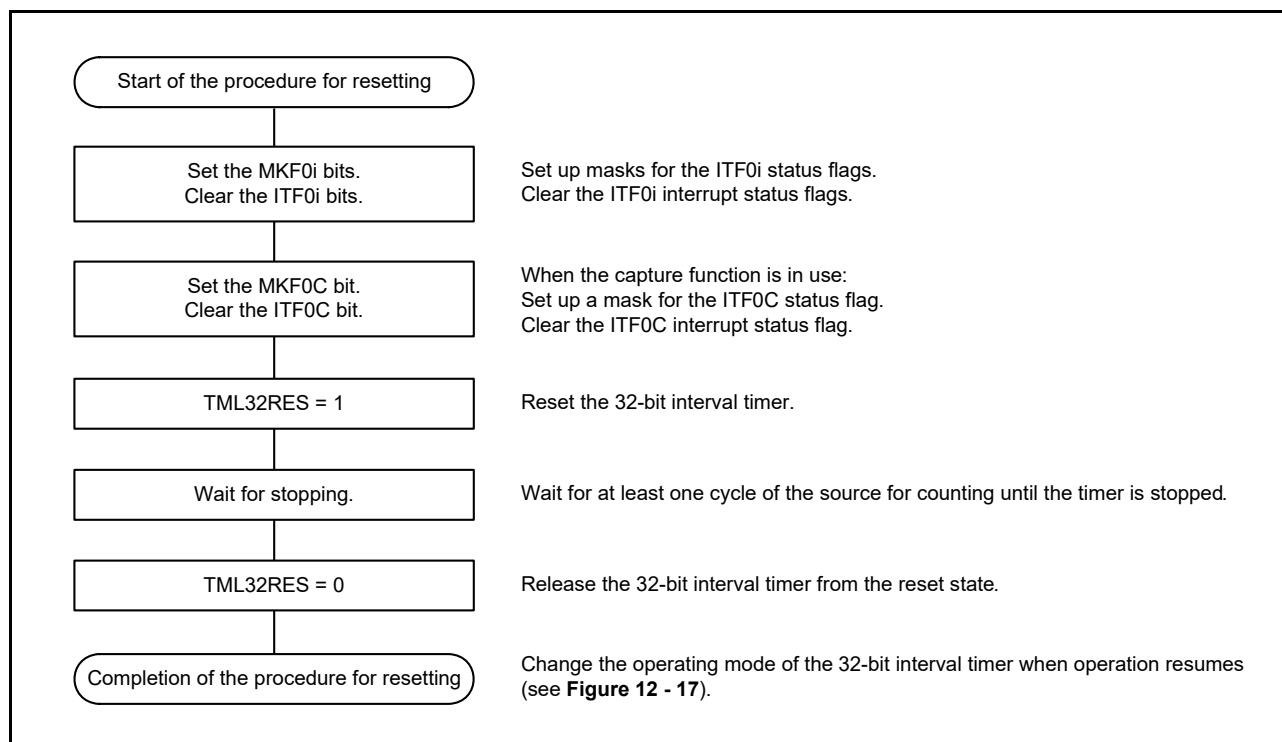


Figure 12 - 21 Procedure for Starting Event Input from the ELCL

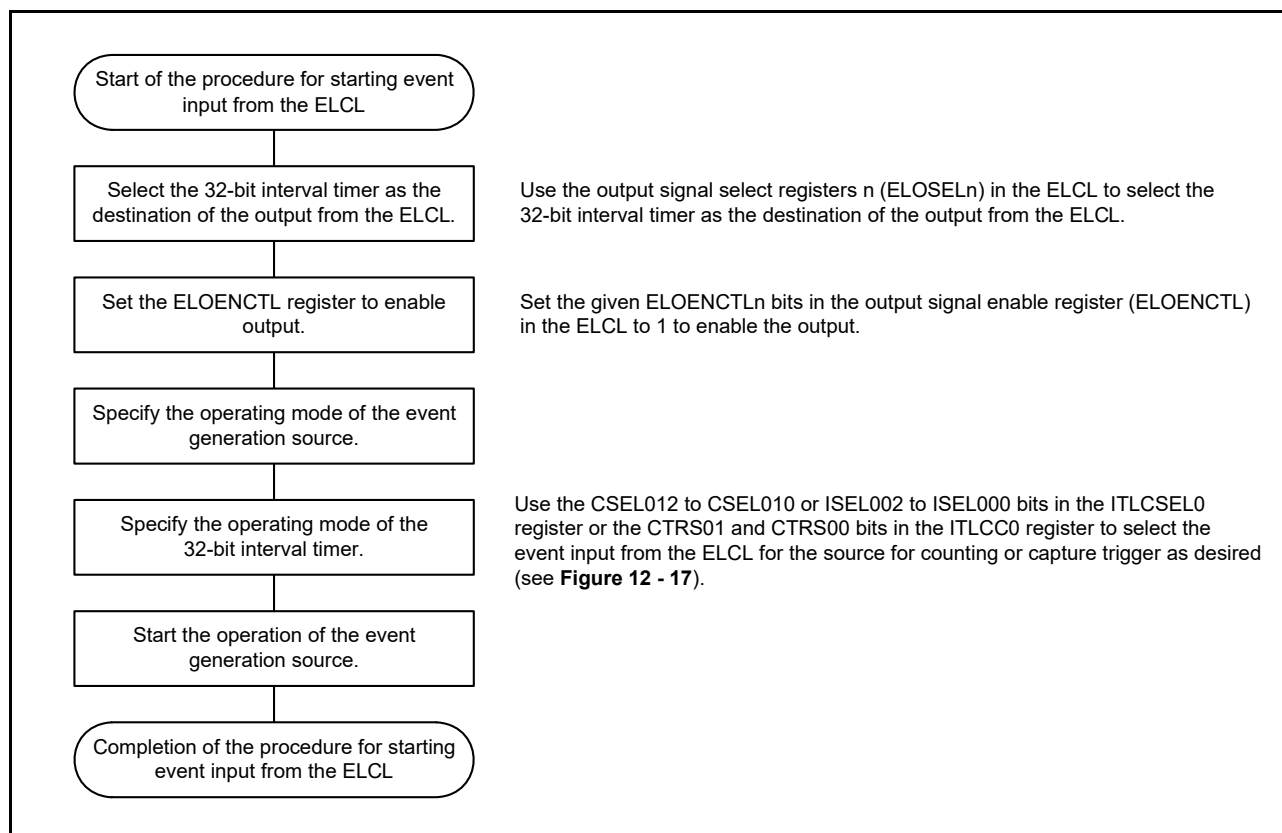
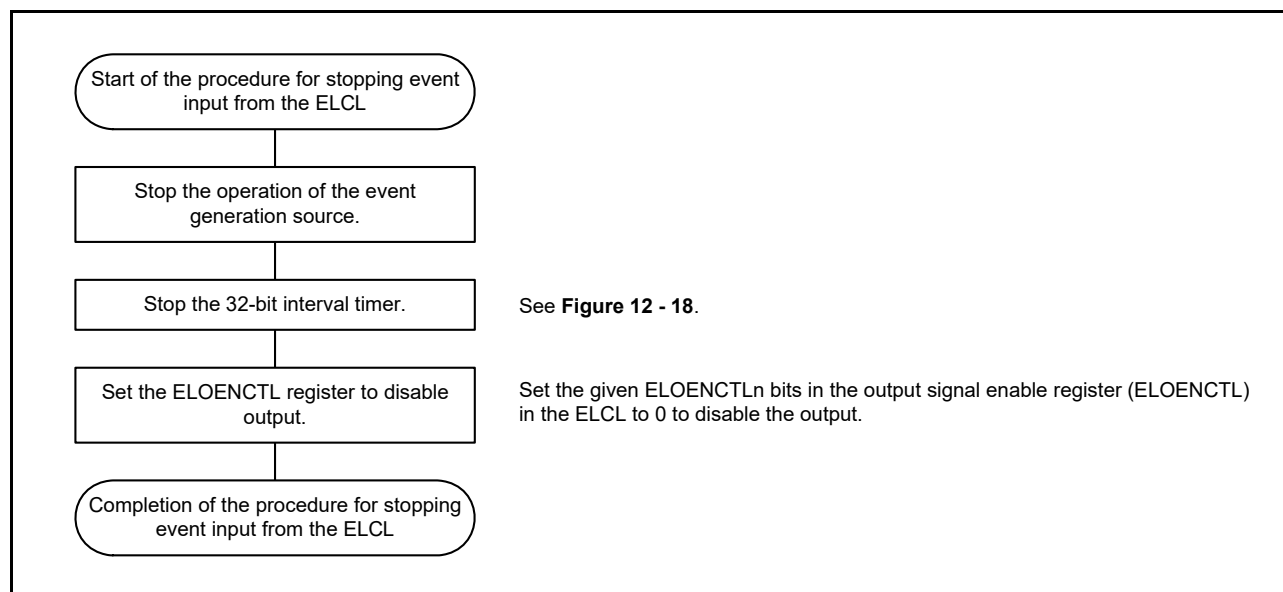


Figure 12 - 22 Procedure for Stopping Event Input from the ELCL



Section 13 External Signal Sampler

13.1 Functions of External Signal Sampler

The external signal sampler has the following features.

- Generates a sampling clock signal based on the timing of generation of the 8-bit interval timer 00 interrupt signal (INTIT00). The clock signal is output through the EXSDO0 and EXSDO1 pins.
- Latches the levels of input signals to the EXSDI0 and EXSDI1 pins on falling edges of the generated sampling clock and shapes the input waveforms. It also detects the phases of the shaped input waveforms.
- Generates the interrupt signal (INTEXSD) on the falling and rising edges of the shaped waveforms derived from the input signal to the EXSDI1 pin.

13.2 Configuration of External Signal Sampler

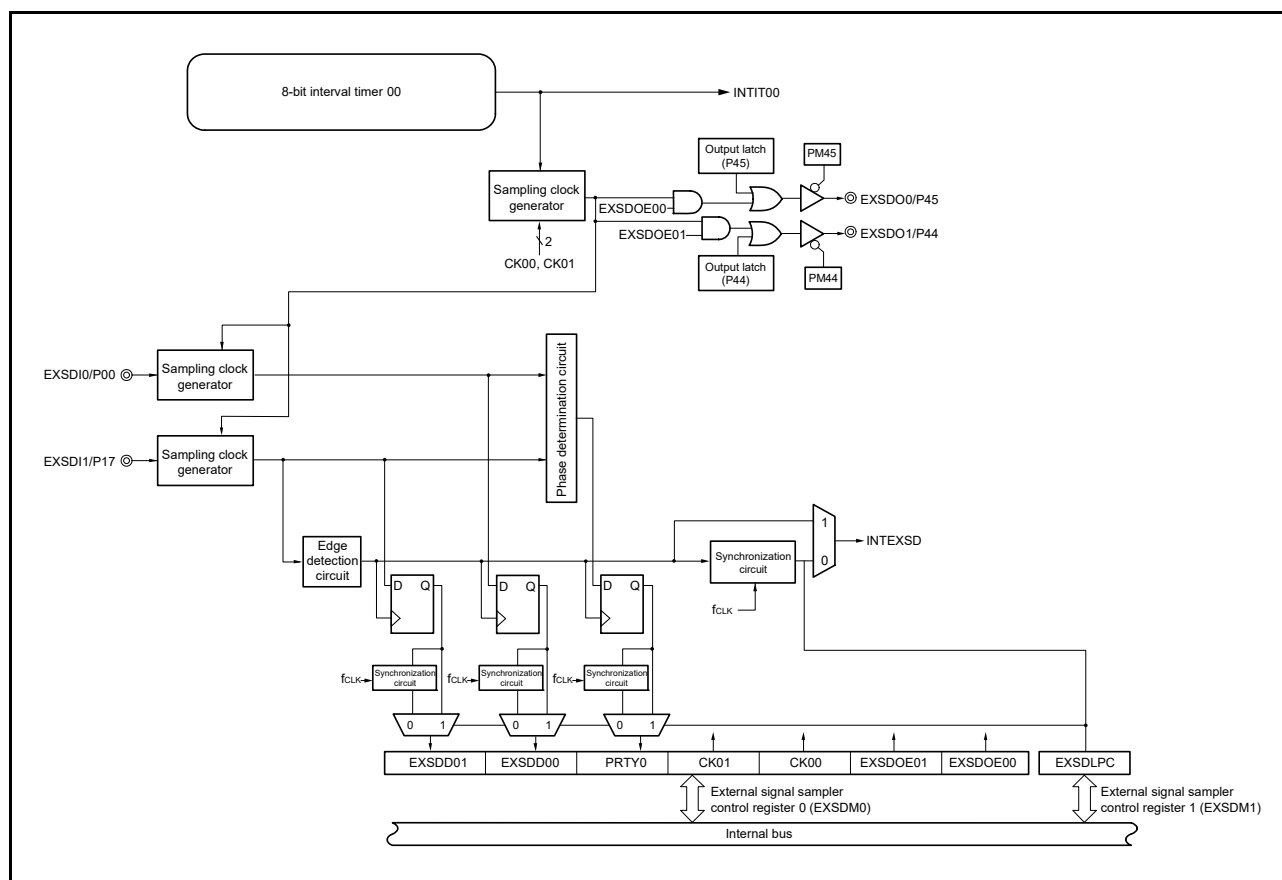
The external signal sampler includes the following hardware.

Table 13 - 1 Configuration of the External Signal Sampler

Item	Configuration
Control registers	Peripheral enable register 2 (PER2) Peripheral reset control register 2 (PRR2) External signal sampler control register 0 (EXSDM0) External signal sampler control register 1 (EXSDM1) Port mode registers (PMxx) Port registers (Pxx) Port output mode registers (POMxx) Port mode control A registers (PMCAxx) Port mode control T registers (PMCTxx) Port mode control E registers (PMCExx) LCD port function registers (PFSEGx)

Figure 13 - 1 shows a block diagram of the external signal sampler.

Figure 13 - 1 Block Diagram of the External Signal Sampler



Remark 1. fCLK: CPU/peripheral hardware clock

Remark 2. The pins of the external signal sampler in the above diagram are those when $PIORx = 0$ in a product with 48 to 100 pins.

13.3 Registers for Controlling the External Signal Sampler

The following registers are used to control the external signal sampler.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- External signal sampler control register 0 (EXSDM0)
- External signal sampler control register 1 (EXSDM1)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

13.3.1 Peripheral enable register 2 (PER2)

The PER2 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 the external signal sampler, be sure to set bit 7 (EXSDEN) to 1. The PER2 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 13 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN	UTA2EN	UTA1EN	TRJ1EN	TRJ0EN
	EXSDEN	Control of supply of an input clock to the external signal sampler						
	0	Stops supply of an input clock. - The SFRs used by the external signal sampler cannot be written. - When an SFR used by the external signal sampler is read, the value returned is 00H.						
	1	Enables supply of an input clock. The SFRs used by the external signal sampler can be read and written.						

Caution Be sure to set the EXSDEN bit to 1 before making the external signal sampler settings. If EXSDEN = 0, writing to the registers for controlling the external signal sampler is ignored, and the value read is 00H. Note, however, that writing to the following registers is valid.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

13.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 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 7 (EXSDRES) to 1 to reset the external signal sampler. The PRR2 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 13 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
PRR2	EXSDRES	OSDCRES	TKBRES	0	0	0	TRJ1RES	TRJ0RES
	EXSDRES	Control resetting of the external signal sampler						
	0	The external signal sampler is released from the reset state.						
	1	The external signal sampler is in the reset state. • The SFRs for use with the external signal sampler are initialized.						

Caution 1. Be sure to clear bits 4 to 2 to 0.

Caution 2. The functions mounted depend on the product. For details, see Section 28 Reset Function.

13.3.3 External signal sampler control register 0 (EXSDM0)

The EXSDM0 register is used to indicate the sampling state and control the operation of the EXSDO0, EXSDO1, EXSDI0, and EXSDI1 pins. The EXSDM0 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 13 - 4 Format of External Signal Sampler Control Register 0 (EXSDM0)

Address: F025BH

After reset: 00H

R/W: R/W^{Note}

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
EXSDM0	EXSDD01	EXSDD00	PRTY0	0	CK01	CK00	EXSDOE01	EXSDOE00
	EXSDD01	Status flag of the internally shaped waveform level derived from the input signal to EXSDI1						
	0	Low level						
	1	High level						
<R>	EXSDD00	Status flag of the internally shaped waveform level derived from the input signal to EXSDI0 on the falling or rising edge of the internally shaped waveform derived from the input signal to EXSDI1						
	0	Internally shaped waveform derived from the input signal to the EXSDI0 pin is low.						
	1	Internally shaped waveform derived from the input signal to the EXSDI0 pin is high.						
<R>	PRTY0	Parity flag for the EXSDD01 and EXSDD00 flags						
	0	Internally shaped waveform derived from the input signal to EXSDI0 is high on the rising edge of internally shaped waveform derived from the input signal to EXSDI1, or low on the falling edge of internally shaped waveform derived from input signal to EXSDI1.						
	1	Internally shaped waveform derived from the input signal to EXSDI0 is low on the rising edge of the internally shaped waveform derived from the input signal to EXSDI1, or high on the falling edge of the internally shaped waveform derived from the input signal to EXSDI1.						
	CK01	CK00	Setting of pulse width of clock signal output from the EXSDO1 and EXSDO0 pins					
	0	0	1/(2 × fsXP) (15 μs)					
	0	1	1/fsXP (30.5 μs)					
	1	0	2/fsXP (61 μs)					
	1	1	2 ⁵ /fsXP (977 μs)					
	EXSDOE01	EXSDO1 pin output control flag						
	0	Disables output.						
	1	Enables output.						
	EXSDOE00	EXSDO0 pin output control flag						
	0	Disables output.						
	1	Enables output.						

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

Note Bits 7 to 5 are read-only bits.

Remark The values in parentheses indicate the settings for $fs_{XP} = 32.768 \text{ kHz}$.

13.3.4 External signal sampler control register 1 (EXSDM1)

The EXSDM1 register is used to switch the operating mode of the external signal sampler. Setting the EXSDLPC bit to 1 enables updating the EXSD00, EXSD01, and PRTY0 bits of the external signal sampler control register 0 (EXSDM0) and generation of the external signal sampler edge detection interrupt signal (INTEXSD) in STOP mode and SNOOZE mode. The EXSDM1 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 13 - 5 Format of External Signal Sampler Control Register 1 (EXSDM1)

Address: F025AH
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
EXSDM1	0	0	0	0	0	0	0	EXSDLPC

EXSDLPC	Switching of the operating mode of the external signal sampler
0	When the external signal sampler is not to be used in STOP mode and SNOOZE mode. Note 1
1	When the external signal sampler is to be used in STOP mode and SNOOZE mode. Notes 2, 3

Note 1. The following functions cannot be used in STOP mode and SNOOZE mode because operation is in synchronization with the CPU/peripheral hardware clock (fCLK), which is stopped.

- Generation of the external signal sampler edge detection interrupt signal (INTEXSD)
- Updating of the EXSD01, EXSD00, and PRTY0 bits in external signal sampler control register 0 (EXSDM0)

Note 2. When the EXSDLPC bit is set to 1, the EXSD00, EXSD01, and PRTY0 bits are updated asynchronously. Read the EXSD00, EXSD01, and PRTY0 bits immediately after generation of the external signal sampler edge detection interrupt signal (INTEXSD).

Note 3. Setting the external signal sampler edge detection as the DTC activation source is prohibited when the setting of the EXSDLPC bit is 1.

13.3.5 Registers for controlling the port functions multiplexed with the inputs and outputs of the external signal sampler

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the external signal sampler.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

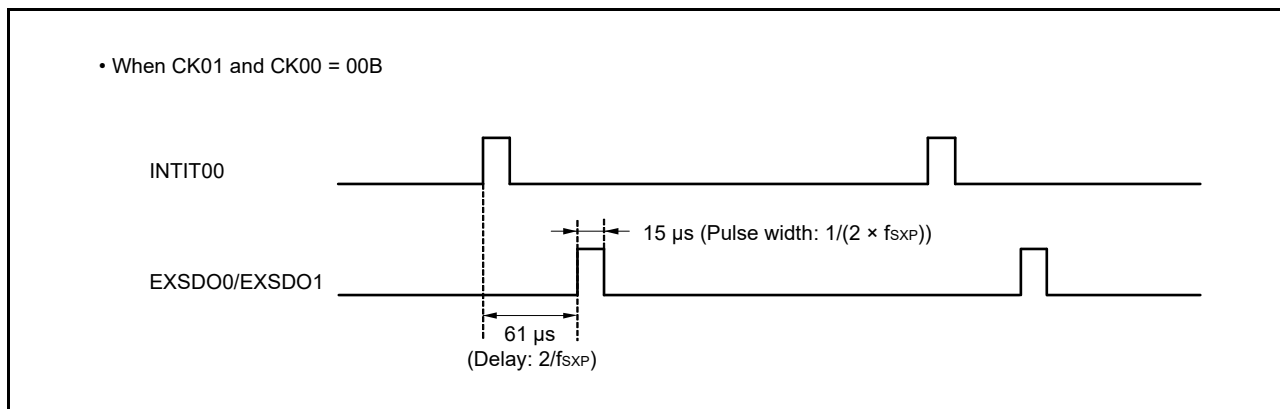
For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

13.4 Sampling Clock Output Operation

The output period of the output signal from the EXSDO0 and EXSDO1 pins is set by the 8-bit interval timer 00 interrupt signal (INTIT00) generation period. The pulse width is set by bits 3 and 2 (CK01 and CK00) of external signal sampler control register 0 (EXSDM0).

Figure 13 - 6 shows an example of the waveform of the output signal from EXSDO0 and EXSDO1.

Figure 13 - 6 Example of Waveforms of Output Signal from the EXSDO0 and EXSDO1 Pins



Caution 1. Set the TRTCMP00 register so that the INTIT00 signal generation interval is at least $4/f_{sXP}$. If f_{sXP} is selected for 8-bit interval timer 00, set $TRTCMP00 \geq 0003H$. In the other cases, set $TRTCMP00 \geq 0001H$.

Caution 2. Set the pulse width of the output signal from the EXSDO0 and EXSDO1 pins so that the duty cycle is no greater than 1/2.

Caution 3. When using the external signal sampler, set the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) to 0 to select the subsystem clock as the operating clock.

Remark $f_{sXP} = 32.768 \text{ kHz}$

13.5 Operation of External Signal Sampler

1. Initial setting

The initial setting of the external signal sampler is made in the following procedure.

- Set bits 4 and 5 (PM44 and PM45) of port mode register 4 (PM4) to output mode and set the output latch bits for the P44 and P45 pins to 0. Set bit 0 (PM00) of port mode register 0 (PM0) and bit 7 (PM17) of port mode register 1 (PM1) to input mode. Set bits 3 and 2 (PFSEG43 and PFSEG42) of LCD port function register 6 (PFSEG6) to 0.
- Set the interrupt mask flag (EXSDMK) for the external signal sampler edge detection interrupt (INTEXSD) to 0 (to enable interrupt processing).
- Stop counting by 8-bit interval timer 00.
- Set a desired counter value in 8-bit interval timer 00.
- Set bit 7 (EXSDEN) of peripheral enable register 2 (PER2) to 1.
- Set the pulse width of the sampling clock output from the EXSDO0 and EXSDO1 pins by bits 3 and 2 (CK01 and CK00) of external signal sampler control register 0 (EXSDM0).
- Set bit 0 (EXSDLPC) of external signal sampler control register 1 (EXSDM1) to 0.

2. Starting the external signal sampler

The external signal sampler is started by the following procedure.

- Enable output from the EXSDO0 and EXSDO1 pins by bits 0 and 1 (EXSDOE00 and EXSDOE01) of the EXSDM0 register.
- Start counting by channel 0 of 8-bit interval time 00.
- The levels of input signals to the EXSDI0 and EXSDI1 pins are latched on falling edges of the sampling clock signals from the EXSDO0 and EXSDO1 pins and the waveforms of the input signal are shaped in response.
- The interrupt signal (INTEXSD) is generated on the rising and falling edges of the internally shaped waveform derived from the input signal to the EXSDI1 pin. The generation of INTEXSD is delayed by $1/(2 \times f_{SXP}) + 1/f_{CLK}$ or $1/(2 \times f_{SXP}) + 2/f_{CLK}$ relative to the shaped waveform derived from the input signal to the EXSDI1 pin.
- The level of the internally shaped waveform derived from the input signal to the EXSDI1 pin is latched at the time of generation of INTEXSD and the level is sent to bit 7 (EXSDD01) of the EXSDM0 register. When the level being input to the EXSDI1 pin is high, the bit is set to 1. When the level is low, the bit is cleared to 0.
- The level of the internally shaped waveform derived from the input signal to the EXSDI0 pin is latched on the falling and rising edges of the internally shaped waveform derived from the input signal to the EXSDI1 pin and the level is sent to bit 6 (EXSDD00) of the EXSDM0 register at the time of generation of INTEXSD. When the level being input to the EXSDI0 pin is high, the bit is set to 1. When the level is low, the bit is cleared to 0.
- Bit 5 (PRTY0) of the EXSDM0 register is cleared to 0 when the level of the shaped waveform derived from the input signal to the EXSDI0 pin is high on the rising edge of the shaped waveform derived from the input signal to the EXSDI1 pin or when the level being input to the EXSDI0 pin is low on the falling edge of the input signal to the EXSDI1 pin. The PRTY0 flag is set to 1 when the level of the shaped waveform derived from the input signal to the EXSDI0 pin is low on the rising edge of the shaped waveform derived from the input signal to the EXSDI1 pin or when the level being input to the EXSDI0 pin is high on the falling edge of the input signal to the EXSDI1 pin.
- When the EXSDLPC bit of external signal sampler control register 1 (EXSDM1) is 0, fCLK is delayed. When the EXSDLPC bit is 1, fCLK is not delayed.

3. Stopping the external signal sampler

The external signal sampler is stopped by the following procedure.

- Set the interrupt mask flag (EXSDMK) for the external signal sampler edge detection interrupt (INTEXSD) to 1 (to disable interrupt processing).
- Stop counting by 8-bit interval time 00.
- Disable output from the EXSDO0 and EXSDO1 pins by bits 0 and 1 (EXSDOE00 and EXSDOE01) of the EXSDM0 register.

Caution 1. The generation of INTEXSD is always delayed by $1/(2 \times f_{SXP}) + 1/f_{CLK}$ or $1/(2 \times f_{SXP}) + 2/f_{CLK}$ relative to the internally shaped waveform derived from the input signal to the EXSDI1 pin.

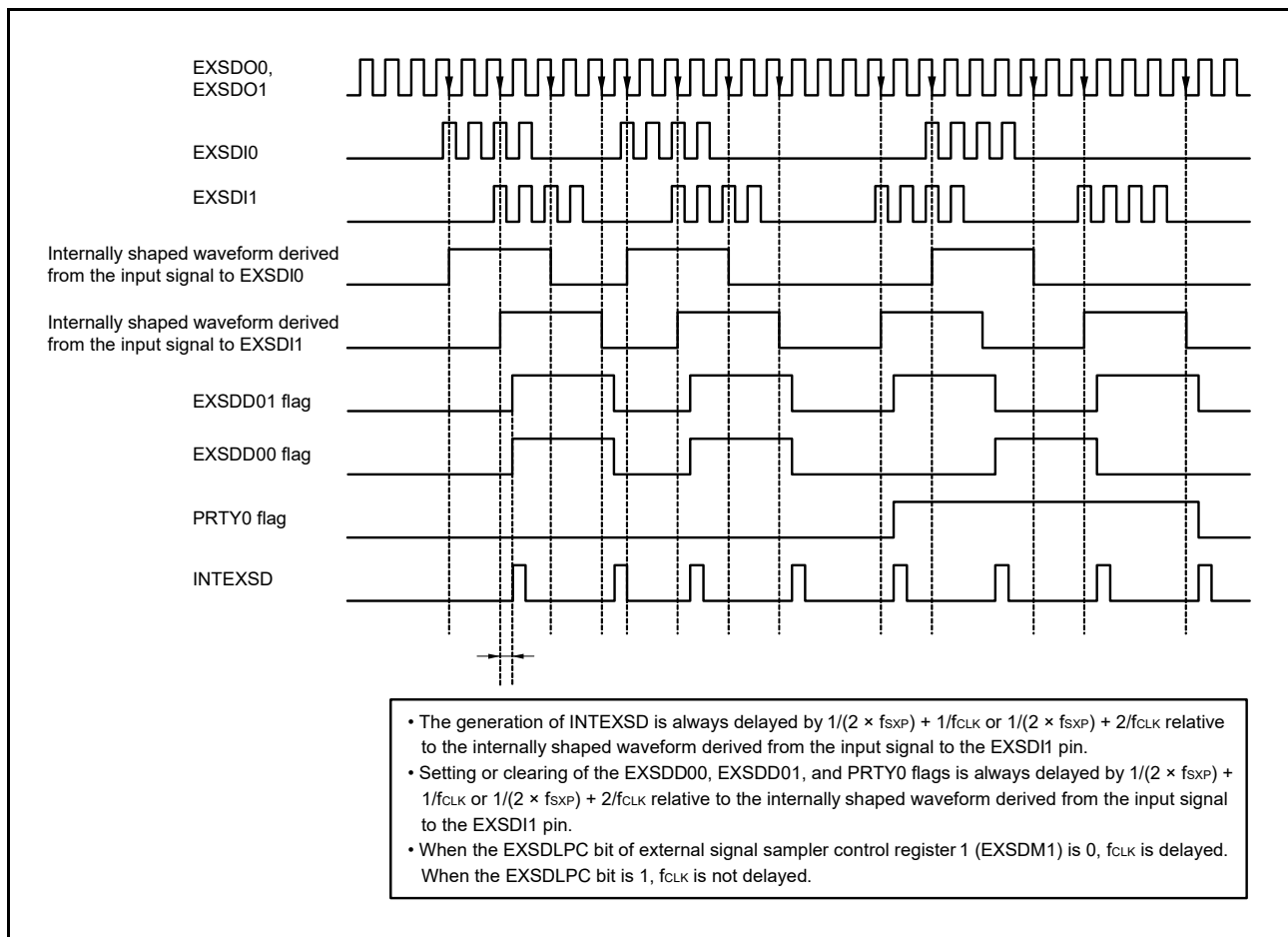
Caution 2. Setting or clearing of the EXSDD00, EXSDD01, and PRTY0 flags is always delayed by $1/(2 \times f_{SXP}) + 1/f_{CLK}$ or $1/(2 \times f_{SXP}) + 2/f_{CLK}$ relative to the internally shaped waveform derived from the input signal to the EXSDI1 pin.

Caution 3. When the EXSDLPC bit of external signal sampler control register 1 (EXSDM1) is 0, fCLK is delayed. When the EXSDLPC bit is 1, fCLK is not delayed.

Remark The procedure is for setting the port functions of the external signal sampler pins when PIORx = 0 in a product with 48 to 100 pins.

Figure 13 - 7 shows a timing of the operation of the external signal sampler.

Figure 13 - 7 Example of Operation of the External Signal Sampler



13.6 Cautions

1. If edges of input signals to the EXSDI0 and EXSDI1 pins occur simultaneously, the value of the level status flag for the internally shaped waveform derived from the input signal to the EXSDI0 pin is undefined.
2. The interrupt signal (INTEXSD) and status flags (EXSDD00, EXSDD01, and PRTY0) change simultaneously.
3. Set the EXSDM0 register while 8-bit interval timer 00 is stopped.
4. Set the TRTCMP00 register so that the interval for generation of the 8-bit interval timer 00 interrupt signal (INTIT00) is at least $4/f_{SXP}$. If f_{SXP} is selected for 8-bit interval timer 00, set $TRTCMP00 \geq 0003H$. In the other cases, set $TRTCMP00 \geq 0001H$.
5. Set the pulse width of the EXSDO0 and EXSDO1 pins so that the duty cycle is no greater than 1/2.
6. When the external signal sampler is to be used in STOP mode or SNOOZE mode, set a value greater than or equal to the number of cycles from interrupt request to normal operation in 8-bit interval timer counter register 00.
If a value less than the number of cycles from interrupt request to normal operation is set, the EXSDD01, EXSDD00, and PRTY0 bits of external signal sampler control register 0 (EXSDM0) may not be read correctly.
For the number of cycles, see **27.3 Standby Function Operation**.

Section 14 Clock Output/Buzzer Output Controller

The number of output pins for the clock output/buzzer output controller depends on the product.

Output Pins	44- to 52-pin	64- to 100-pin
PCLBUZ0	(✓)	✓
PCLBUZ1	(✓)	✓

Caution Most of the following descriptions in this section use the 100-pin products as an example.

Remark Parentheses indicate that the pin can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

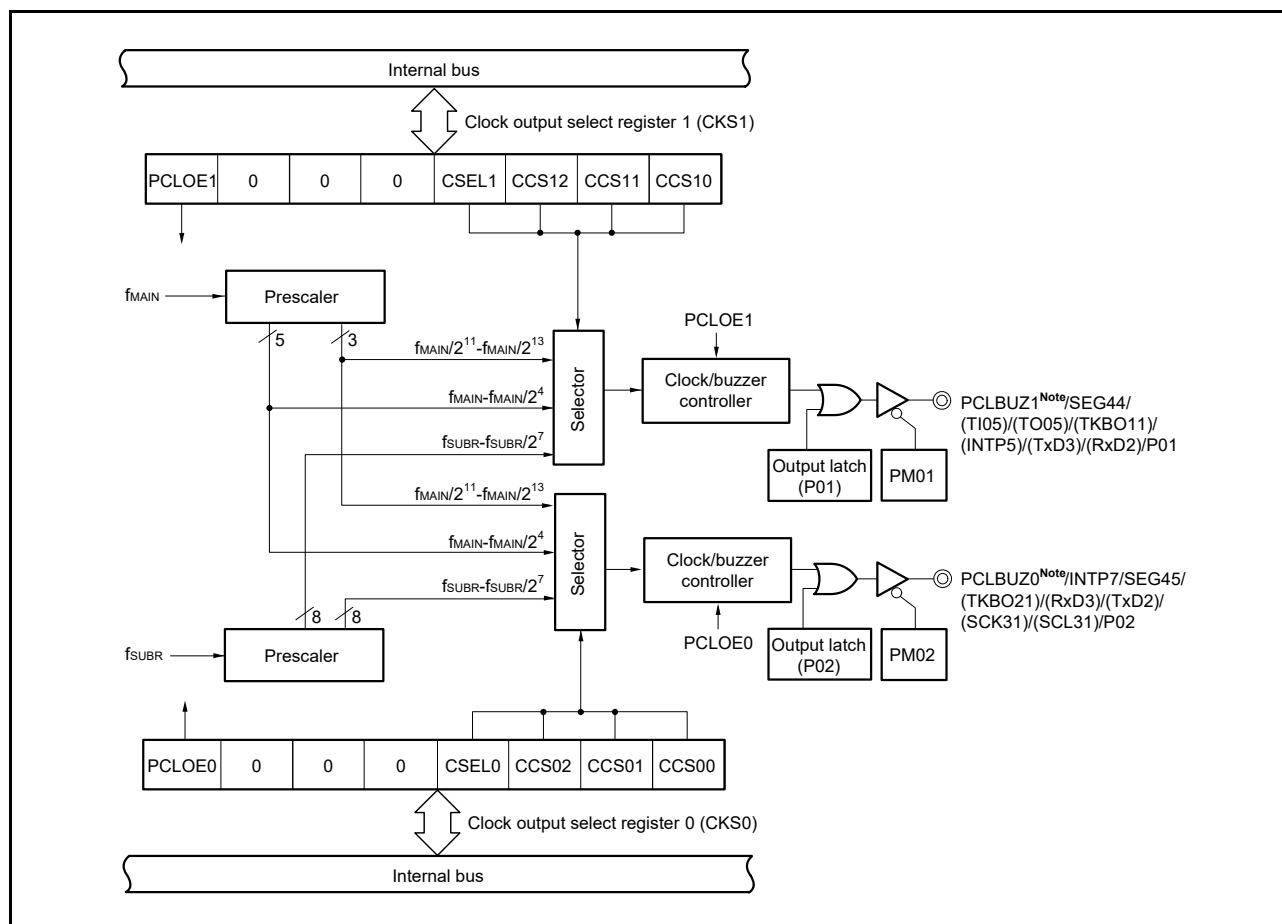
14.1 Functions of Clock Output/Buzzer Output Controller

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 14 - 1 shows the block diagram of clock output/buzzer output controller.

Remark n = 0, 1

Figure 14 - 1 Block Diagram of the Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, see **42.4 AC Characteristics**.

Remark The clock output/buzzer output pins in the above diagram are those when the value of PIOR50 to PIOR52 is 0H. If this is not the case, the names of pins, output latches (Pxx), and PMxx should be read differently in accord with the setting.

14.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 14 - 1 Configuration of the Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode registers (PMxx) Port registers (Pxx) Port output mode registers (POMxx) Port mode control A registers (PMCAxx) Port mode control T registers (PMCTxx) LCD port function registers (PFSEGx)

14.3 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 output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- LCD port function registers (PFSEGx)

14.3.1 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 14 - 2 Format of Clock Output Select Registers 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
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 _{SUBR}	32.768 kHz			
1	0	0	1	f _{SUBR} /2	16.384 kHz			
1	0	1	0	f _{SUBR} /2 ²	8.192 kHz			
1	0	1	1	f _{SUBR} /2 ³	4.096 kHz			
1	1	0	0	f _{SUBR} /2 ⁴	2.048 kHz			
1	1	0	1	f _{SUBR} /2 ⁵	1.024 kHz			
1	1	1	0	f _{SUBR} /2 ⁶	512 Hz			
1	1	1	1	f _{SUBR} /2 ⁷	256 Hz			

Note The selectable output clock frequency depends on the power supply voltage (V_{DD}). The output clock frequency must be no greater than 16 MHz. See **42.4 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 to 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn can be set to 1 because the clock can be output in STOP mode.

(Remarks are listed on the next page.)

Remark 1. $n = 0, 1$

Remark 2. f_{MAIN} : Main system clock frequency
 f_{SUBR} : Subsystem clock R frequency

14.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs

Set the following registers for controlling the port functions multiplexed with the clock or buzzer outputs.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

14.4 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).

14.4.1 Operation of output pins

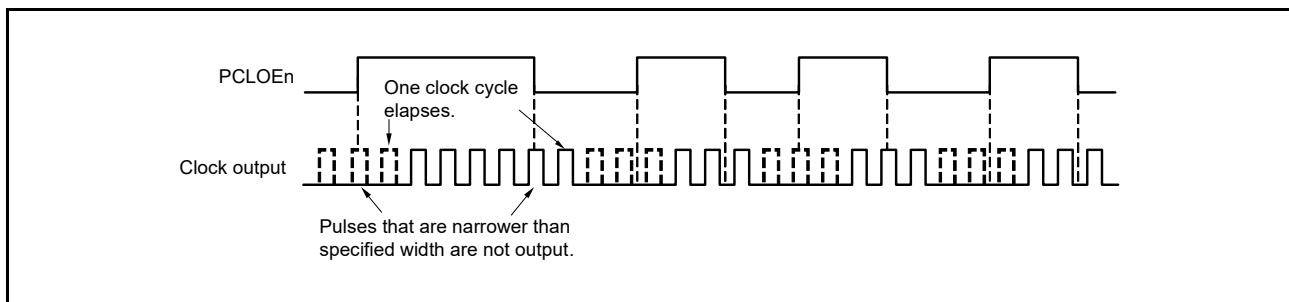
Follow the steps below to enable output from a PCLBUZn pin.

- <1> Make the port settings for the pin to be used as the PCLBUZn pin. See **14.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs**.
- <2> Select the output frequency with bits 3 to 0 (CSELn, CCSn2 to CCSn0) of clock output select register n (CKSn) for the PCLBUZn pin (output is 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 14 - 3** shows the relationship between the setting of the PCLOEn bit and the timing of clock output.

Remark 2. n = 0, 1

Figure 14 - 3 Timing of the Clock Output from a PCLBUZn Pin



14.5 Point for Caution When the Clock Output/Buzzer Output Controller Is to be Used

When the main system clock is selected for output from a PCLBUZn pin (CKSn.CSELn = 0), if the setting to stop output from the PCLBUZn pin is made (CKSn.PCLOEn = 0) and then STOP mode is entered within 1.5 clock cycles of the main system clock, the width of output from the PCLBUZn pin will be shortened.

Section 15 Watchdog Timer (WDT)

15.1 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 28 Reset Function**. When 75% of the overflow time + 1/4 f_{IL} is reached, an interval interrupt can be generated.

15.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 15 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

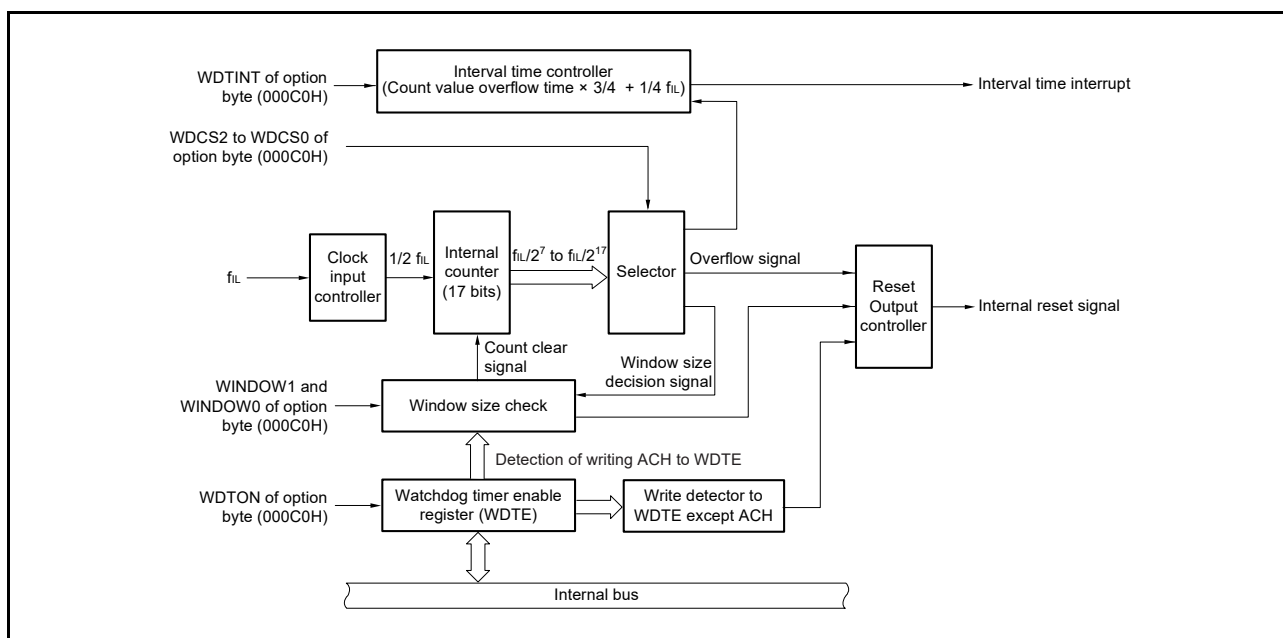
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 15 - 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 37 Option Bytes**.

Figure 15 - 1 Block Diagram of Watchdog Timer



Remark f_{IL} : Low-speed on-chip oscillator clock

15.3 Register for Controlling the Watchdog Timer

The following register is used to control the watchdog timer.

- Watchdog timer enable register (WDTE)

15.3.1 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}.

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.

Figure 15 - 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								

WDTON Bit Setting Value	Value of the WDTE Register following a Reset
0 (Watchdog timer counting disabled)	1AH
1 (Watchdog timer counting enabled)	9AH

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.

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.

15.4 Operation of Watchdog Timer

15.4.1 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 release from the reset state). For details, see **Section 37 Option Bytes**.

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after release from the reset state)
1	Counter operation enabled (counting started after release from the reset state)

- 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 **15.4.2** and **Section 37 Option Bytes**.
 - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H). For details, see **15.4.3** and **Section 37 Option Bytes**.
- After release from the reset state, 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 release from the reset state, 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.

15.4.2 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 15 - 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 (FFFA BH) 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.

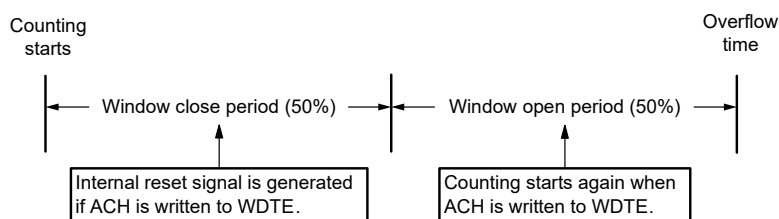
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

15.4.3 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 release from the reset state, 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 15 - 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:
 $2^{10}/f_{IL} \text{ (max.)} = 2^{10}/37.683 \text{ kHz} = 27.17 \text{ ms}$
- Window close time:
 $0 \text{ to } 2^{10}/f_{IL} \text{ (min.)} \times (1 - 0.5) = 0 \text{ to } 2^{10}/27.852 \text{ kHz} \times 0.5 = 0 \text{ to } 18.38 \text{ ms}$
- Window open time:
 $2^{10}/f_{IL} \text{ (min.)} \times (1 - 0.5) \text{ to } 2^{10}/f_{IL} \text{ (max.)} = 2^{10}/27.853 \text{ kHz} \times 0.5 \text{ to } 2^{10}/37.683 \text{ kHz} = 18.38 \text{ to } 27.17 \text{ ms}$

15.4.4 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 15 - 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 16 A/D Converter (ADC)

The number of analog input channels of the A/D converter differs, depending on the product.

	44-pin	48-pin	52- and 64-pin	80- and 100-pin
Number of the analog input channels	8 (ANI0, ANI1, ANI16, ANI20 to ANI24)	9 (ANI0, ANI1, ANI16, ANI20 to ANI25)	10 (ANI0, ANI1, ANI16, ANI20 to ANI26)	13 (ANI0, ANI1, ANI16 to ANI26)

16.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control up to 13 channels of A/D converter analog inputs (ANI0, ANI1, and ANI16 to ANI26). 12-bit, 10-bit, or 8-bit resolution can be selected by the ADTYP1 and ADTYP0 bits of A/D converter mode register 2 (ADM2). The A/D converter has the following function.

- 12-bit/10-bit/8-bit resolution A/D conversion
12-bit, 10-bit, or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0, ANI1, and ANI16 to ANI26. Each time an A/D conversion operation ends, an interrupt request signal (INTAD) is generated.

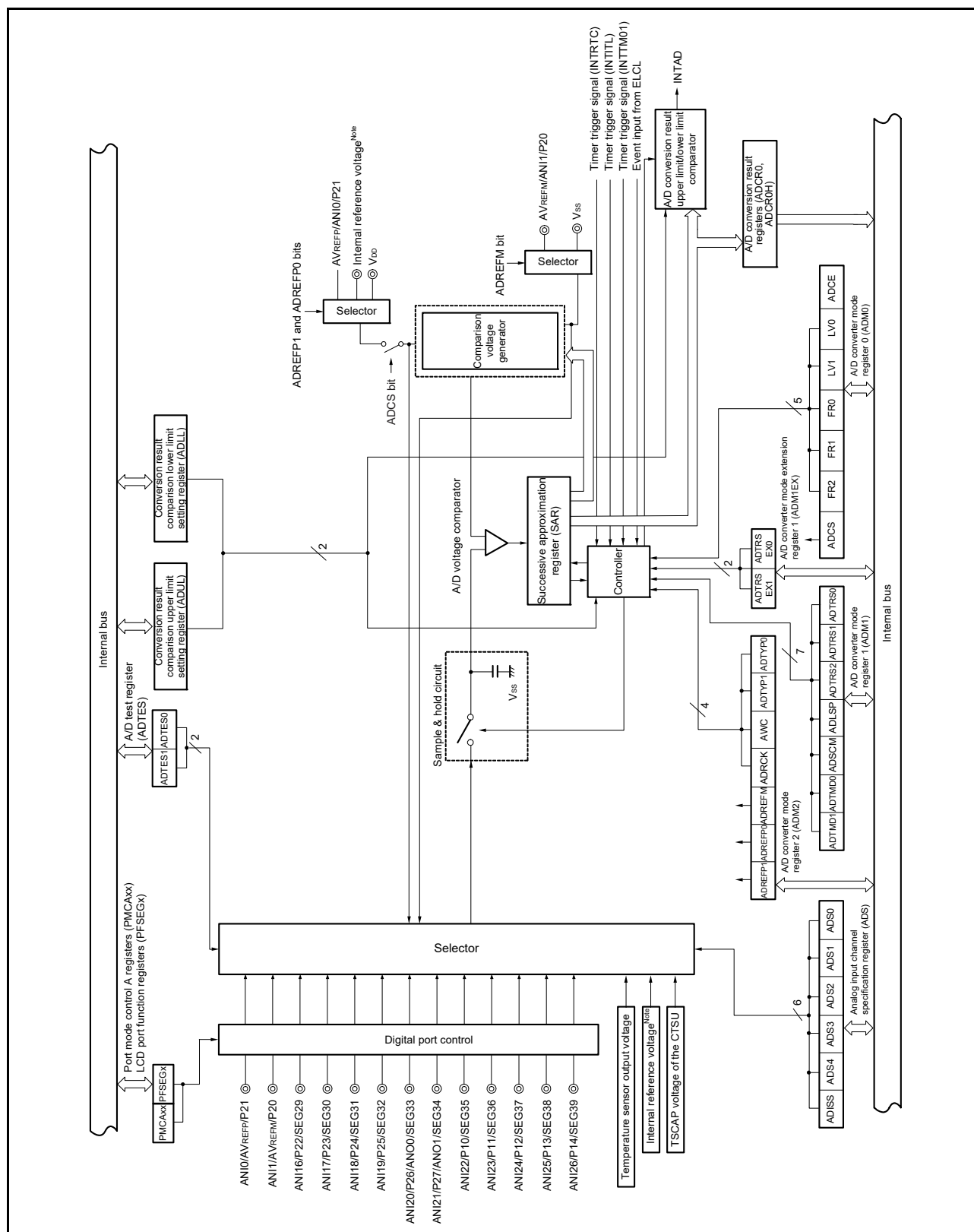
Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger no-wait mode	Conversion is started by setting the ADCE bit to 1 by software, and then setting ADCS to 1 after the A/D power supply stabilization wait time has passed.
	Software trigger wait mode	The power is turned on by setting the ADCS bit to 1 by software while A/D conversion is stopped and conversion is then started automatically after the A/D power supply stabilization wait time has passed.
	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 A/D converter is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode, specify the hardware trigger wait mode.
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 ^{Note}	Sampling clock cycles	
Normal 1	43 fAD	Set the number of sampling clock cycles so that the sampling capacitor is sufficiently charged according to the output impedance of the analog input source.
Normal 2	160 fAD	
Low voltage 1	53 fAD	
Low voltage 2	80 fAD	

Note The operation mode that can be selected differs depending on the analog input channel, VDD voltage, AVREFP voltage, trigger mode, and fCLK. See **Table 16 - 3 A/D Conversion Time Selection (1/4)** for details.

Figure 16 - 1 Block Diagram of A/D Converter



Note For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Remark Analog input pins in this figure are for a 100-pin product.

16.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

1. ANI0, ANI1, and ANI16 to ANI26 pins

These are the analog input pins of the 13 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 ($1/2 AV_{REF}$) 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 ($1/2 AV_{REF}$), the MSB of the SAR is reset. After that, bit 10 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 11, to which the result has already been set.

- Bit 11 = 0: ($1/4 AV_{REF}$)
- Bit 11 = 1: ($3/4 AV_{REF}$)

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

- Analog input voltage $\geq$ Voltage tap of comparison voltage generator: Bit 10 = 1
- Analog input voltage $\leq$ Voltage tap of comparison voltage generator: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR register.

4. Comparison voltage generator

The comparison voltage generator generates the voltage to be compared with the input from an analog input pin.

5. Successive approximation register (SAR)

The SAR is used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, one 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 (ADCR0). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

6. 12-bit/10-bit A/D conversion result register (ADCR0)

Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and then operation is performed as follows:

When this register is used to specify 12-bit resolution, it holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).

When this register is used to specify 10-bit resolution, it 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 (ADCR0H)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR0H register holds the higher 8 bits of the A/D conversion result.

8. Controller

This circuit controls the conversion time of an analog input 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 + side 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 ANI16 to ANI26 are converted to digital signals based on the voltage applied between AVREFP and the – side reference voltage (AVREFM/VSS).

10. AVREFM pin

This pin inputs an external reference voltage (AVREFM). To use AVREFM as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

Remark AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage, and VDD. For the internal reference voltage, see **Section 42 Electrical Characteristics**.

16.3 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 extension register 1 (ADM1EX)
- A/D converter mode register 2 (ADM2)
- 12-bit/10-bit A/D conversion result register (ADCR0)
- 8-bit A/D conversion result register (ADCR0H)
- 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 registers (PMxx)
- Port mode control A registers (PMCAxx)
- LCD port function registers (PFSEGx)

16.3.1 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 16 - 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	IICA1EN	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 value of each register which controls the A/D converter is 00H and writing to any of those registers is ignored.

- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode extension register 1 (ADM1EX)
- A/D converter mode register 2 (ADM2)
- 12-bit/10-bit A/D conversion result register (ADCR0)
- 8-bit A/D conversion result register (ADCR0H)
- 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).

Note, however, that writing to the following registers is valid.

- Port mode registers 1 and 2 (PM1 and PM2)
- Port mode control A registers 1 and 2 (PMCA1 and PMCA2)
- LCD port function registers 4 and 5 (PFSEG4 and PFSEG5)

Caution 2. Be sure to clear bit 1 to 0.

16.3.2 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 5 (ADCRES) to 1 to reset the A/D converter. 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 16 - 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	IICA1RES	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 Be sure to clear bits 7 and 1 to 0.

16.3.3 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 16 - 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	0	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 no wait mode: Conversion is enabled. While in the software trigger wait mode: A/D power supply stabilization wait time + conversion While in the hardware trigger no wait mode: Conversion is enabled. While in the hardware trigger wait mode: A/D power supply stabilization wait time + conversion

ADCE	A/D voltage comparator operation control
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 16 - 3 A/D Conversion Time Selection (1/4)**.

Note 2. While in the software trigger no-wait 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 $1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD}) from the start of operation for the operation to stabilize. Therefore, immediately after the ADCS bit is set to 1 after at least $1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD}) have elapsed from the time ADCE bit is set to 1, the conversion result becomes valid. When ADCS is set to 1 while ADCE = 0, A/D conversion starts after the stabilization wait time has passed. If ADCS is set before at least $1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD}) have elapsed, ignore data of the first conversion.

Caution 1. Change the FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Setting change from ADCS = 1 and ADCE = 1 to ADCS = 1 and ADCE = 0 is prohibited.

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 follow the procedure described in 16.7 A/D Converter Setup Flowchart.

Caution 4. Following stoppage of conversion by setting the ADCS and ADCE bits to 0 from the conversion standby or conversion state, wait for at least $5\ \mu\text{s}$ before restoring the values of the bits to 1. Note that, when changing the settings of bits FR2 to FR0, LV1, and LV0, start by setting the ADCS and ADCE bits to 0, then wait for at least $0.2\ \mu\text{s}$ before changing the rest of the bits.

Caution 5. Be sure to clear bit 6 to 0.

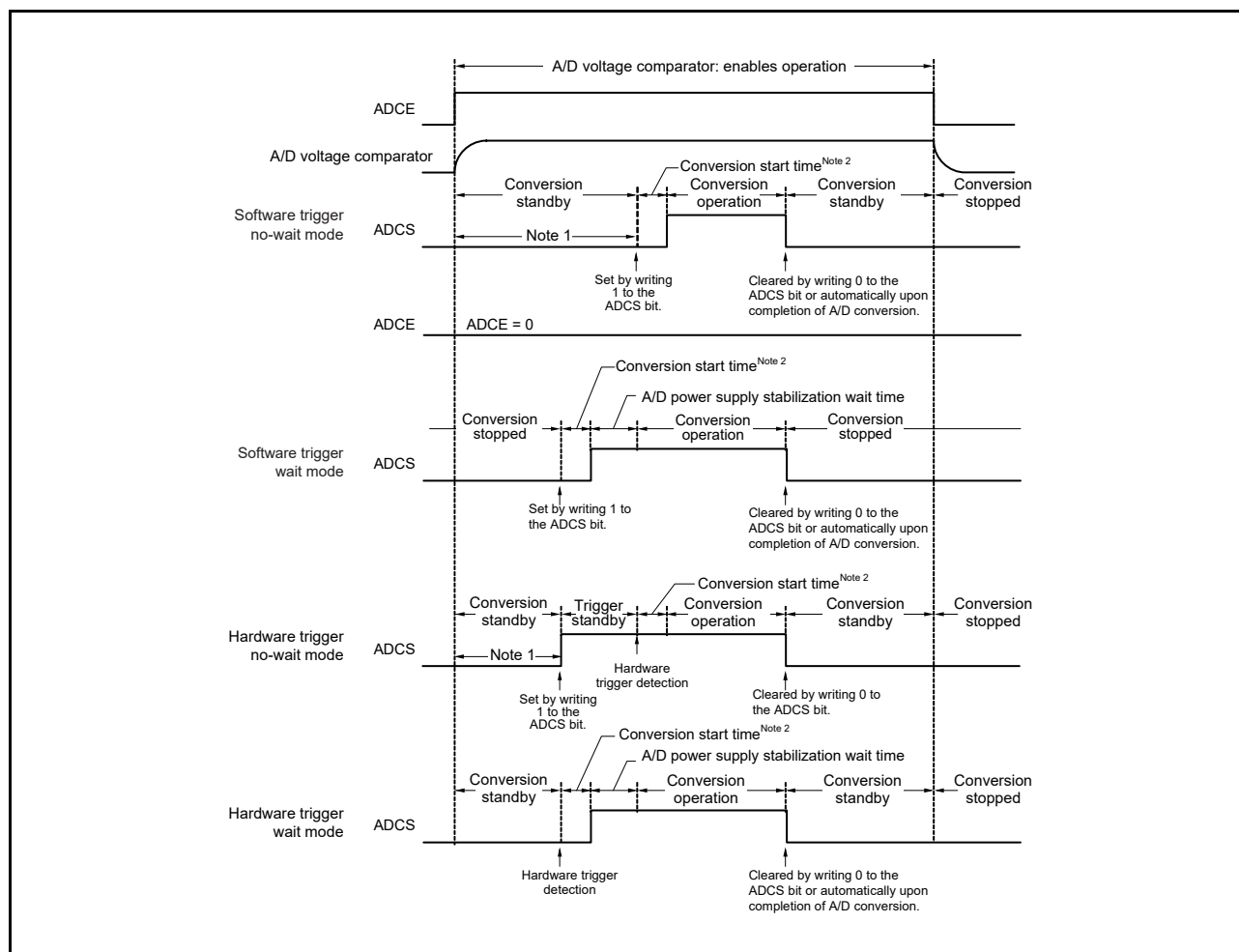
Table 16 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Conversion operation state (software trigger wait mode) or conversion stopped state
1	1	Conversion operation state (other than software trigger wait mode)

Table 16 - 2 Conditions for Setting and Clearing the ADCS Bit

A/D Conversion Mode		Set Conditions	Clear Conditions
Software trigger no-wait 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.
Software trigger wait 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 A/D conversion ends.
Hardware trigger no-wait 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
Hardware trigger wait 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 A/D conversion ends.

Figure 16 - 5 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger no-wait 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 $1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD}) or longer to stabilize the internal circuit.

Note 2. Starting the conversion takes up to the following time. However, in the second and subsequent conversions in sequential conversion mode, the conversion start time and A/D power supply stabilization wait time do not occur after detection of a hardware trigger.

ADM1	ADM0			Conversion Clock (fAD)	Conversion Start Time (Number of fCLK Clock)	
ADLSP	FR2	FR1	FR0		Software Trigger No-wait Mode/ Hardware Trigger No-wait Mode	Software Trigger Wait Mode/ Hardware Trigger Wait Mode
0	0	0	0	fCLK/32	31	1
0	0	0	1	fCLK/16	15	1
0	0	1	0	fCLK/8	7	1
0	0	1	1	fCLK/4	3	1
0	1	0	0	fCLK/2	1	1
0	1	0	1	fCLK	1	1
1	0	1	1	fCLK/4	3	1
1	1	0	0	fCLK/2	1	1
1	1	0	1	fCLK	1	1

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 fCLK clock cycles + conversion start time + A/D conversion time
 Hardware trigger wait mode: 2 fCLK clock cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 5 μ s

Remark fCLK: CPU/peripheral hardware clock frequency

Table 16 - 3 A/D Conversion Time Selection (1/4)

1. When there is no A/D power supply stabilization wait time

Normal mode 1 and 2 (for software trigger no-wait mode and hardware trigger no-wait mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (fAD)	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)					
(AD M1)	(ADM0)										2.4 V ≤ AVREFP ≤ VDD ≤ 5.5 V					
ADL SP	FR2	FR1	FR0	LV1	LV0						fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz	
0	0	0	0	0	0	Normal 1	fCLK/32	1 fAD	64 fAD	1 fAD	2112/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	66 μs
0	0	0	1				fCLK/16	1 fAD	64 fAD	1 fAD	1056/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	66 μs	33 μs
0	0	1	0				fCLK/8	1 fAD	64 fAD	1 fAD	528/fCLK	Setting prohibited	Setting prohibited	66 μs	33 μs	16.5 μs
0	0	1	1				fCLK/4	1 fAD	64 fAD	1 fAD	264/fCLK	Setting prohibited	Setting prohibited	33 μs	16.5 μs	8.25 μs
0	1	0	0				fCLK/2	1 fAD	64 fAD	1 fAD	132/fCLK	Setting prohibited	Setting prohibited	16.5 μs	8.25 μs	4.125 μs
0	1	0	1				fCLK	1 fAD	64 fAD	1 fAD	66/fCLK	Setting prohibited	Setting prohibited	8.25 μs	4.125 μs	2.0625 μs
1	0	1	1				fCLK/4	1 fAD	64 fAD	1 fAD	264/fCLK	Setting prohibited	66 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	1 fAD	64 fAD	1 fAD	132/fCLK	Setting prohibited	33 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	1 fAD	64 fAD	1 fAD	66/fCLK	66 μs	16.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	0	Normal 2	fCLK/32	1 fAD	181 fAD	1 fAD	5856/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	183 μs
0	0	0	1				fCLK/16	1 fAD	181 fAD	1 fAD	2928/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	183 μs	91.5 μs
0	0	1	0				fCLK/8	1 fAD	181 fAD	1 fAD	1464/fCLK	Setting prohibited	Setting prohibited	183 μs	91.5 μs	45.75 μs
0	0	1	1				fCLK/4	1 fAD	181 fAD	1 fAD	732/fCLK	Setting prohibited	Setting prohibited	91.5 μs	45.75 μs	22.875 μs
0	1	0	0				fCLK/2	1 fAD	181 fAD	1 fAD	366/fCLK	Setting prohibited	Setting prohibited	45.75 μs	22.875 μs	11.4375 μs
0	1	0	1				fCLK	1 fAD	181 fAD	1 fAD	183/fCLK	Setting prohibited	Setting prohibited	22.875 μs	11.4375 μs	5.71875 μs
1	0	1	1				fCLK/4	1 fAD	181 fAD	1 fAD	732/fCLK	Setting prohibited	183 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	1 fAD	181 fAD	1 fAD	366/fCLK	Setting prohibited	91.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	1 fAD	181 fAD	1 fAD	183/fCLK	183 μs	45.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 42.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. Use normal mode 2 when the internal reference voltage 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 + side 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 16 - 3 A/D Conversion Time Selection (2/4)

2. When there is no A/D power supply stabilization wait time

Low-voltage mode 1 and 2 (for software trigger no-wait mode and hardware trigger no-wait mode)

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (fAD)	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)					
(AD M1)	(ADM0)										1.6 V ≤ AVREFP ≤ VDD ≤ 5.5 V	1.6 V ≤ AVREFP ≤ VDD ≤ 5.5 V	1.8 V ≤ AVREFP ≤ VDD ≤ 5.5 V	2.4 V ≤ AVREFP ≤ VDD ≤ 5.5 V	2.7 V ≤ AVREFP ≤ VDD ≤ 5.5 V	
	ADL SP	FR2	FR1	FR0	LV1						LV0	fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz
0	0	0	0	1	0	Low voltage 1	fCLK/32	1 fAD	80 fAD	1 fAD	2624/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	82 μs
0	0	0	1				fCLK/16	1 fAD	80 fAD	1 fAD	1312/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	82 μs	41 μs
0	0	1	0				fCLK/8	1 fAD	80 fAD	1 fAD	656/fCLK	Setting prohibited	Setting prohibited	82 μs	41 μs	20.5 μs
0	0	1	1				fCLK/4	1 fAD	80 fAD	1 fAD	328/fCLK	Setting prohibited	Setting prohibited	41 μs	20.5 μs	10.25 μs
0	1	0	0				fCLK/2	1 fAD	80 fAD	1 fAD	164/fCLK	Setting prohibited	Setting prohibited	20.5 μs	10.25 μs	5.125 μs
0	1	0	1				fCLK	1 fAD	80 fAD	1 fAD	82/fCLK	Setting prohibited	Setting prohibited	10.25 μs	5.125 μs	Setting prohibited
1	0	1	1				fCLK/4	1 fAD	80 fAD	1 fAD	328/fCLK	Setting prohibited	82 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	1 fAD	80 fAD	1 fAD	164/fCLK	Setting prohibited	41 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	1 fAD	80 fAD	1 fAD	82/fCLK	82 μs	20.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	fCLK/32	1 fAD	107 fAD	1 fAD	3488/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	109 μs
0	0	0	1				fCLK/16	1 fAD	107 fAD	1 fAD	1744/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	109 μs	54.5 μs
0	0	1	0				fCLK/8	1 fAD	107 fAD	1 fAD	872/fCLK	Setting prohibited	Setting prohibited	109 μs	54.5 μs	27.25 μs
0	0	1	1				fCLK/4	1 fAD	107 fAD	1 fAD	436/fCLK	Setting prohibited	Setting prohibited	54.5 μs	27.25 μs	13.625 μs
0	1	0	0				fCLK/2	1 fAD	107 fAD	1 fAD	218/fCLK	Setting prohibited	Setting prohibited	27.25 μs	13.625 μs	6.8125 μs
0	1	0	1				fCLK	1 fAD	107 fAD	1 fAD	109/fCLK	Setting prohibited	Setting prohibited	13.625 μs	6.8125 μs	Setting prohibited
1	0	1	1				fCLK/4	1 fAD	107 fAD	1 fAD	436/fCLK	Setting prohibited	109 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	1 fAD	107 fAD	1 fAD	218/fCLK	Setting prohibited	54.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	1 fAD	107 fAD	1 fAD	109/fCLK	109 μs	27.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (fAD) and conversion times (tCONV) described in 42.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.

Caution 4. When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (fAD) with a frequency no greater than 16 MHz.

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

Caution 5. When the internal reference voltage is selected as the + side 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 16 - 3 A/D Conversion Time Selection (3/4)

3. When there is A/D power supply stabilization wait time

Normal mode 1 and 2 (for software trigger wait mode and hardware trigger wait mode^{Note 1})

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (fAD)	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay Note 2	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)					
(AD M1)	(ADM0)										2.4 V ≤ AVREFP ≤ VDD ≤ 5.5 V					
ADL SP	FR2	FR1	FR0	LV1	LV0						fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz	
0	0	0	0	0	0	Normal 1	fCLK/32	4 fAD	64 fAD	4 fAD	2304/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72 μs
0	0	0	1				fCLK/16	4 fAD	64 fAD	4 fAD	1152/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	72 μs	36 μs
0	0	1	0				fCLK/8	6 fAD	64 fAD	4 fAD	592/fCLK	Setting prohibited	Setting prohibited	74 μs	37 μs	18.5 μs
0	0	1	1				fCLK/4	10 fAD	64 fAD	4 fAD	312/fCLK	Setting prohibited	Setting prohibited	39 μs	19.5 μs	9.75 μs
0	1	0	0				fCLK/2	18 fAD	64 fAD	4 fAD	172/fCLK	Setting prohibited	Setting prohibited	21.5 μs	10.75 μs	5.375 μs
0	1	0	1				fCLK	34 fAD	64 fAD	4 fAD	102/fCLK	Setting prohibited	Setting prohibited	12.75 μs	6.375 μs	3.1875 μs
1	0	1	1				fCLK/4	4 fAD	64 fAD	4 fAD	288/fCLK	Setting prohibited	72 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	4 fAD	64 fAD	4 fAD	144/fCLK	Setting prohibited	36 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	6 fAD	64 fAD	4 fAD	74/fCLK	74 μs	18.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	0	Normal 2	fCLK/32	4 fAD	181 fAD	4 fAD	6048/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	189 μs
0	0	0	1				fCLK/16	4 fAD	181 fAD	4 fAD	3024/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	189 μs	94.5 μs
0	0	1	0				fCLK/8	6 fAD	181 fAD	4 fAD	1528/fCLK	Setting prohibited	Setting prohibited	191 μs	95.5 μs	47.75 μs
0	0	1	1				fCLK/4	10 fAD	181 fAD	4 fAD	780/fCLK	Setting prohibited	Setting prohibited	97.5 μs	48.75 μs	24.375 μs
0	1	0	0				fCLK/2	18 fAD	181 fAD	4 fAD	406/fCLK	Setting prohibited	Setting prohibited	50.75 μs	25.375 μs	12.6875 μs
0	1	0	1				fCLK	34 fAD	181 fAD	4 fAD	219/fCLK	Setting prohibited	Setting prohibited	27.375 μs	13.6875 μs	6.84375 μs
1	0	1	1				fCLK/4	4 fAD	181 fAD	4 fAD	756/fCLK	Setting prohibited	189 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	4 fAD	181 fAD	4 fAD	378/fCLK	Setting prohibited	94.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	6 fAD	181 fAD	4 fAD	191/fCLK	191 μs	47.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected (see **Table 16 - 3 A/D Conversion Time Selection (1/4)**).

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 42.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

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

- Caution 3.** The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.
- Caution 4.** The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.
- Caution 5.** Use normal mode 2 when the internal reference voltage 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 + side 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 16 - 3 A/D Conversion Time Selection (4/4)

4. When there is A/D power supply stabilization wait time

Low-voltage mode 1 and 2 (for software trigger wait mode and hardware trigger wait mode^{Note 1})

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (fAD)	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay Note 2	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)					
(AD M1)	(ADM0)										1.6 V ≤ AVREFP ≤ VDD ≤ 5.5 V	1.6 V ≤ AVREFP ≤ VDD ≤ 5.5 V	1.8 V ≤ AVREFP ≤ VDD ≤ 5.5 V	2.4 V ≤ AVREFP ≤ VDD ≤ 5.5 V	2.7 V ≤ AVREFP ≤ VDD ≤ 5.5 V	
	ADL SP	FR2	FR1	FR0	LV1						LV0	fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 32 MHz
0	0	0	0	1	0	Low voltage 1	fCLK/32	4 fAD	80 fAD	4 fAD	2816/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	88 μs
0	0	0	1				fCLK/16	4 fAD	80 fAD	4 fAD	1408/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	88 μs	44 μs
0	0	1	0				fCLK/8	6 fAD	80 fAD	4 fAD	720/fCLK	Setting prohibited	Setting prohibited	90 μs	45 μs	22.5 μs
0	0	1	1				fCLK/4	10 fAD	80 fAD	4 fAD	376/fCLK	Setting prohibited	Setting prohibited	47 μs	23.5 μs	11.75 μs
0	1	0	0				fCLK/2	18 fAD	80 fAD	4 fAD	204/fCLK	Setting prohibited	Setting prohibited	25.5 μs	12.75 μs	6.375 μs
0	1	0	1				fCLK	34 fAD	80 fAD	4 fAD	118/fCLK	Setting prohibited	Setting prohibited	14.75 μs	7.375 μs	Setting prohibited
1	0	1	1				fCLK/4	4 fAD	80 fAD	4 fAD	352/fCLK	Setting prohibited	88 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	4 fAD	80 fAD	4 fAD	176/fCLK	Setting prohibited	44 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	6 fAD	80 fAD	4 fAD	90/fCLK	90 μs	22.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	fCLK/32	4 fAD	107 fAD	4 fAD	3680/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	115 μs
0	0	0	1				fCLK/16	4 fAD	107 fAD	4 fAD	1840/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	115 μs	57.5 μs
0	0	1	0				fCLK/8	6 fAD	107 fAD	4 fAD	936/fCLK	Setting prohibited	Setting prohibited	117 μs	58.5 μs	29.25 μs
0	0	1	1				fCLK/4	10 fAD	107 fAD	4 fAD	484/fCLK	Setting prohibited	Setting prohibited	60.5 μs	30.25 μs	15.125 μs
0	1	0	0				fCLK/2	18 fAD	107 fAD	4 fAD	258/fCLK	Setting prohibited	Setting prohibited	32.25 μs	16.125 μs	8.0625 μs
0	1	0	1				fCLK	34 fAD	107 fAD	4 fAD	145/fCLK	Setting prohibited	Setting prohibited	18.125 μs	9.0625 μs	Setting prohibited
1	0	1	1				fCLK/4	4 fAD	107 fAD	4 fAD	460/fCLK	Setting prohibited	115 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				fCLK/2	4 fAD	107 fAD	4 fAD	230/fCLK	Setting prohibited	57.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				fCLK	6 fAD	107 fAD	4 fAD	117/fCLK	117 μs	29.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Note 1. For the second and subsequent conversion in sequential conversion mode, the conversion start time and A/D power supply stabilization wait time do not occur after a hardware trigger is detected (see **Table 16 - 3 A/D Conversion Time Selection (2/4)**).

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (fAD).

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (fAD) and conversion times (tCONV) described in 42.6.1 A/D converter characteristics.

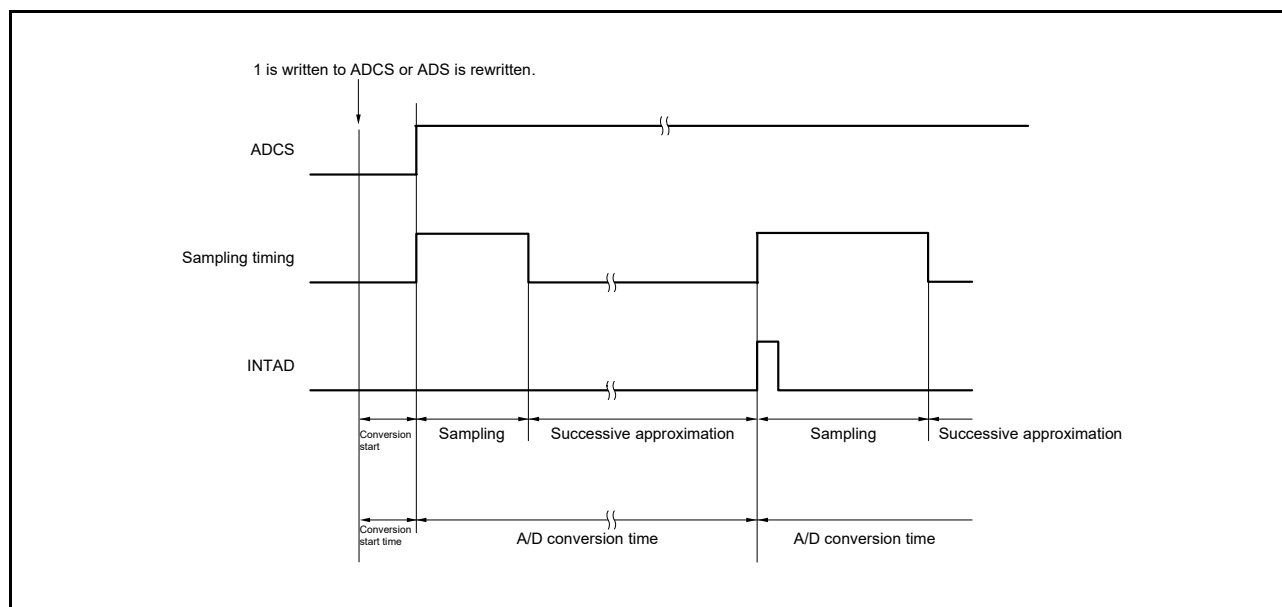
Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

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

- Caution 3.** The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the conversion times do not include clock frequency errors. Consider clock frequency errors when selecting the conversion time.
- Caution 4.** The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.
- Caution 5.** When the internal reference voltage 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 + side 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 16 - 6 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger No-wait Mode)



16.3.4 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. This register is used in combination with the ADTRSEX1 and ADTRSEX0 bits in the ADM1EX register. 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 16 - 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	ADLSP	ADTRS2	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	x	Software trigger no-wait mode or software trigger wait mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADLSP	fCLK input frequency setting
0	4 MHz < fCLK ≤ 32 MHz
1	1 MHz ≤ fCLK ≤ 4 MHz

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS 2	ADTRS 1	ADTRS 0	ADTRS EX1	ADTRS EX0	Selection of the hardware trigger signal
0	0	0	x	x	Timer channel 01 count or capture end interrupt signal (INTTM01)
0	1	0	x	x	Realtime clock interrupt signal (INTRTC)
0	1	1	x	x	Trigger signal (ADITL0) for the A/D converter from the 32-bit interval timer output
1	0	0	0	0	Event input from the ELCL
1	0	0	0	1	16-bit timer KB40 A/D trigger signal
1	0	0	1	0	16-bit timer KB41 A/D trigger signal
1	0	0	1	1	16-bit timer KB42 A/D trigger signal
Other than above					Setting prohibited

Caution 1. Only rewrite the value of the ADM1 register while conversion operation 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 cycles + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 fCLK clock cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 5 μs

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

Caution 3. In modes other than the SNOOZE mode, input of the next INTRTC or INTITL will not be recognized as a valid hardware trigger for up to four fCLK cycles after the first INTRTC or INTITL is input.

Remark 1. ×: Don't care

Remark 2. fCLK: CPU/peripheral hardware clock frequency

16.3.5 A/D converter mode extension register 1 (ADM1EX)

The ADM1EX register is used to specify the hardware trigger signal. This register is used in combination with the ADTRS2 to ADTRS0 bits in the ADM1 register. The ADM1EX register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 16 - 8 Format of A/D Converter Mode Extension Register 1 (ADM1EX)

Address: F029EH
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADM1EX	0	0	0	0	0	0	ADTRSEX1	ADTRSEX0

ADTRS 2	ADTRS 1	ADTRS 0	ADTRS EX1	ADTRS EX0	Selection of the hardware trigger signal
0	0	0	×	×	Timer channel 01 count or capture end interrupt signal (INTTM01)
0	1	0	×	×	Realtime clock interrupt signal (INTRTC)
0	1	1	×	×	Trigger signal (ADITL0) for the A/D converter from the 32-bit interval timer output
1	0	0	0	0	Event input from the ELCL
1	0	0	0	1	16-bit timer KB40 A/D trigger signal
1	0	0	1	0	16-bit timer KB41 A/D trigger signal
1	0	0	1	1	16-bit timer KB42 A/D trigger signal
Other than above					Setting prohibited

Remark ×: Don't care

16.3.6 A/D converter mode register 2 (ADM2)

The ADM2 register is used to select the + side and - side 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 16 - 9 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	ADTYP1	ADTYP0

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from VDD
0	1	Supplied from P21/ANI0/AVREFP
1	0	Supplied from the internal reference voltage ^{Note 1}
1	1	Discharged

• Use the following procedures to rewrite the ADREFP1 and ADREFP0 bits.

<1> Set ADCE = 0

<2> Wait for at least 0.2 μ s.

<3> Set ADREFP1 and ADREFP0 to 11B.

This step is only necessary when the values of ADREFP1 and ADREFP0 are changed to 10B.

<4> Reference voltage discharge time: 1 μ s

This step is only necessary when the values of ADREFP1 and ADREFP0 are changed to 10B.

<5> Change the values of ADREFP1 and ADREFP0

<6> Wait for reference voltage stabilization time A.

<7> Set ADCE = 1

<8> Wait for reference voltage stabilization time B.

Reference voltage stabilization times A and B

Stabilization of the reference voltage will require the following waiting times; the duration of time A depends on the settings of ADREFP1 and ADREFP0.

- ADREFP1 and ADREFP0 = 10B
 - A = 5 μ s
 - B = 1 μ s + 2 cycles of the conversion clock (f_{AD})
- ADREFP1 and ADREFP0 = 00B or 01B
 - A = 4.8 μ s
 - B = 1 μ s + 2 cycles of the conversion clock (f_{AD})

Note that the waiting time of A = 4.8 μ s is not required when you set the reference voltage for the first time.

After waiting for reference voltage stabilization time B, start the A/D conversion.

• When ADREFP1 and ADREFP0 are set to 10B, 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 - side reference voltage of the A/D converter
0	Supplied from VSS
1	Supplied from P20/ANI1/AVREFM/EI20/(TI07)/(INTP7)/(PCLBUZ1)

Figure 16 - 9 Format of A/D Converter Mode Register 2 (ADM2) (2/2)

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register $\leq$ the ADCR0 register $\leq$ the ADUL register (AREA 1).
1	The interrupt signal (INTAD) is output when the ADCR0 register $<$ the ADLL register (AREA 2) or the ADUL register $<$ the ADCR0 register (AREA 3).

Figure 16 - 10 shows the generation range of the interrupt signal (INTAD) for AREA 1 to AREA 3.

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode.
1	Use the SNOOZE mode.

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 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.
- When using the SNOOZE mode, set AWC to 0 in software trigger wait mode, and set AWC to 1 in hardware trigger wait mode.
- Using the SNOOZE mode in the software trigger no-wait mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode in hardware trigger no-wait mode in sequential conversion mode is prohibited.
- When using the SNOOZE mode, specify a hardware trigger interval of at least "shift time to SNOOZE mode^{Note} + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 fCLK clock cycles + 5 μ s".
- Even when using the 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.

ADTYP1	ADTYP0	Selection of the A/D conversion resolution
0	0	10-bit resolution
0	1	8-bit resolution
1	0	12-bit resolution
1	1	Setting prohibited

Note 1. For the internal reference voltage, see **Section 42 Electrical Characteristics**.

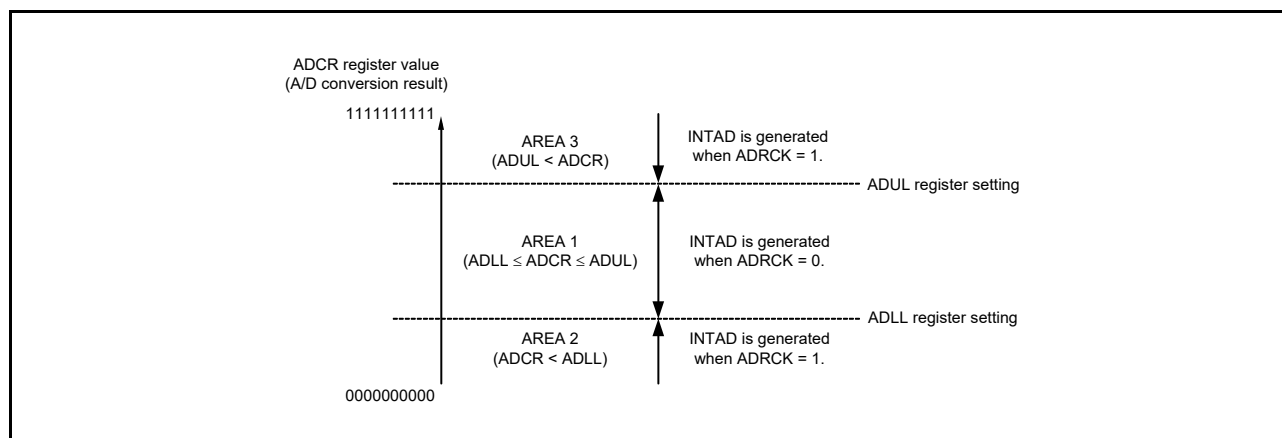
Note 2. Refer to "Transition time from STOP mode to SNOOZE mod" in **27.3.3 SNOOZE mode**.

Caution 1. Only rewrite the value of the ADM2 register while conversion operation 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 and ADREFP0 = 10B), the A/D converter reference voltage current (IADREF) indicated in 42.3.2 Supply current characteristics 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 16 - 10 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR0 or ADCR0H register.

16.3.7 12-bit/10-bit A/D conversion result register (ADCR0)

ADCR0 is a 16-bit register that holds the A/D conversion result. When A/D conversion with 12-bit resolution is selected, the higher 4 bits are fixed to 0. When A/D conversion with 10-bit resolution is selected, 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 ADCR0 register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H. The conversion results are stored in the ADCR and ADCR0 registers^{Note}.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (set up by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 10**), the result is not stored.

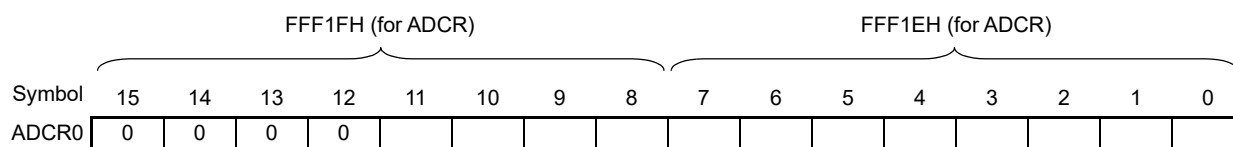
Figure 16 - 11 Format of 12-bit/10-bit A/D Conversion Result Register (ADCR0)

Address: FFF1FH, FFF1EH (ADCR)^{Note}, F0021H, F0020H (ADCR0)^{Note}

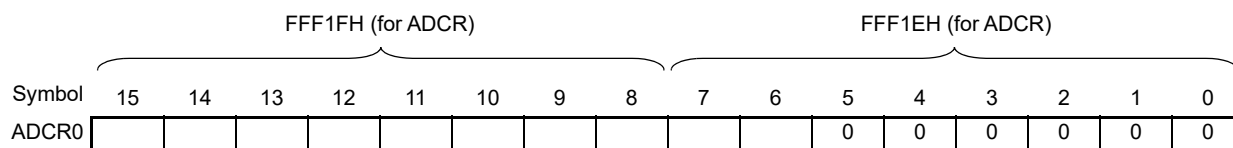
After reset: 0000H

R/W: R

- When A/D conversion with 12-bit resolution is selected



- When A/D conversion with 10-bit resolution is selected



Note The contents of the ADCR register are stored in the ADCR0 register.

Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP1 and ADTYP0 bits of A/D converter mode register 2 (ADM2) are respectively set to 01) and the ADCR0 register is read, 0 is read from the bits other than the higher 8 bits.

Caution 2. When the ADCR0 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 ADCR0 register. When A/D conversion with 12-bit resolution is selected, the higher 12 bits of the conversion result are read in order starting at bit 11 of the ADCR0 register.

Caution 3. The contents of the ADCR0 register may become undefined when writing to any of the following registers.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register (ADS)
- Port mode control A registers 1 and 2 (PMCA1 and PMCA2)
- LCD port function registers 4 and 5 (PFSEG4 and PFSEG5)

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

16.3.8 8-bit A/D conversion result register (ADCR0H)

ADCR0H is an 8-bit register that holds the A/D conversion result. The higher 8 bits of 12-bit resolution are stored^{Note}. The ADCR0H register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

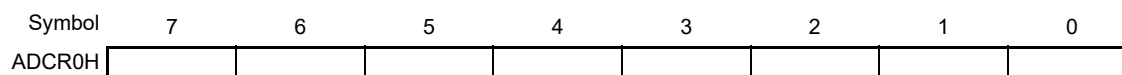
Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (set up by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 10**), the result is not stored.

Figure 16 - 12 Format of 8-bit A/D Conversion Result Register (ADCR0H)

Address: FFF1FH (ADCRH)^{Note}, F0021H (ADCR0H)^{Note}

After reset: 00H

R/W: R



Note The contents of the ADCRH register are stored in the ADCR0H register.

Caution The contents of the ADCR0H register may become undefined when writing to any of the following registers.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register (ADS)
- Port mode control A registers 1 and 2 (PMCA1 and PMCA2)
- LCD port function registers 4 and 5 (PFSEG4 and PFSEG5)

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

16.3.9 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 16 - 13 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

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P21/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P20/ANI1/AVREFM pin
0	1	0	0	0	0	ANI16	P22/ANI16 pin
0	1	0	0	0	1	ANI17	P23/ANI17 pin ^{Note 1}
0	1	0	0	1	0	ANI18	P24/ANI18 pin ^{Note 1}
0	1	0	0	1	1	ANI19	P25/ANI19 pin ^{Note 1}
0	1	0	1	0	0	ANI20	P26/ANI20 pin
0	1	0	1	0	1	ANI21	P27/ANI21 pin
0	1	0	1	1	0	ANI22	P10/ANI22 pin
0	1	0	1	1	1	ANI23	P11/ANI23 pin
0	1	1	0	0	0	ANI24	P12/ANI24 pin
0	1	1	0	0	1	ANI25	P13/ANI25 pin ^{Note 2}
0	1	1	0	1	0	ANI26	P14/ANI26 pin ^{Note 3}
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 4}
Other than the above						Setting prohibited	

Note 1. This is only applicable to the 80- and 100-pin products.

Note 2. This is only applicable to the 48- to 100-pin products.

Note 3. This is only applicable to the 52- to 100-pin products.

Note 4. For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Caution 1. Be sure to clear bits 6 and 5 to 0.

Caution 2. Set the channel that is specified as the analog input by a PMCAxx or PFSEGx register to the input mode by using port mode register 1 or 2 (PM1 or PM2).

Caution 3. When using the ADS register to specify an input channel, do not select a pin that is specified as digital I/O by port mode control A register 1 or 2 (PMCA1 or PMCA2) or LCD port function register 4 or 5 (PFSEG4 or PFSEG5).

Caution 4. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

(Cautions are listed on the next page.)

- Caution 5.** If using AVREFP as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.
- Caution 6.** If using AVREFM as the – side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.
- Caution 7.** When the setting of the ADISS bit is 1, the internal reference voltage cannot be used for the + side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 16.7.5 Example of using the ADC when selecting the temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU, and software trigger no-wait mode and one-shot conversion mode. For details about the internal reference voltage, see Section 42 Electrical Characteristics.
- 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 42.3.2 Supply current characteristics will be added.
- Caution 9.** When the setting of the ADISS bit is 1, the hardware trigger wait mode and one-shot conversion mode cannot be used at the same time.
- Caution 10.** When the setting of the ADISS bit is 1, the software trigger wait mode and one-shot conversion mode cannot be used at the same time.

16.3.10 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 by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 16 - 10 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 16 - 14 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

16.3.11 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 by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 16 - 10 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 16 - 15 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 A/D conversion result register ADCR0[15:8] value is compared with the values in the ADUL and ADLL registers. When A/D conversion with 12-bit resolution is selected, the A/D conversion result register ADCR0[11:4] value is 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 register must be greater than that of the ADLL register.

16.3.12 A/D test register (ADTES)

The ADTES register is used to select the + side reference voltage or - side reference voltage for the converter, an analog input channel (ANLxx), the temperature sensor output voltage, the internal reference voltage^{Note}, or the TSCAP voltage of the CTSU as the target for A/D conversion. When using this register to test the converter, set as follows.

- For zero-scale measurement, select the - side reference voltage as the target for conversion.
- For full-scale measurement, select the + side 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.

Note For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Figure 16 - 16 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/temperature sensor output voltage/internal reference voltage ^{Note} /TSCAP voltage of the CTSU (set by analog input channel specification register (ADS))
1	0	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 and ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Note For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Caution Be sure to clear bits 7 to 2 to 0.

16.3.13 Registers for controlling the port functions multiplexed with the analog inputs of the A/D converter

Set the following registers for controlling the port functions multiplexed with the analog inputs of the A/D converter.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

16.4 A/D Converter 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 11 of the successive approximation register (SAR) is set to 1. 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 of the SAR register remains set to 1. If the analog input is smaller than 1/2 AVREF, the MSB is reset to 0.
 - <5> Next, bit 10 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 11, as described below.
 - Bit 11 = 1: (3/4) AVREF
 - Bit 11 = 0: (1/4) AVREF
- The voltage tap and sampled voltage are compared and bit 10 of the SAR register is manipulated as follows.
- Sampled voltage $\geq$ Voltage tap: Bit 10 = 1
 - Sampled voltage < Voltage tap: Bit 10 = 0
- <6> Comparison is continued in this way up to bit 0 of the SAR register.
 - <7> Upon completion of the comparison of 12 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR0, ADCR0H) 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 16 - 10 ADRCK Bit Interrupt Signal Generation Range**), the A/D conversion end interrupt request signal is not generated and no A/D conversion results are stored in the ADCR0 and ADCR0H registers.

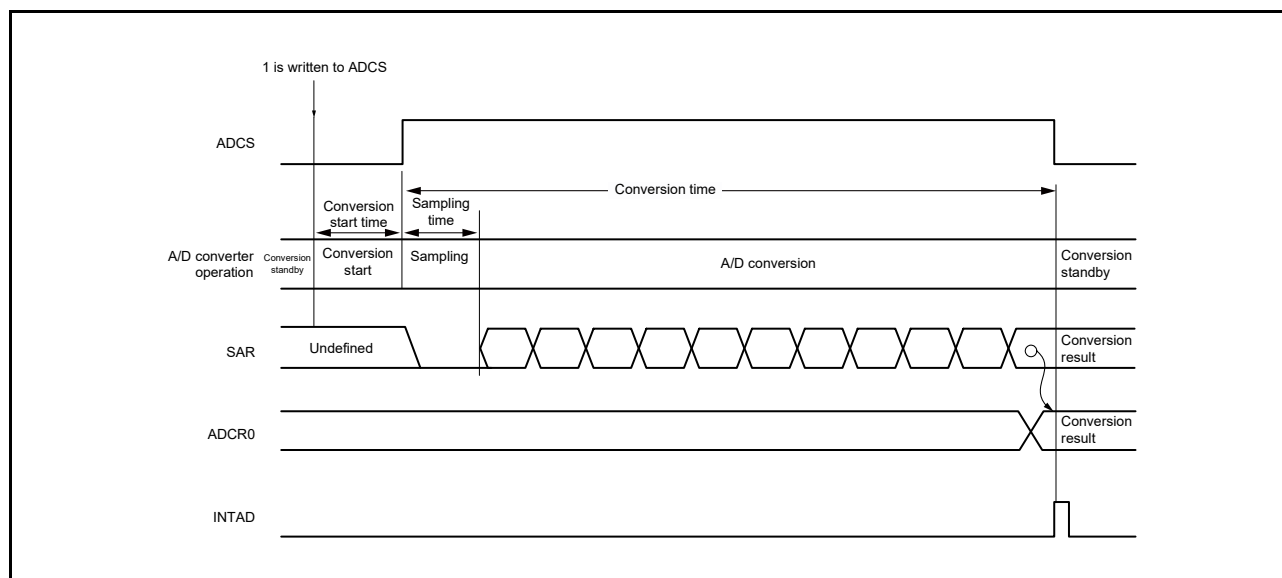
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR0 register (16 bits): Store 12-bit/10-bit A/D conversion value
- ADCR0H register (8 bits): Store 8-bit A/D conversion value

Remark 2. AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage, and VDD. For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Figure 16 - 17 Conversion Operation of A/D Converter (Software Trigger No-wait 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 the 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 value of the A/D conversion result register (ADCR0, ADCR0H) following a reset is 00H or 0000H.

16.5 Input Voltage and Conversion Results

The relationship between the analog voltage input to the analog input pins (ANI0, ANI1, ANI16 to ANI26) and the theoretical A/D conversion result (stored in the 12-bit/10-bit A/D conversion result register (ADCR0)) is shown by the following expression.

$$\text{ADCR0} = \text{INT} \left(\frac{V_{\text{AIN}}}{V_{\text{REF}}} \times 4096 + 0.5 \right)$$

or

$$(\text{ADCR0} - 0.5) \times \frac{V_{\text{REF}}}{4096} \leq V_{\text{AIN}} < (\text{ADCR0} + 0.5) \times \frac{V_{\text{REF}}}{4096}$$

where, INT(): Function which returns integer part of value in parentheses

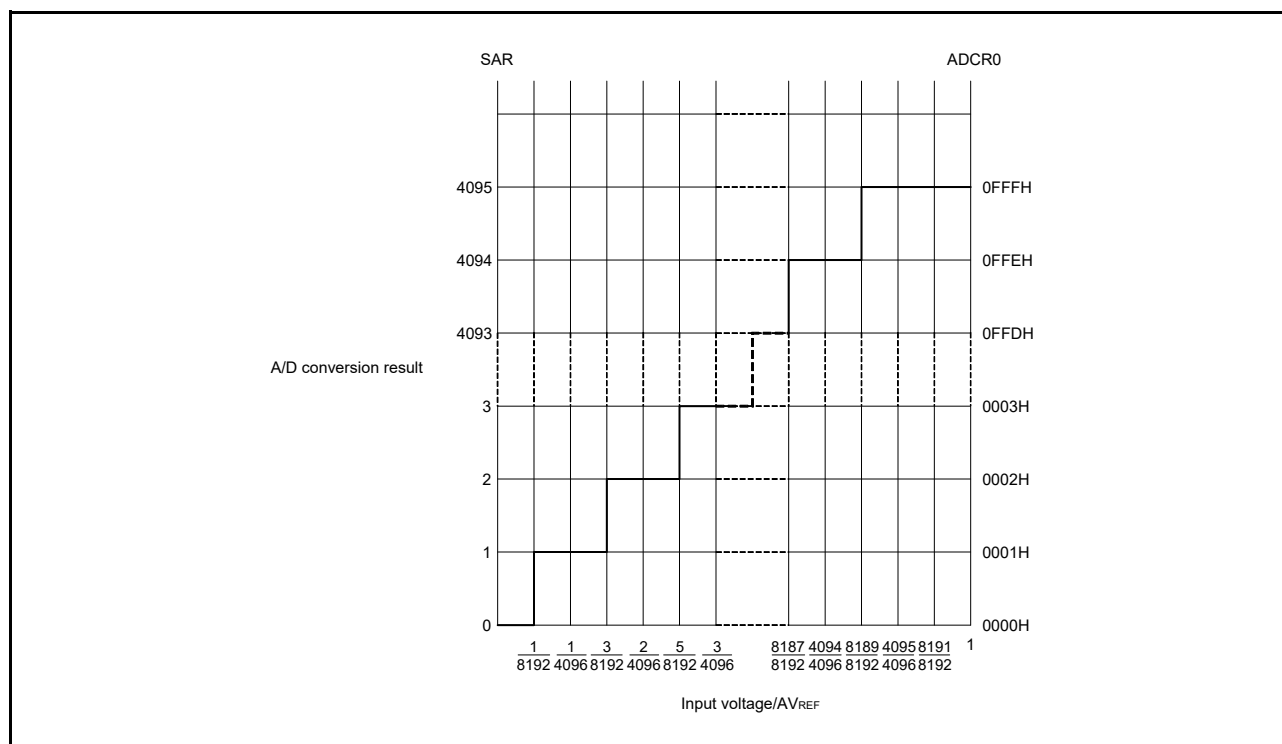
V_{AIN} : Analog input voltage

V_{REF} : V_{REF} pin voltage

ADCR0: 12-bit/10-bit A/D conversion result register (ADCR0) value

Figure 16 - 18 shows the relationship between analog input voltage and A/D conversion result.

Figure 16 - 18 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark V_{REF} : The + side reference voltage of the A/D converter. This can be selected from V_{REFP} , the internal reference voltage, and V_{DD} . For the internal reference voltage, see **Section 42 Electrical Characteristics**.

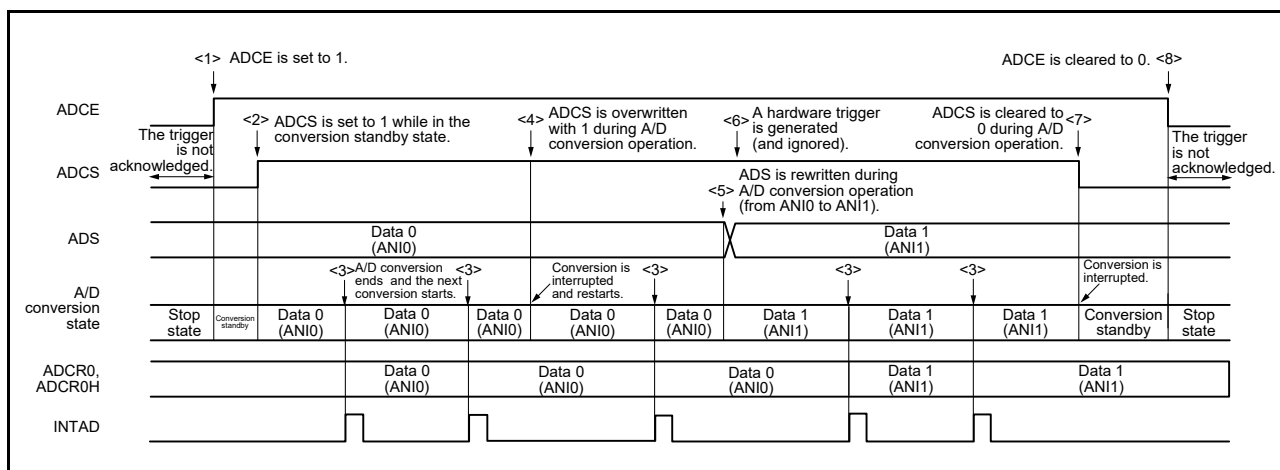
16.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **16.7 A/D Converter Setup Flowchart**.

16.6.1 Software trigger no-wait mode (sequential conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state.
- <2> After the software counts up to the stabilization wait time ($1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD})), 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, ADCR0, and ADCR0H), 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 written 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.

Figure 16 - 19 Example of Software Trigger No-wait Mode (Sequential Conversion Mode) Operation Timing

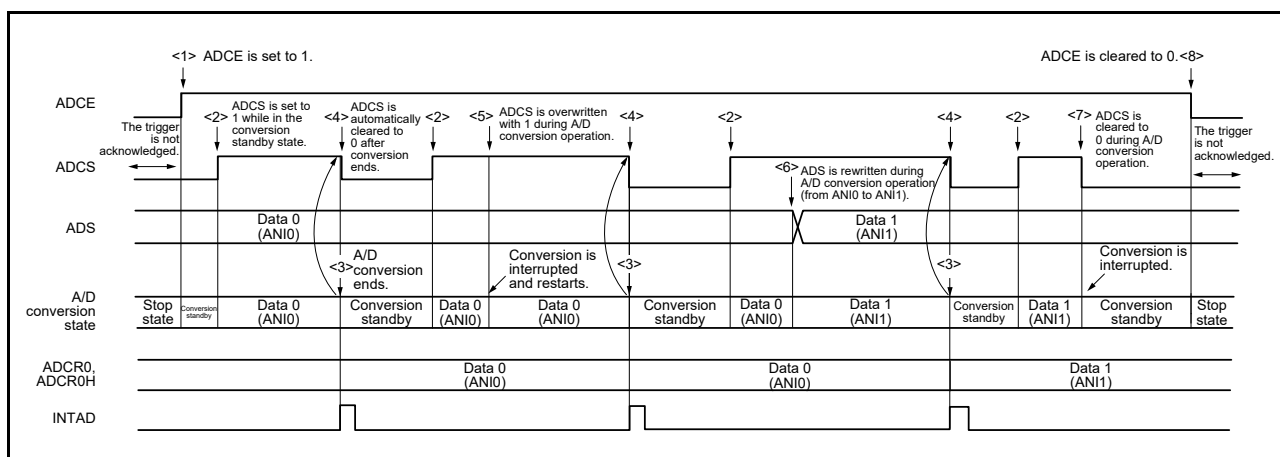


Caution When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.2 Software trigger no-wait mode (one-shot conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state.
- <2> After the software counts up to the stabilization wait time ($1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD})), 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, ADCR0, and ADCR0H), 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. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Figure 16 - 20 Example of Software Trigger No-wait Mode (One-shot Conversion Mode) Operation Timing

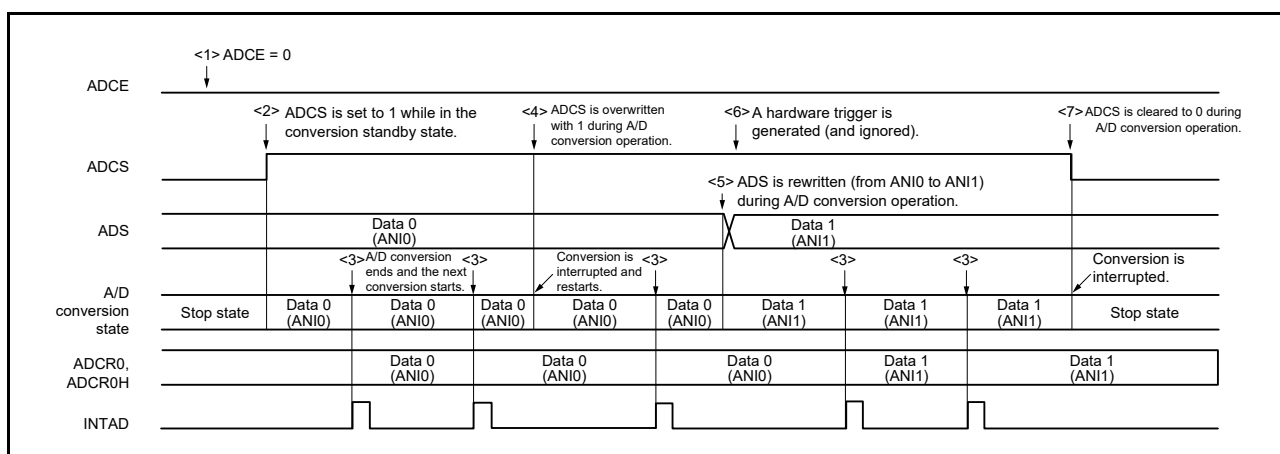


Caution When <5> or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.3 Software trigger wait mode (sequential conversion mode)

- <1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).
- <2> If ADCS is set to 1 in the stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), 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 stop state.

Figure 16 - 21 Example of Software Trigger Wait Mode (Sequential Conversion Mode) Operation Timing

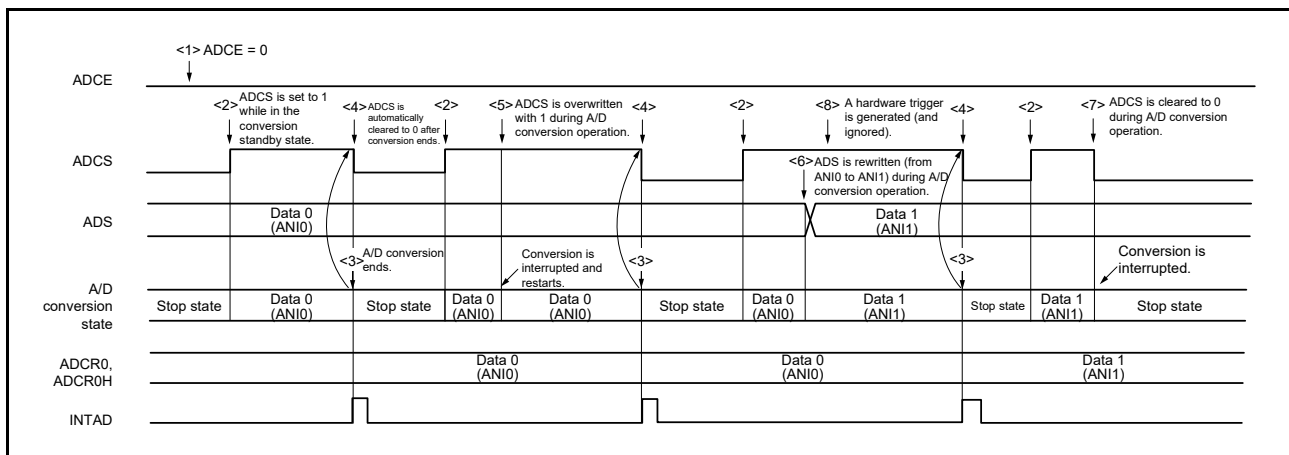


Caution When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.4 Software trigger wait mode (one-shot conversion mode)

- <1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stop state).
- <2> If ADCS is set to 1 in the stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).
- <3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), 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> 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.
- <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 stop state.
- <8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 16 - 22 Example of Software Trigger Wait Mode (One-shot Conversion Mode) Operation Timing

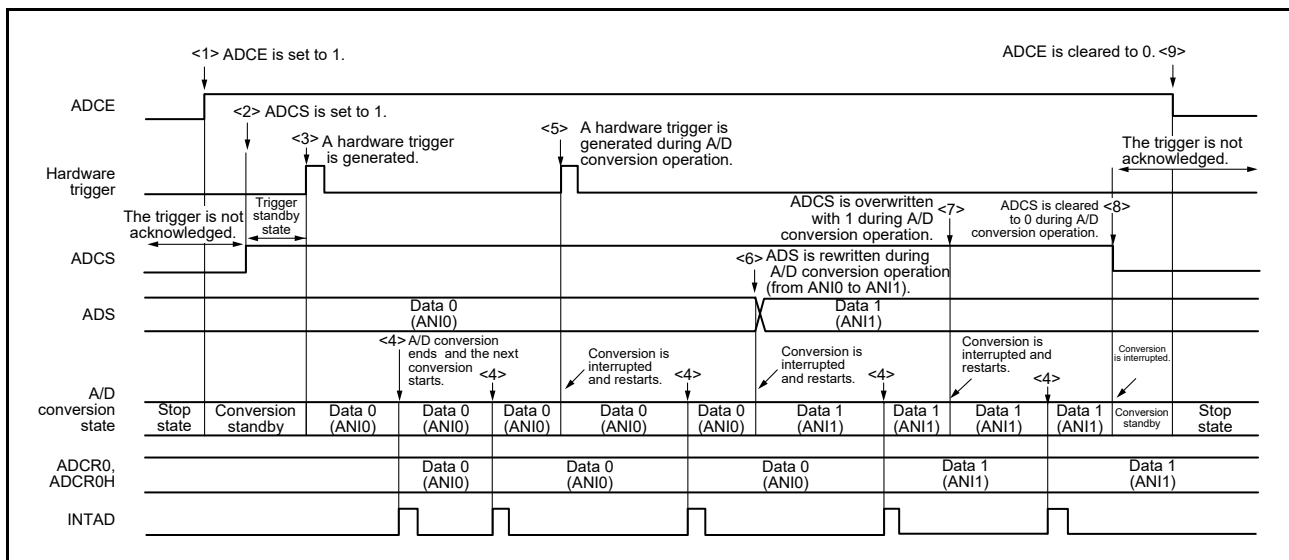


- Caution 1.** When <5> or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)
- Caution 2.** The setting of ADISS being 1 (the input source is temperature sensor output voltage or internal reference voltage) cannot be used in the software trigger wait mode (one-shot conversion mode).

16.6.5 Hardware trigger no-wait mode (sequential conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state.
- <2> After the software counts up to the stabilization wait time ($1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD})), 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, ADCR0, and ADCR0H), 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, input of a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 23 Example of Hardware Trigger No-wait Mode (Sequential Conversion Mode) Operation Timing

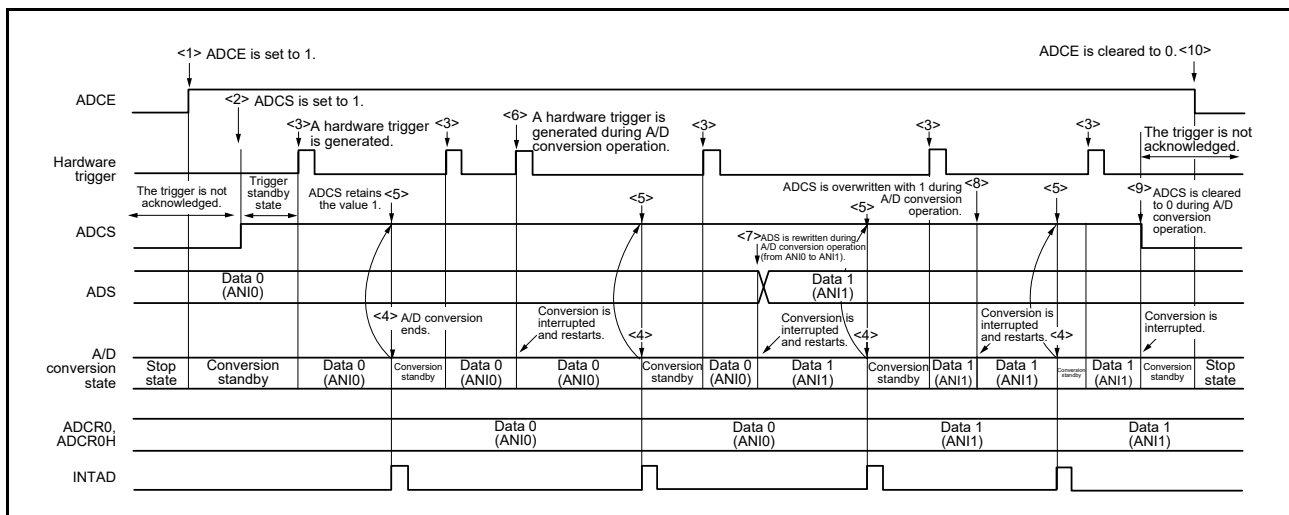


Caution When <5>, <6>, or <7> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.6 Hardware trigger no-wait mode (one-shot conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the standby state.
- <2> After the software counts up to the stabilization wait time ($1\ \mu\text{s} + 2$ cycles of the conversion clock (f_{AD})), 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, ADCR0, and ADCR0H), 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, input of a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 24 Example of Hardware Trigger No-wait Mode (One-shot Conversion Mode) Operation Timing

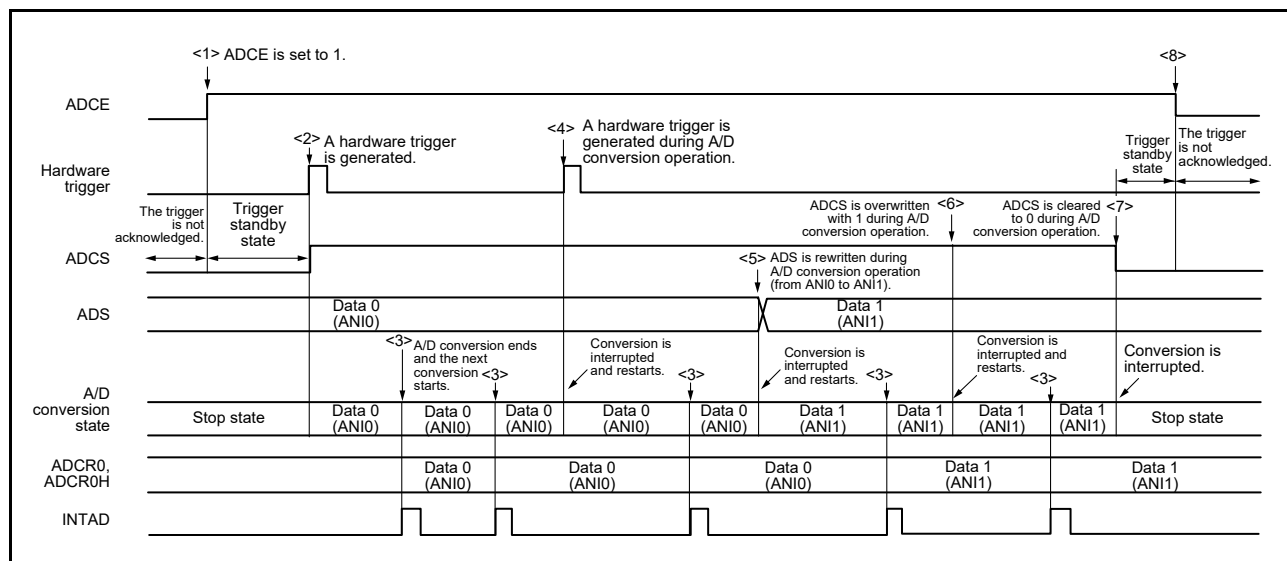


Caution When <6>, <7>, or <8> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.7 Hardware trigger wait mode (sequential conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters 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, ADCR0, and ADCR0H), 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.
- <8> When ADCE is cleared to 0 while waiting for a hardware trigger, the A/D converter enters the stop state. When ADCE = 0, input of a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 25 Example of Hardware Trigger Wait Mode (Sequential Conversion Mode) Operation Timing

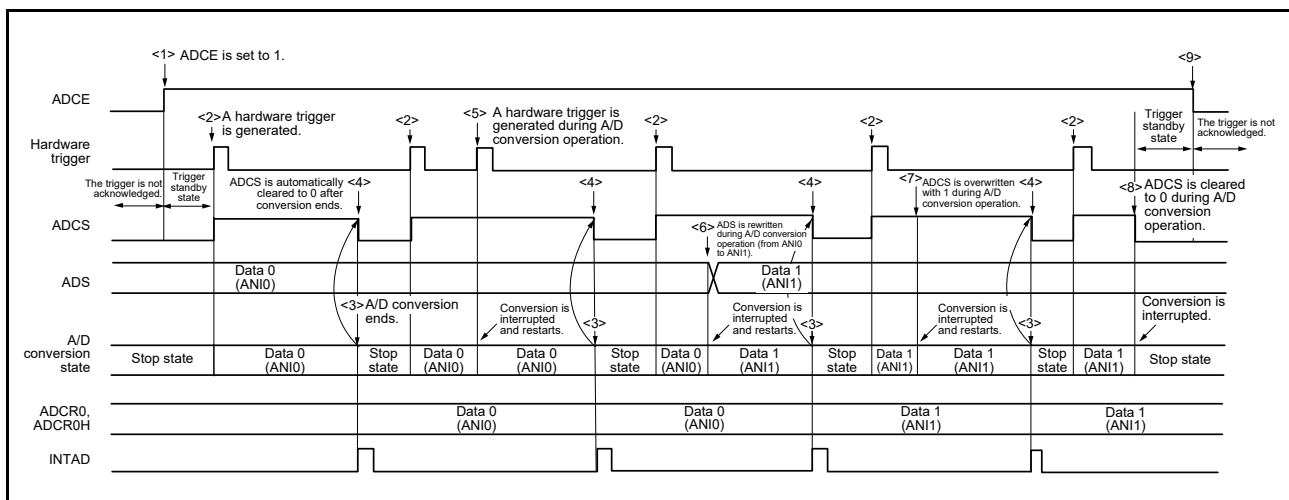


Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

16.6.8 Hardware trigger wait mode (one-shot conversion mode)

- <1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters 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, ADCR0, and ADCR0H), 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.
- <9> When ADCE is cleared to 0 while waiting for a hardware trigger, the A/D converter enters the stop state. When ADCE = 0, input of a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 26 Example of Hardware Trigger Wait Mode (One-shot Conversion Mode) Operation Timing



Caution 1. When <5>, <6>, or <7> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. (See Table 16 - 3 A/D Conversion Time Selection (3/4) and Table 16 - 3 A/D Conversion Time Selection (4/4).)

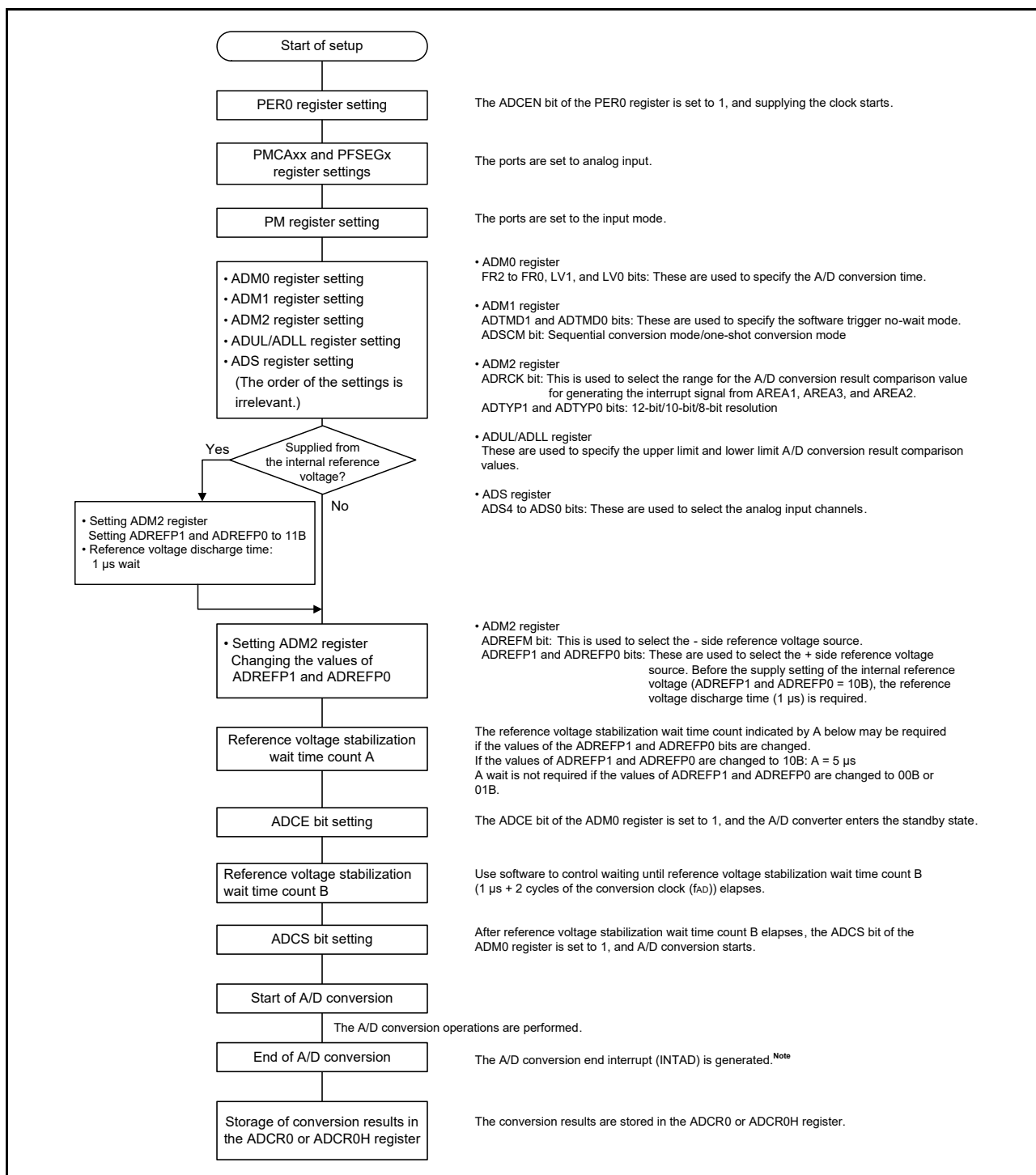
Caution 2. The setting of ADISS being 1 (the input source is temperature sensor output voltage or internal reference voltage) cannot be used in the hardware trigger wait mode (one-shot conversion mode).

16.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

16.7.1 Setting up software trigger no-wait mode

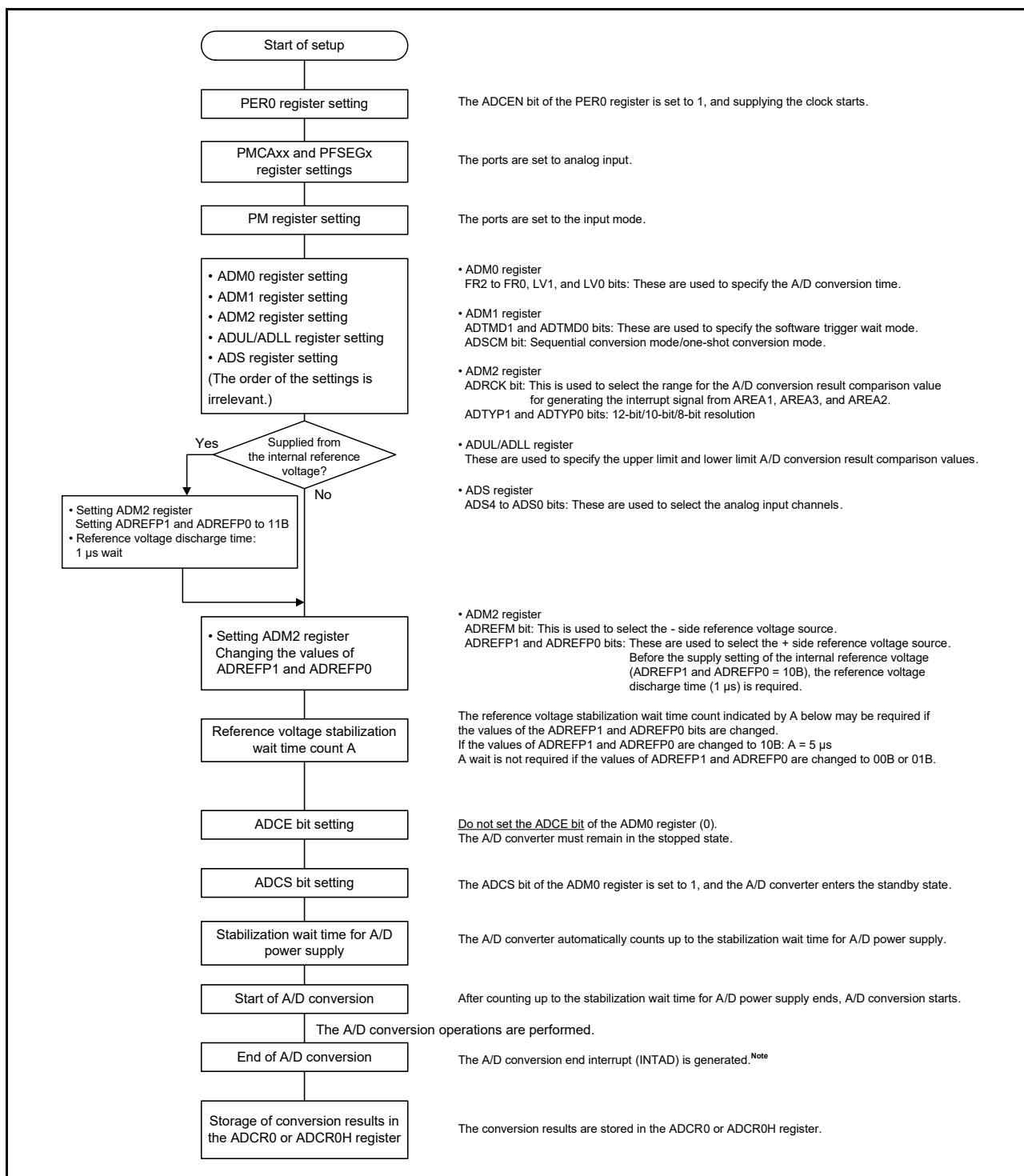
Figure 16 - 27 Setting up Software Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

16.7.2 Setting up software trigger wait mode

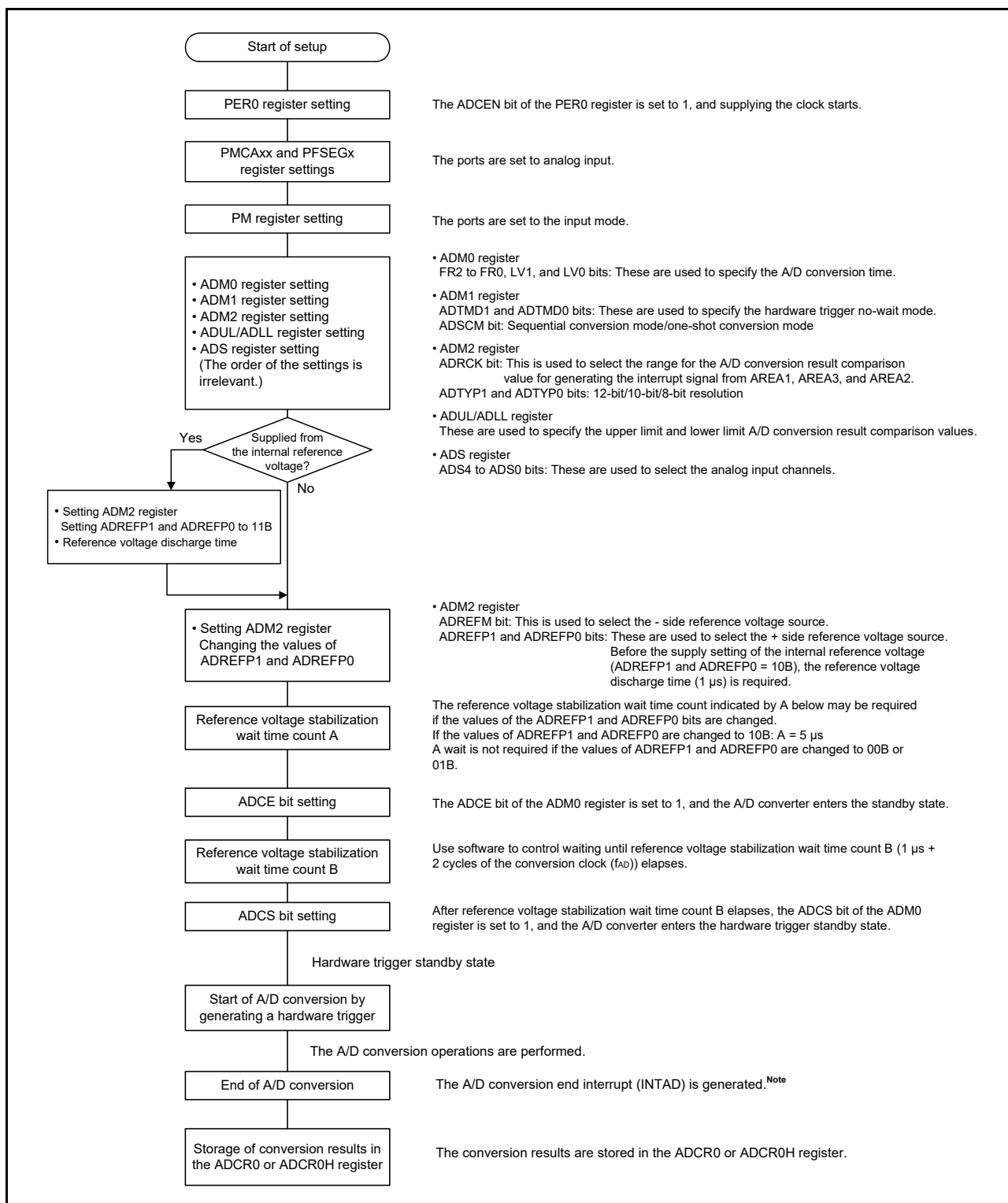
Figure 16 - 28 Setting up Software Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

16.7.3 Setting up hardware trigger no-wait mode

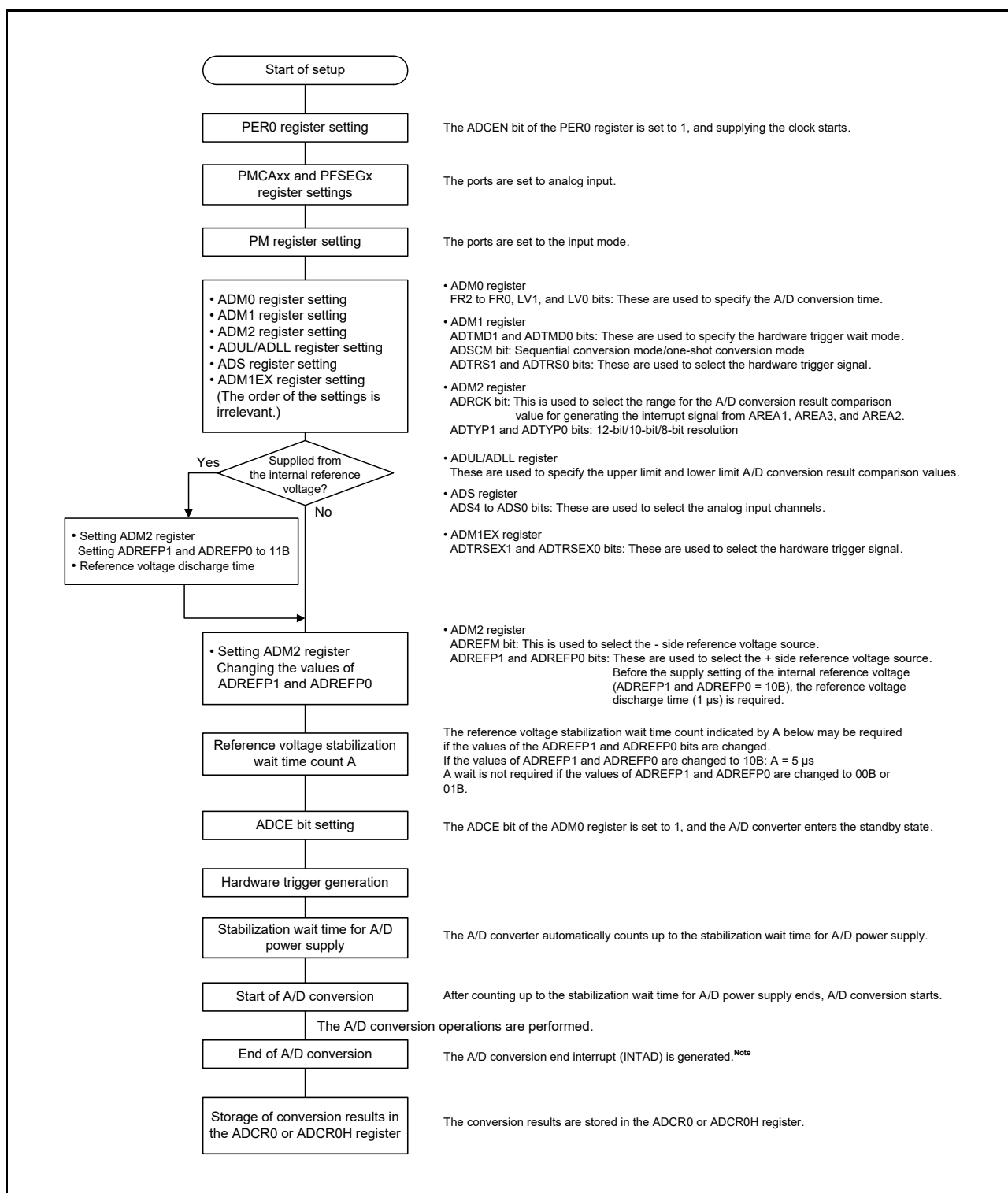
Figure 16 - 29 Setting up Hardware Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

16.7.4 Setting up hardware trigger wait mode

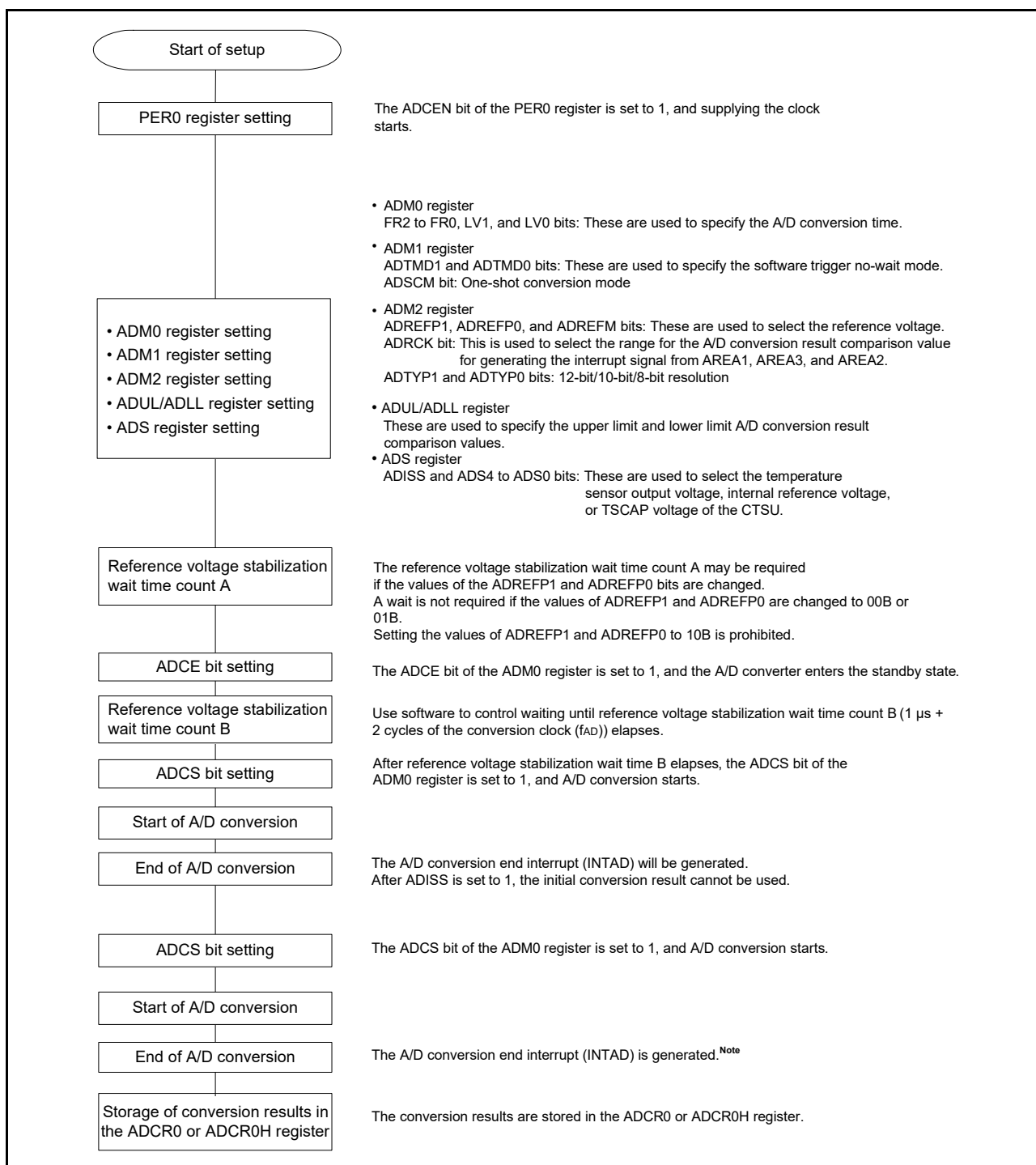
Figure 16 - 30 Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

16.7.5 Example of using the ADC when selecting the temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU, and software trigger no-wait mode and one-shot conversion mode

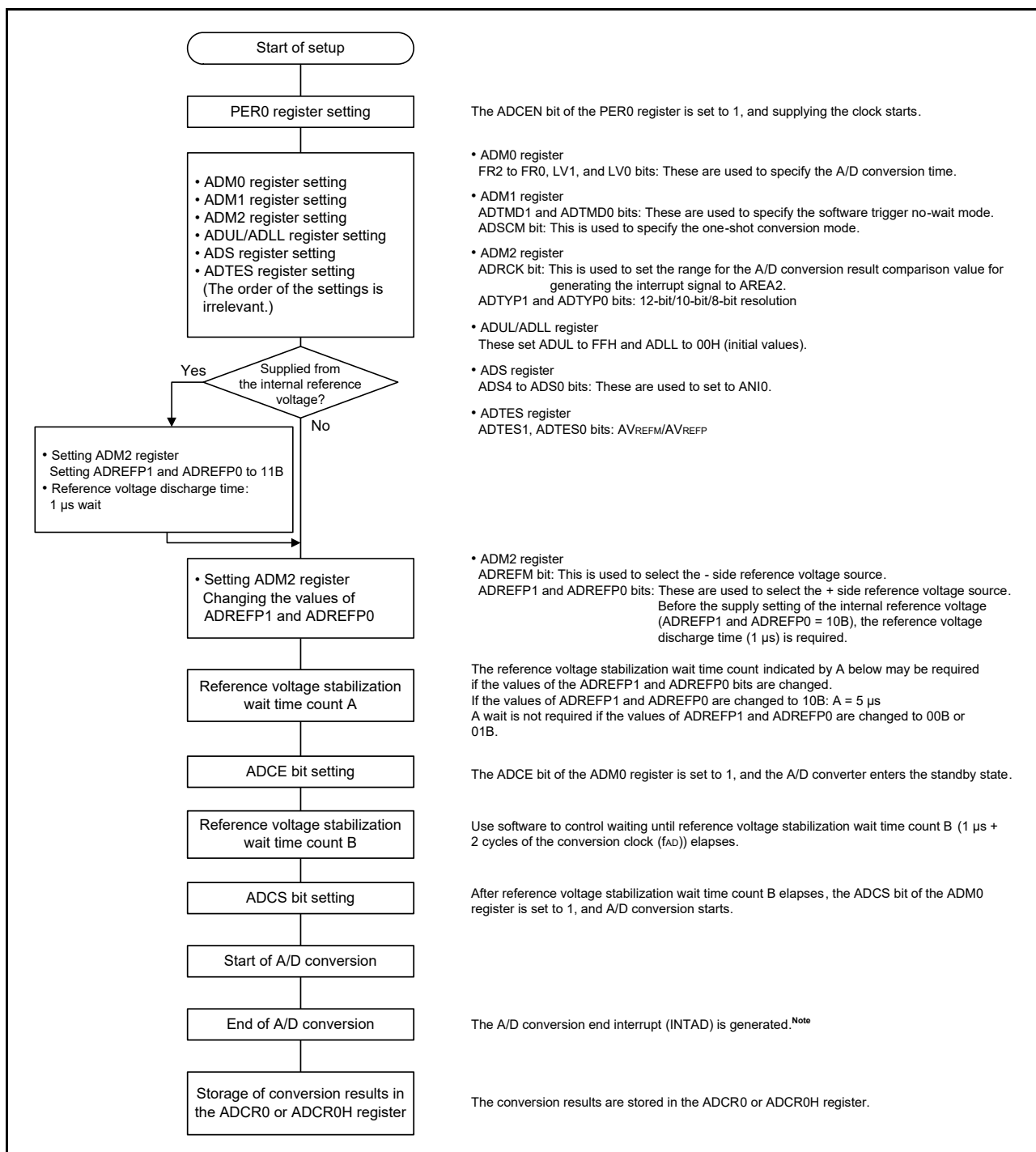
Figure 16 - 31 Setup When Temperature Sensor Output Voltage/Internal Reference Voltage/TSCAP Voltage of the CTSU Is Selected



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

16.7.6 Setting up test mode

Figure 16 - 32 Setting up Test Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR0 or ADCR0H register.

Caution For the procedure for testing the A/D converter, see 32.3.10 Testing of the A/D converter.

16.8 SNOOZE Mode

In the SNOOZE mode, A/D conversion is triggered by inputting a software trigger or a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.

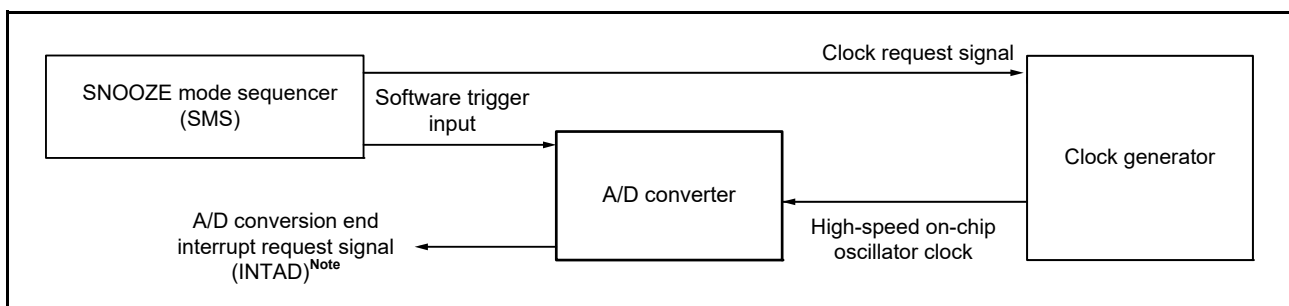
16.8.1 A/D conversion by inputting a software trigger

In the SNOOZE mode, A/D conversion is triggered by inputting a software trigger. A software trigger generated by the SNOOZE mode sequencer (SMS) is used as an input trigger for A/D conversion. When performing A/D conversion by inputting a software trigger in SNOOZE mode, only the following two conversion modes can be used.

- Software trigger wait mode (one-shot conversion mode)
- Software trigger wait mode (sequential conversion mode)

Caution 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.

Figure 16 - 33 Block Diagram When Using SNOOZE Mode in Software Trigger Wait Mode



When using the SNOOZE mode, the initial setting of each register is specified before switching to the STOP mode (for details about these settings, see **16.7.2 Setting up software trigger wait mode**). If a software trigger (ADCS = 1) 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.

Note Clear the setting of the A/D conversion result comparison function (ADRCK bit and ADUL/ADLL register) to the initial value so that an interrupt signal is generated each time conversion is completed.

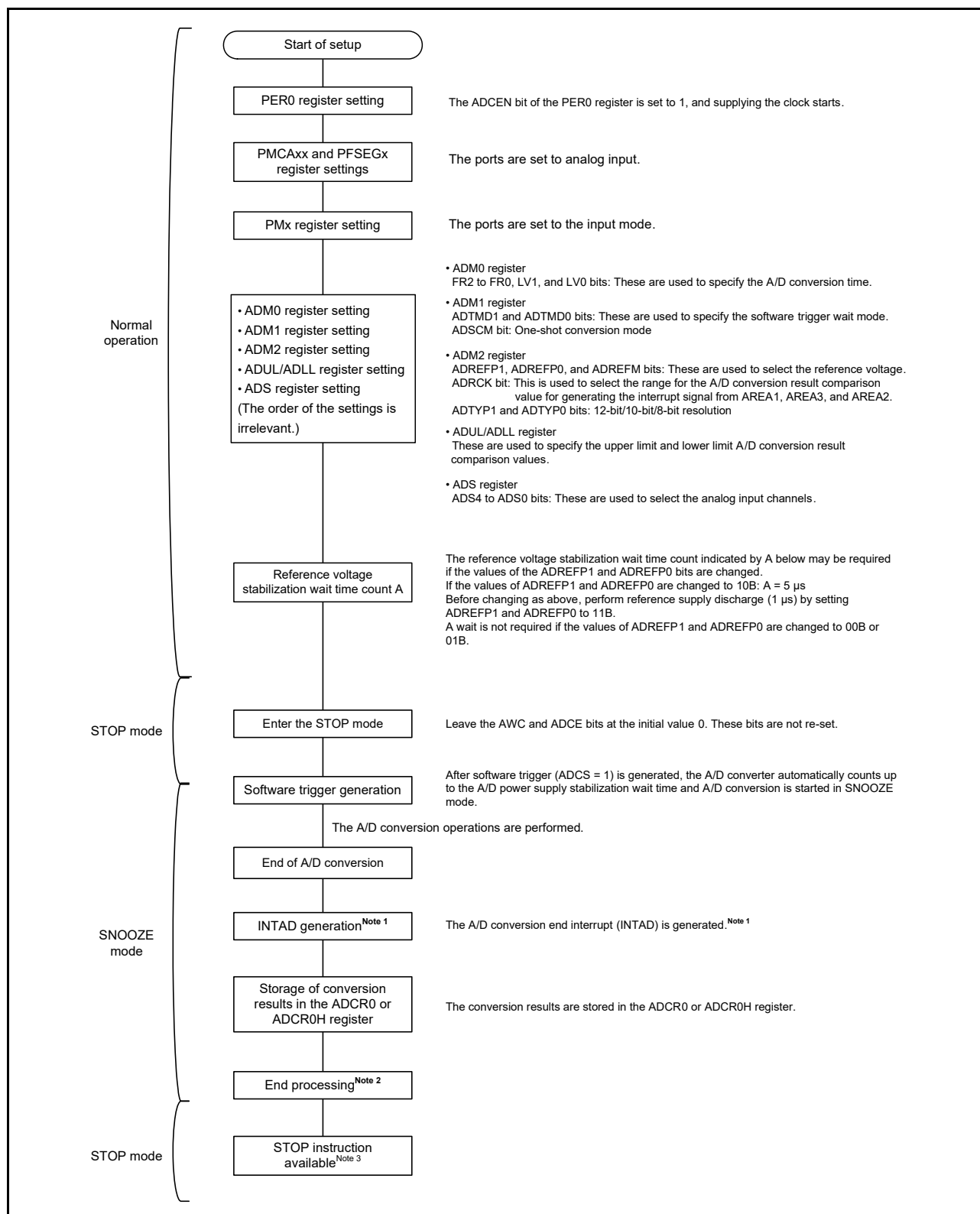
- One-shot conversion mode

When the STOP instruction is executed after the A/D conversion is completed and the interrupt request signal (INTAD) is generated, the supplied clock can be stopped.

- Sequential conversion mode

When ADCS is written with 0 after the A/D conversion is completed and the interrupt request signal (INTAD) is generated, the sequential conversion mode is stopped. After that, when the STOP instruction is executed after 2 cycles of the conversion clock (fAD) have elapsed, the supplied clock can be stopped.

Figure 16 - 34 Flowchart for Setting up SNOOZE Mode (Software Trigger)



Note 1. Clear the ADRCK bit and ADUL/ADLL register to the initial value in the initial setting so that an interrupt request signal (INTAD) is generated each time A/D conversion is completed.

Note 2. Sequential conversion mode requires the end processing. Write 0 to ADCS after INTAD is generated. Then, after 2 cycles of the conversion clock (f_{AD}) have elapsed, the STOP instruction can be executed.

Note 3. If a software trigger is input after the STOP instruction, A/D conversion operation is again performed in the SNOOZE mode.

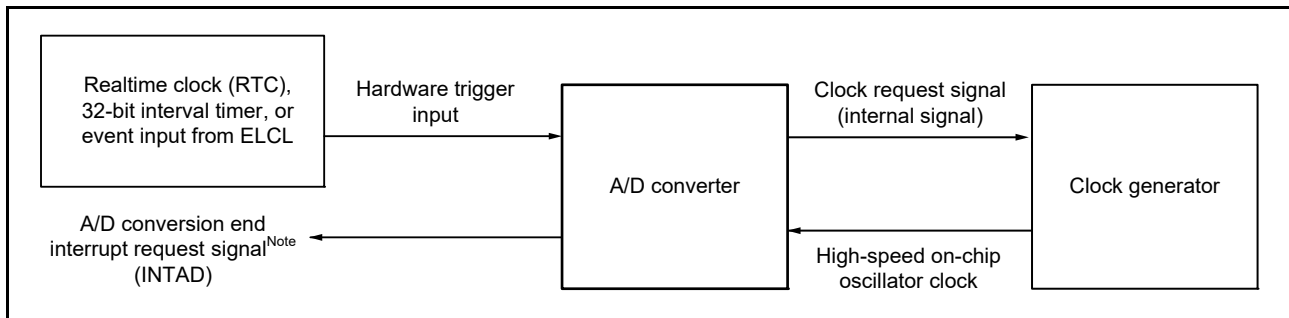
16.8.2 A/D conversion by inputting a hardware trigger

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger. When performing A/D conversion by inputting a hardware trigger in SNOOZE mode, only the hardware trigger wait mode (one-shot conversion mode) can be used.

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 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.

Figure 16 - 35 Block Diagram When Using SNOOZE Mode in Hardware Trigger Wait Mode



When using the SNOOZE mode, the initial setting of each register is specified before switching to the STOP mode (for details about these settings, see **Figure 16 - 36 Flowchart for Setting up SNOOZE Mode (Hardware Trigger)**). Just before moving to STOP mode, set bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 1. After the initial settings are specified, set bit 0 (ADCE) of A/D converter mode register 0 (ADM0) 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/ADLL register), there is a possibility of no interrupt signal being generated.

When the 32-bit interval timer interrupt signal (INTITL) is selected as a hardware trigger, the detection flag in the ITLS0 register has to be cleared each time a 32-bit interval timer interrupt signal (INTITL) is generated. Therefore, clear the setting of the A/D conversion result comparison function (ADRCK bit and ADUL/ADLL register) to the initial value so that an A/D conversion end interrupt request signal (INTAD) is generated upon completion of A/D conversion.

Caution Select the hardware trigger signal from among the realtime clock interrupt signal (INTRTC), 32-bit interval timer interrupt signal (INTITL), and event input from the ELCL.

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/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

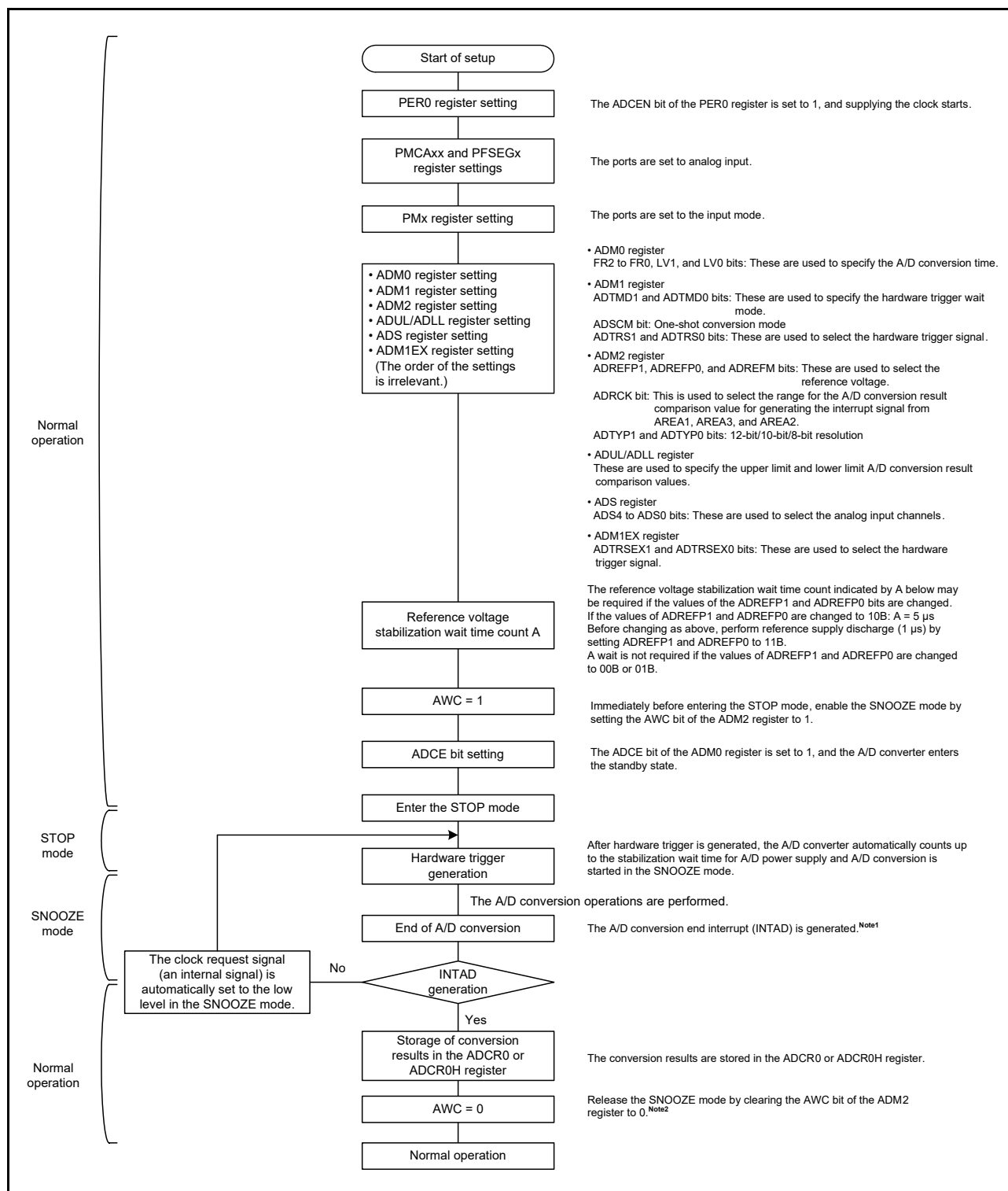
When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

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/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

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.

Figure 16 - 36 Flowchart for Setting up SNOOZE Mode (Hardware Trigger)



Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated depending on the settings of the ADRCCK bit and ADUL and ADLL registers, the result is not stored in the ADCR0 or ADCR0H register. The A/D converter enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

16.9 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 1 LSB (Least Significant Bit). The percentage of 1 LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1 LSB is as follows when the resolution is 12 bits.

$$1 \text{ LSB} = 1/2^{12} = 1/4096 \\ \approx 0.024 \% \text{FSR}$$

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, differential linearity errors, and 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 $\pm 1/2$ LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of $\pm 1/2$ 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 16 - 37 Overall Error

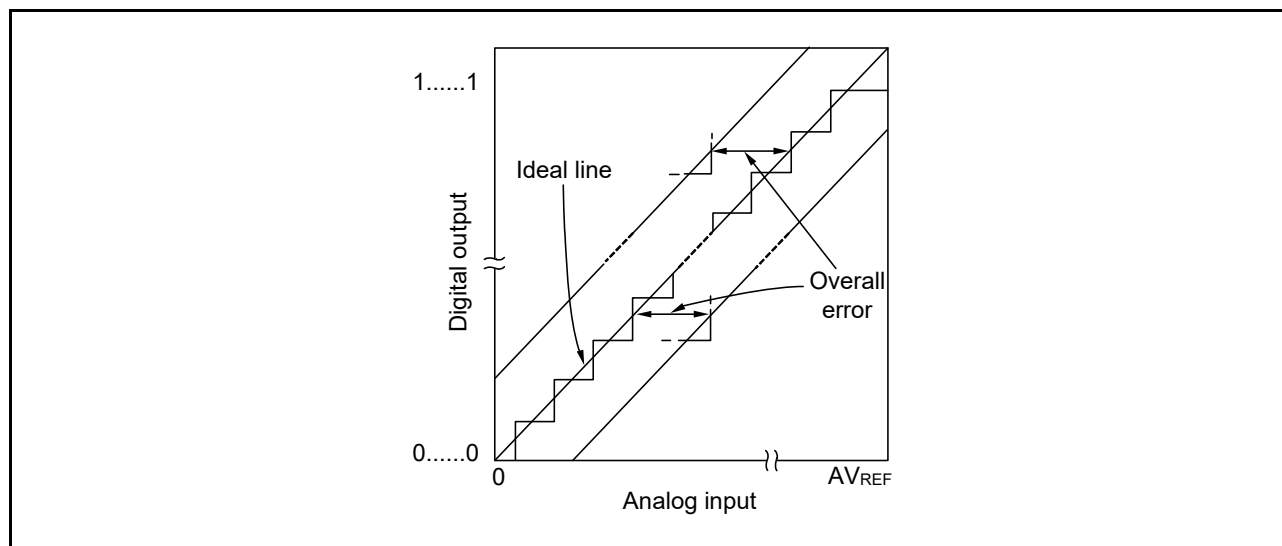
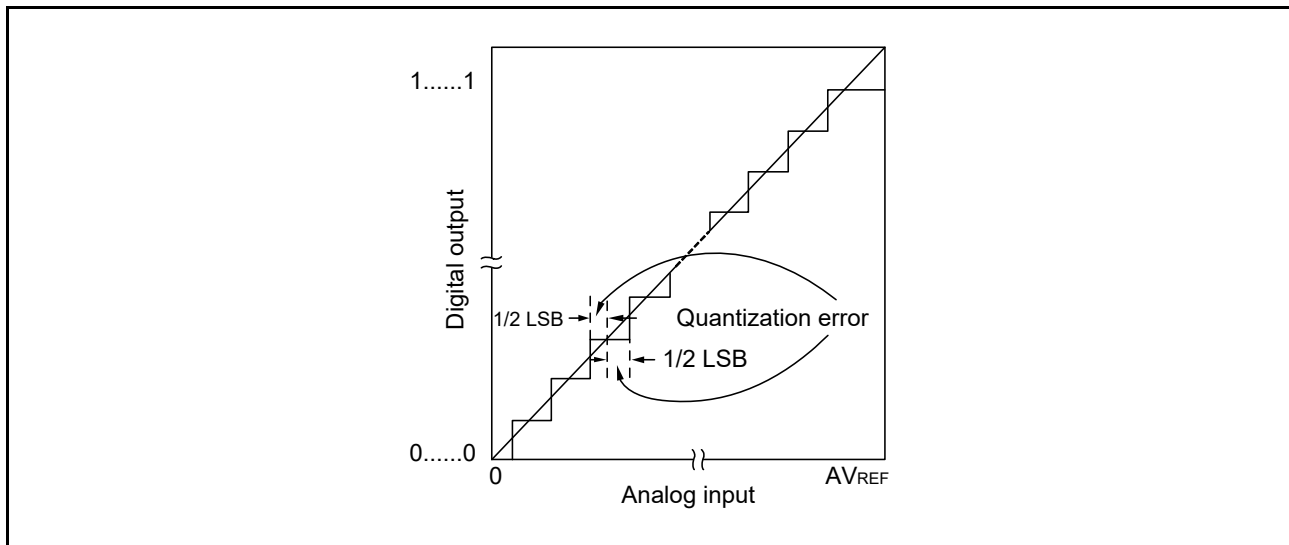


Figure 16 - 38 Quantization Error



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.....000 to 0.....001. 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.....001 to 0.....010.

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.....110 to 1.....111.

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 1 LSB, this indicates the difference between the actual measurement value and the ideal value of the width of output code.

Figure 16 - 39 Zero-scale Error

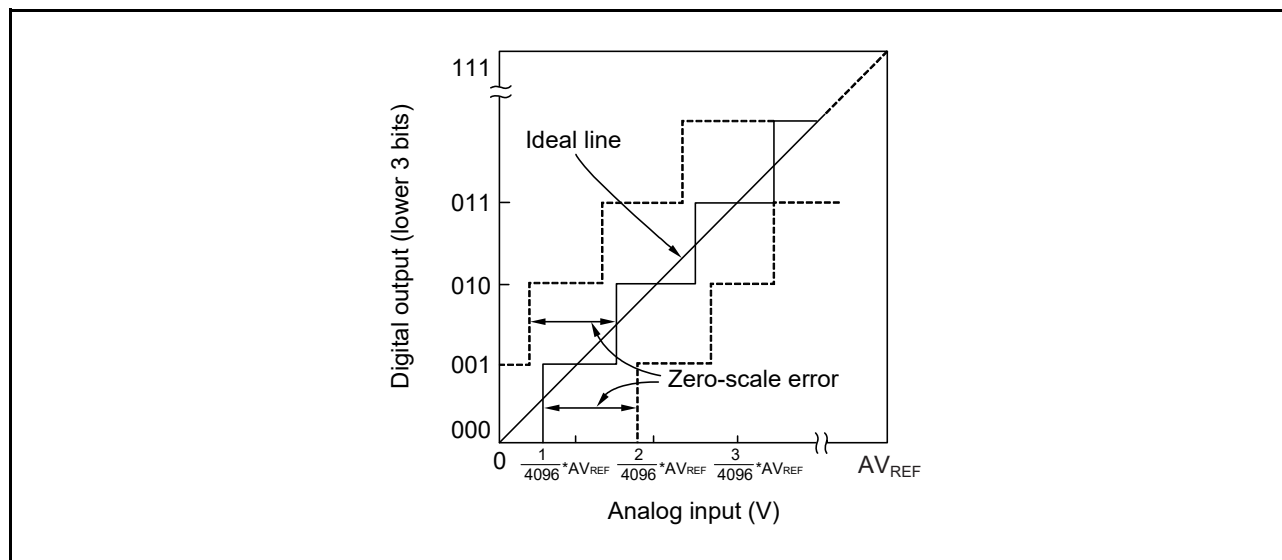


Figure 16 - 40 Full-scale Error

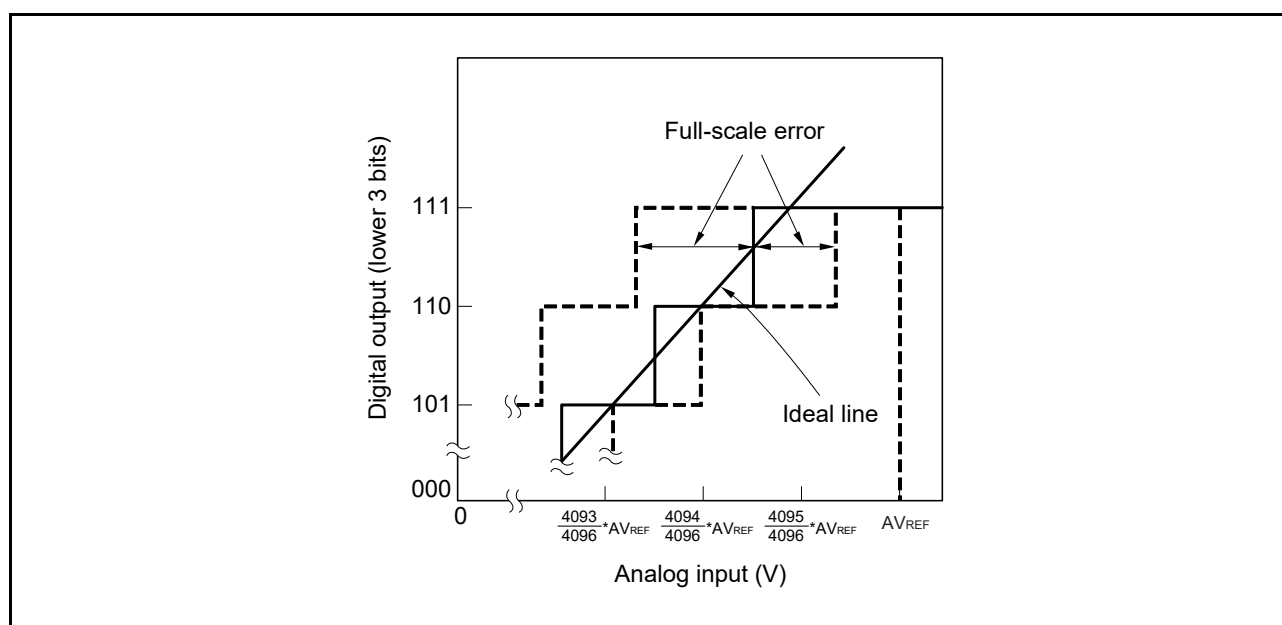


Figure 16 - 41 Integral Linearity Error

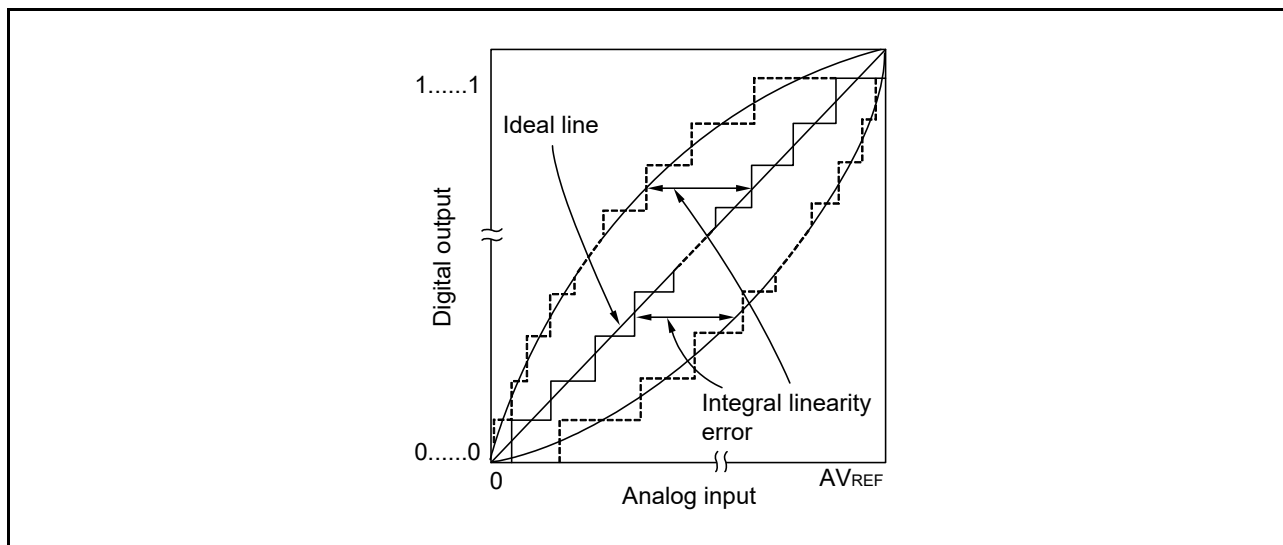
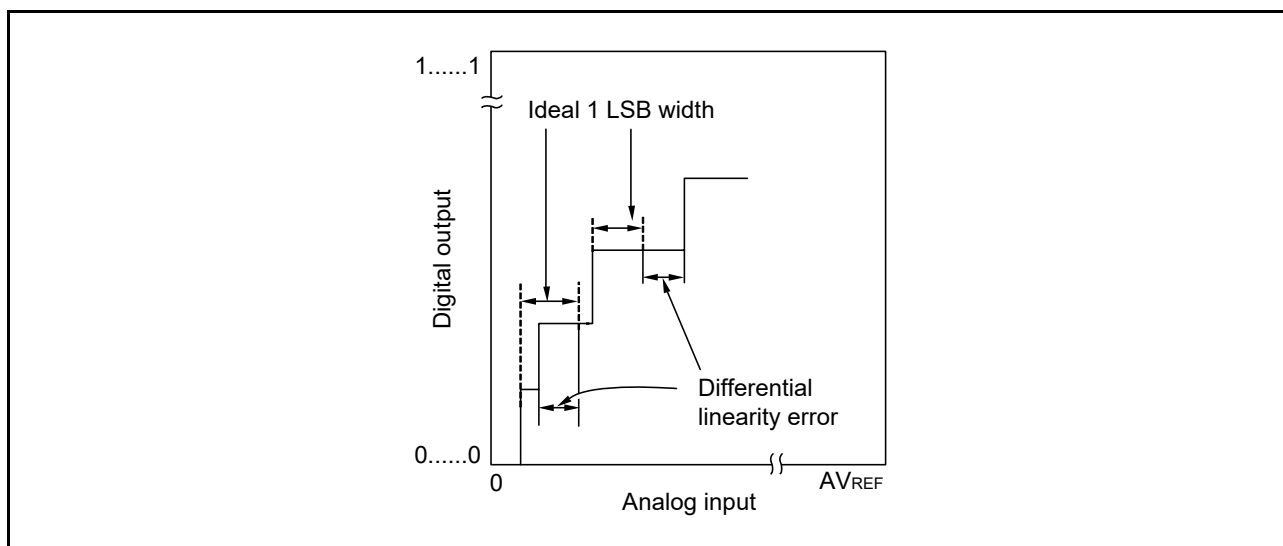


Figure 16 - 42 Differential Linearity Error

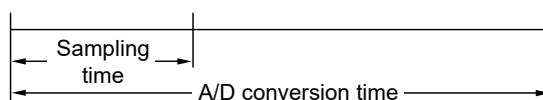


8. Conversion time

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.



16.10 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, ANI1, and ANI16 to ANI26 pins

Observe the rated range of the ANI0, ANI1, and ANI16 to ANI26 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 reference voltage for the + side 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 42 Electrical Characteristics.

3. Conflicting operations

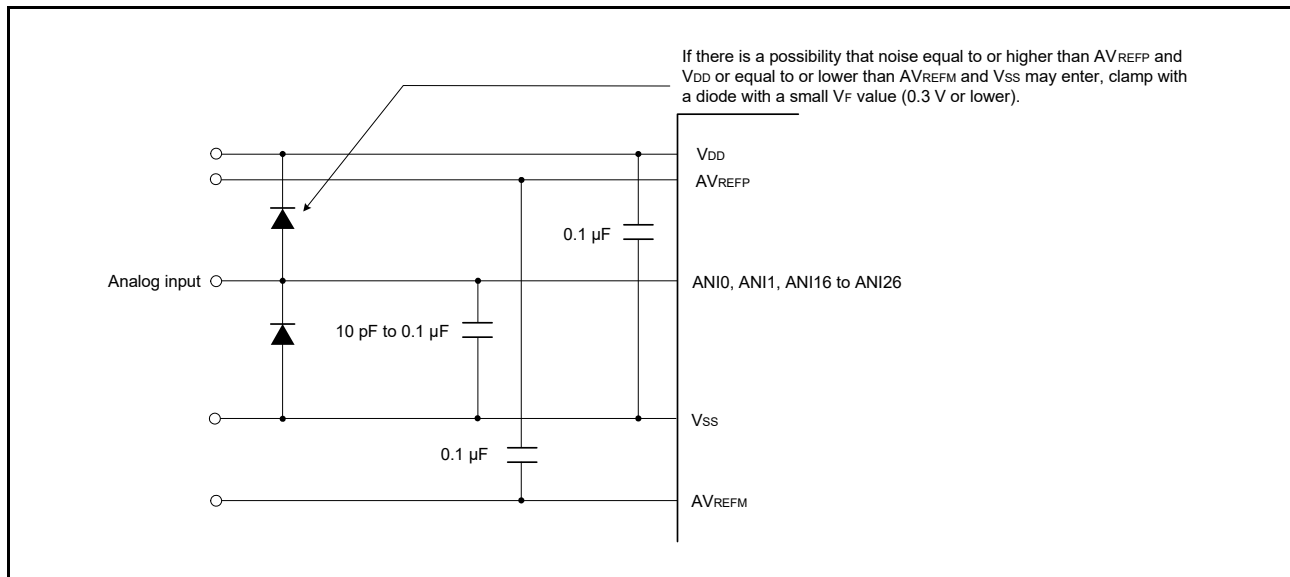
- <1> Conflict between the conversion result being stored in the A/D conversion result register (ADCR0 or ADCR0H) at the end of conversion and the read access to the ADCR0 or ADCR0H register by instruction. The ADCR0 or ADCR0H register read has priority. After the read operation, the new conversion result is written to the ADCR0 or ADCR0H registers.
- <2> Conflict between the conversion result being stored in the A/D conversion result register (ADCR0 or ADCR0H) at the end of conversion and the write access to the A/D converter mode register 0 (ADM0) or analog input channel specification register (ADS) by instruction. The ADM0 and ADS registers write has priority. The ADCR0 and ADCR0H registers write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

4. Noise countermeasures

To maintain the 12-bit/10-bit resolution, attention must be paid to noise input to the AV_{REFP} , V_{DD} , ANI0, ANI1, and ANI16 to ANI26 pins.

- <1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.1 μF) 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 16 - 43** 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 16 - 43 Connections of VDD, AVREFP, and Analog Input Pins



5. Analog input (ANlxx) pins

<1> The analog input pins (ANI0, ANI1, and ANI16 to ANI26) are also used as input port pins (P10 to P14 and P20 to P27). When A/D conversion is performed with any of the ANI0, ANI1, and ANI16 to ANI26 pins selected, do not change the output values of P10 to P14 and P20 to P27 while conversion is in progress; otherwise the conversion resolution may be degraded.

<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 avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.

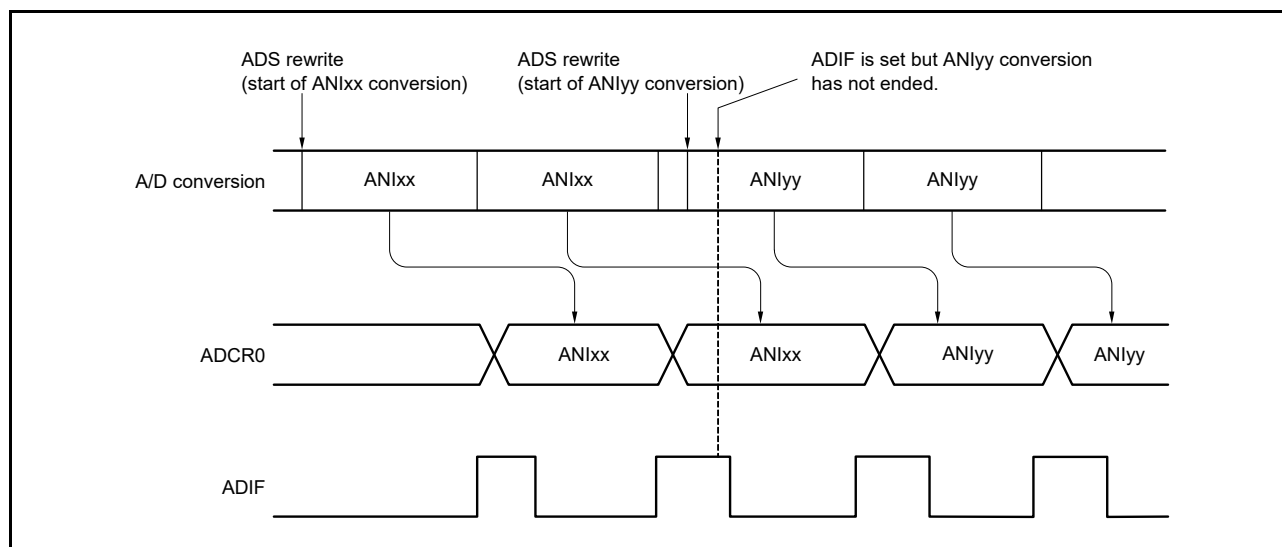
6. Input impedance of analog input (ANlxx) pins

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. 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 k Ω . If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μ F) to the pin from among ANI0, ANI1, and ANI16 to ANI26 to which the source is connected (see **Figure 16 - 43**). 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 sampling.

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 have been set before the ADS register is rewritten. When reading the ADIF flag immediately after rewriting to the ADS register, note that the ADIF flag is set although 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 16 - 44 Timing of A/D Conversion End Interrupt Request Generation



8. Conversion results just after A/D conversion start

While in the software trigger no-wait 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 $1 \mu\text{s} + 2$ cycles of the conversion clock (f_{AD}) 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 (ADCR0, ADCR0H) 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 registers (PMCAxx), or LCD port function registers (PFSEGx), the contents of the ADCR0 and ADCR0H registers may become undefined. After the completion of conversion, read the conversion result before writing to the ADM0, ADS, PMCAxx, or PFSEGx register; otherwise, an incorrect conversion result may be read.

10. Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 16 - 45 Internal Equivalent Circuit of ANIxx Pin

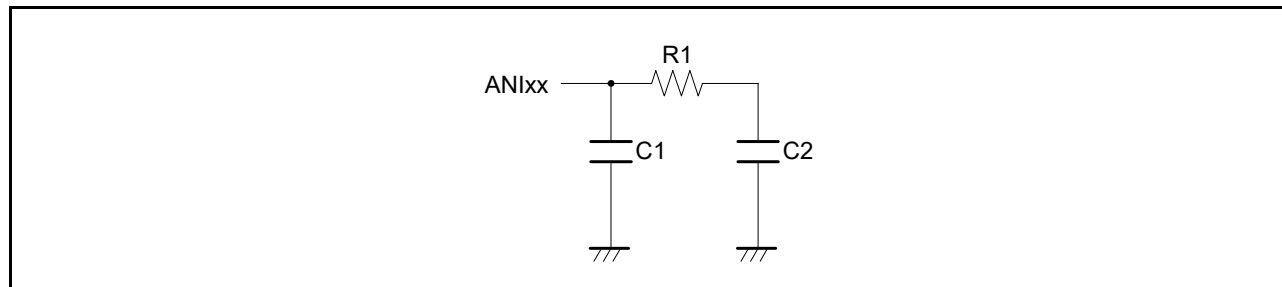


Table 16 - 4 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AVREFF, VDD	ANIxx pin	R1 [kΩ]	C1 [pF]	C2 [pF]
2.4 V ≤ VDD ≤ 5.5 V	ANI0, ANI1	11	8	9
	ANI16 to ANI26	12	8	10
1.8 V ≤ VDD < 2.4 V	ANI0, ANI1	55	8	9
	ANI16 to ANI26	60	8	10
1.6 V ≤ VDD < 1.8 V	ANI0, ANI1	110	8	9
	ANI16 to ANI26	120	8	10

Remark The resistance and capacitance values are not guaranteed values.

11. Starting the A/D converter

Start the A/D converter after the AVREFF and VDD voltages stabilize.

Section 17 D/A Converter (DAC)

17.1 Functions of D/A Converter

The D/A converter is an 8-bit resolution converter that converts digital inputs into analog signals. It is used to control analog outputs for two independent channels (ANO_j), and the reference voltage output to comparator 1. The D/A converter has the following features.

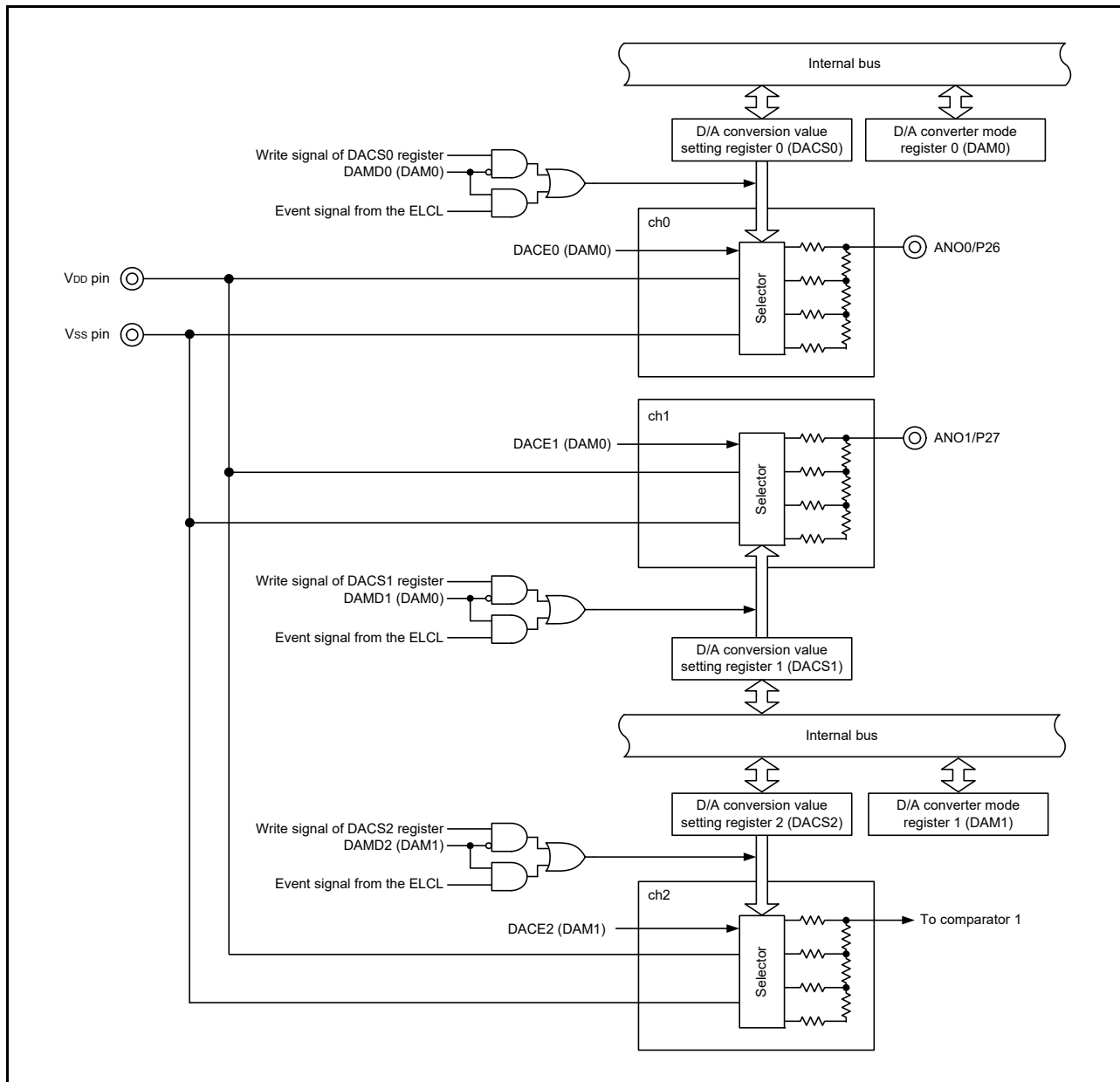
- 8-bit resolution × 3 channels
- R-2R ladder method
- Output analog voltage
 - 8-bit resolution: $V_{DD} \times m8/256$ (m8: Value set to DACSi register)
- Operation mode
 - Normal mode
 - Realtime output mode

Remark $i = 0 \text{ to } 2; j = 0, 1$

17.2 Configuration of D/A Converter

Figure 17 - 1 shows the block diagram of D/A converter.

Figure 17 - 1 Block Diagram of D/A Converter



17.3 Registers for Controlling the D/A Converter

The following registers are used to control the D/A converter.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- D/A converter mode register 0 (DAM0)
- D/A converter mode register 1 (DAM1)
- D/A conversion value setting register i (DACSi) (i = 0 to 2)
- Output signal select registers n (ELOSELn) (n = 0 to 7)
- Output signal enable register (ELOENCTL)
- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- LCD port function registers (PFSEGx)

17.3.1 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 D/A converter is to be used, be sure to set bit 7 (DACEN) 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 17 - 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>
<R> PER1	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUEN

DACEN	Control of supply of an input clock to the D/A converter
0	Stops supply of an input clock. - The SFRs used by the D/A converter cannot be written. - When an SFR used by the D/A converter is read, the value returned is 00H or 0000H.
1	Enables supply of an input clock. The SFRs used by the D/A converter can be read and written.

Caution 1. Be sure to set the DACEN bit to 1 before making the D/A converter settings. If DACEN = 0, writing to the registers for controlling the D/A converter is ignored, and the value read is 00H. Note, however, that writing to the port mode registers (PMxx), port mode control A registers (PMCAxx), and LCD port function registers (PFSEGx) is valid.

Caution 2. Be sure to set bit 1 to 0.

17.3.2 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 7 (DACRES) to 1 to reset the D/A converter. 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 17 - 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	DACRES	SMSRES	CMPRES	TML32RES	0	0	0	CTSURES
	DACRES	Control resetting of the D/A converter						
	0	The D/A converter is released from the reset state.						
	1	The D/A converter is in the reset state. • The SFRs for use with the D/A converter are initialized.						

Caution Be sure to set bits 3 to 1 to 0.

17.3.3 D/A converter mode register 0 (DAM0)

This register controls the operation of the D/A converter channels 1 and 0. The DAM0 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 17 - 4 Format of D/A Converter Mode Register 0 (DAM0)

Address: F0332H

After reset: 00H

R/W: R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM0	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACE1	Operation control of D/A converter channel 1
0	Stops D/A conversion
1	Enables D/A conversion

DACE0	Operation control of D/A converter channel 0
0	Stops D/A conversion
1	Enables D/A conversion

DAMD1	Operation mode selection for D/A converter channel 1
0	Normal mode
1	Realtime output mode

DAMD0	Operation mode selection for D/A converter channel 0
0	Normal mode
1	Realtime output mode

Caution Be sure to set bits 7, 6, 3, and 2 to 0.

17.3.4 D/A converter mode register 1 (DAM1)

This register controls the operation of the D/A converter channel 2. The DAM1 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 17 - 5 Format of D/A Converter Mode Register 1 (DAM1)

Address: F033AH
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	3	2	1	0
DAM1	0	0	0	DACE2	0	0	0	DAMD2

DACE2	Operation control of D/A converter channel 2
0	Stops D/A conversion
1	Enables D/A conversion

DAMD2	Operation mode selection for D/A converter channel 2
0	Normal mode
1	Setting prohibited

Caution Be sure to set bits 7 to 5, and 3 to 1 to 0.

17.3.5 D/A conversion value setting register i (DACS_i) (i = 0 to 2)

These registers are used to set the analog voltage values to be output to the ANO_j pins and to comparator 1 as a reference voltage when the D/A converter is in use. The DACS_i register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Remark j = 0, 1

Figure 17 - 6 Format of D/A Conversion Value Setting Register i (DACS_i) (i = 0 to 2)

Address: F0330H (DACS0), F0331H (DACS1), F0338H (DACS2)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DACS _i	DACS _i 7	DACS _i 6	DACS _i 5	DACS _i 4	DACS _i 3	DACS _i 2	DACS _i 1	DACS _i 0

Remark The relation between the resolution and analog output voltage (VANO_i) of the D/A converter are as follows.

$$VANO_i = V_{DD} \times (DACS_i) / 256$$

When the D/A converter is not used, set the DACE_i bit to 0 (output disable) and set the DACS_i register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce unnecessary current consumption.

17.3.6 Registers for controlling the event output from the logic and event link controller

D/A conversion in the realtime output mode is triggered by an event signal from the logic and event link controller. For details, see **24.3.23 Output signal select registers n (ELOSELn) (n = 0 to 7)** and **24.3.24 Output signal enable register (ELOENCTL)**.

Caution The realtime output mode is not available for channel 2 of the D/A converter.

17.3.7 Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter

Set the following registers for controlling the port functions multiplexed with the analog outputs of the D/A converter.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

17.4 Operations of D/A Converter

17.4.1 Operation in normal mode

D/A conversion is triggered by writing to the DACSi register. An example of the settings is given below.

- Output of the analog voltage to an ANOj pin
 - <1> Set the DACEN bit of the peripheral enable register 1 (PER1) to start the supply of the input clock to the D/A converter.
 - <2> Set the corresponding port bits of the port mode control A register (PMCA) for operation as analog pins (1). Also, set the corresponding port bits of the LCD port function register (PFSEG) to 1.
 - <3> Set the DAMD0 bit of the D/A converter mode register 0 (DAM0) to 0 (normal mode).
 - <4> Set the analog voltage value to be output to the ANOj pin to the D/A conversion value setting register j (DACSj).

Steps <1> to <4> complete the initial settings.

- <5> Set the DACEj bit of the DAM0 register to 1 to start D/A conversion. Once the settling time has elapsed, the analog voltage set in step <4> is output to the ANOj pin.
- <6> To proceed with subsequent D/A conversions, write to the DACSj register.

- Output of the analog voltage to comparator 1
 - <1> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converter.
 - <2> Set the DAMD2 bit of the D/A converter mode register 1 (DAM1) to 0 (normal mode).
 - <3> Set the analog voltage value to be output to comparator 1 to the D/A conversion value setting register 2 (DACS2).

Steps <1> to <3> complete the initial settings.

- <4> Set the DACE2 bit of the DAM1 register to 1 to start D/A conversion. Once the settling time has elapsed, the analog voltage set in step <3> is output to comparator 1.
- <5> To proceed with subsequent D/A conversions, write to the DACS2 register.

The previous result of D/A conversion is retained until the next D/A conversion proceeds. Setting of the DACEj and DACE2 bits of the DAM0 and DAM1 registers, respectively, to 0 stops D/A conversion.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOj pin when a settling time has elapsed after 1 is set for the last time.

Caution 2. If the DACSi register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.

Caution 3. Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converter.

Remark i = 0 to 2; j = 0, 1

17.4.2 Operation in realtime output mode

D/A conversion on each channel is triggered by the event signals from the ELCL. An example of the settings is given below.

- <1> Set the DACRES bit of the peripheral reset control register 1 (PRR1) to 0 to release the D/A converter from the reset state.
- <2> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converter.
- <3> Set the corresponding port bits of the port mode control A register (PMCA) to analog pins (1). Also, set the corresponding port bits of the LCD port function register (PFSEG) to 1.
- <4> Set the DAMDj bit of the D/A converter mode register 0 (DAM0) to 0 (normal mode).
- <5> Set the analog voltage value to be output to the ANOj pin to the D/A conversion value setting register j (DACSj).
- <6> Set the DACEj bit of the DAM0 register to 1 to start D/A conversion. Once the settling time has elapsed, the analog voltage set in step <5> is output to the ANOj pin.
- <7> Use the registers (ELOSEL6, ELOSEL7, and ELOENCTL) that control the event signal output from the logic and event link controller to set the trigger signal for use in realtime output mode.
- <8> Set the DAMDj bit of the DAM0 register to 1 (realtime output mode).
- <9> Start the operation of the event source.

Steps <1> to <9> completes initial settings.

- <10> Upon generation of the trigger signals used for realtime output mode, D/A conversion starts and the analog voltage set in step <5> will be output to the ANOj pin once the settling time has elapsed.

Set the analog voltage value to be output to the ANOj pin, to the DACSj register before starting the next D/A conversion (trigger signal used for realtime output mode is generated). Setting of the DACEi bit of the DAM register to 0 stops D/A conversion.

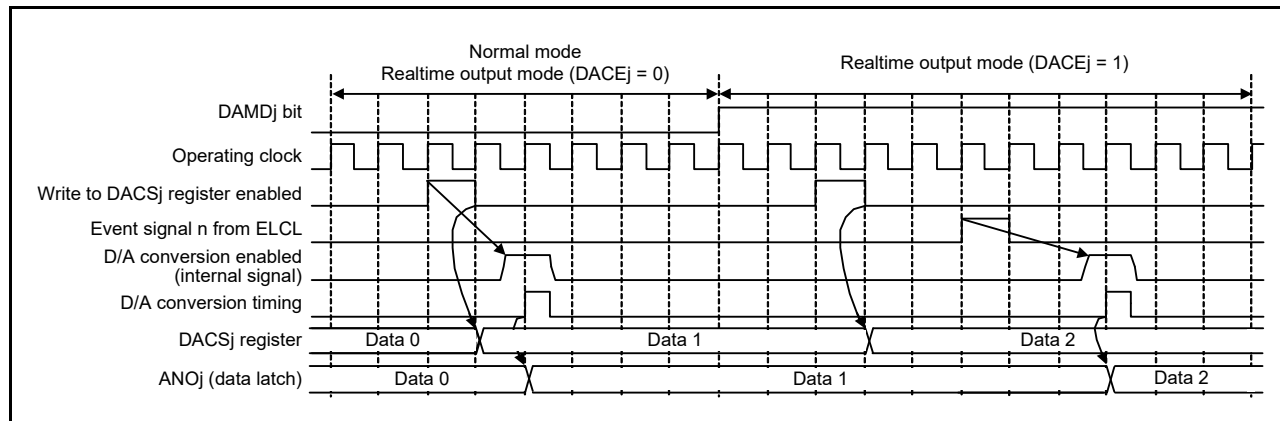
- Caution 1.** Even if 1, 0, and then 1 is set to the DACEj bit, the analog voltage set by the DACSj register is output to the ANOj pin when a settling time has elapsed after 1 is set for the last time.
- Caution 2.** Set the interval between each generation of the trigger signal used for realtime output mode of the same channel to longer than the settling time. If a trigger signal used for realtime output mode is generated during the settling time, D/A conversion is aborted and reconversion starts.
- Caution 3.** Set the interval between each generation of the trigger signal used for realtime output mode of the same channel to longer than the three clock cycles of fCLK. When a trigger signal is generated consecutively at intervals of three or fewer fCLK clock cycles, only the first signal triggers D/A conversion.
- Caution 4.** Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converter.
- Caution 5.** The realtime output mode is not available in channel 2 of the D/A converter.

Remark j = 0, 1

17.4.3 Timing for outputting D/A conversion value

Figure 17 - 7 shows the timing for outputting D/A conversion value.

Figure 17 - 7 Timing for Outputting D/A Conversion Value



Remark j = 0, 1; n = 6, 7

- Normal mode and realtime output mode when conversion is disabled
The value is written to the data latch (output from the ANOj pin) after one cycle of the operating clock when the DACSj register is written.
- Realtime output mode when conversion is enabled
The value is written to the data latch (output from the ANOj pin) after three cycles of the operating clock when the event signal from the ELCL is accepted.

17.5 Points for Caution when the D/A Converter is to be Used

Observe the following cautions when using the D/A converter.

1. The digital port I/O functions which are multiplexed with the functions of the ANOj pins do not operate if the port settings in the port mode control A register (PMCA) are for analog pins. When the P2 register is read while the port settings in the PMCA register are for analog pins, 0 is read in the input mode and the values set in the P2 register are read in the output mode. If the digital output mode is set, data are not output to the pins.
2. The operation of the D/A converter continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACEi bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converter.
3. To stop the realtime output mode (including when changing to normal mode), one of the following procedures must be used:
 - Wait for at least three cycles of the operating clock after having stopped the trigger output source, and then set bits DACEj and DAMDj to 0.
 - Set bits DACEj and DAMDj to 0, then set the DACEN bit of the PER1 register to 0 (stopping supply of the input clock to the DAC). Note that setting the DACEN bit to 0 does not initialize the D/A converter. Use bit 7 (DACRES) of PRR1 to initialize the D/A converter and its SFRs.
4. Do not start A/D conversion by using analog input pin functions that are multiplexed with the ANOj pins while D/A conversion is enabled.
5. In realtime output mode, set the value of the DACSj register before a trigger signal used for realtime output mode is generated. Do not change the set value of the DACSj register while the trigger signal is output.
6. Since the output impedance of the D/A converter is high, no current can be taken out from the ANOj pin. If the input impedance of the load is low, insert a follower amplifier between the load and the ANOj pin before use. In addition, the wiring length between the follower amplifier and the load must be as short as possible due to the high output impedance. If the wiring length is long, take measures such as placing a ground pattern around the wiring area.
7. Before entering STOP mode while realtime output mode is enabled, disable any linking of ELCL events.

Remark i = 0 to 2; j = 0, 1

Section 18 Comparator (CMP)

The number of comparators depends on the products.

	44-Pin	48-Pin	52-Pin	64-Pin	80-Pin	100-Pin
Number of Comparators	1		2			

18.1 Functions of Comparator

The comparator has the following functions.

- Comparator high-speed mode, or comparator low-speed mode can be selected.
- The external reference voltage input and the internal reference voltage^{Note 1} or D/A converter output ^{Note 2} are selectable as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event signal can be output to the logic and event link controller (ELCL) by detecting an active edge of the comparator output.

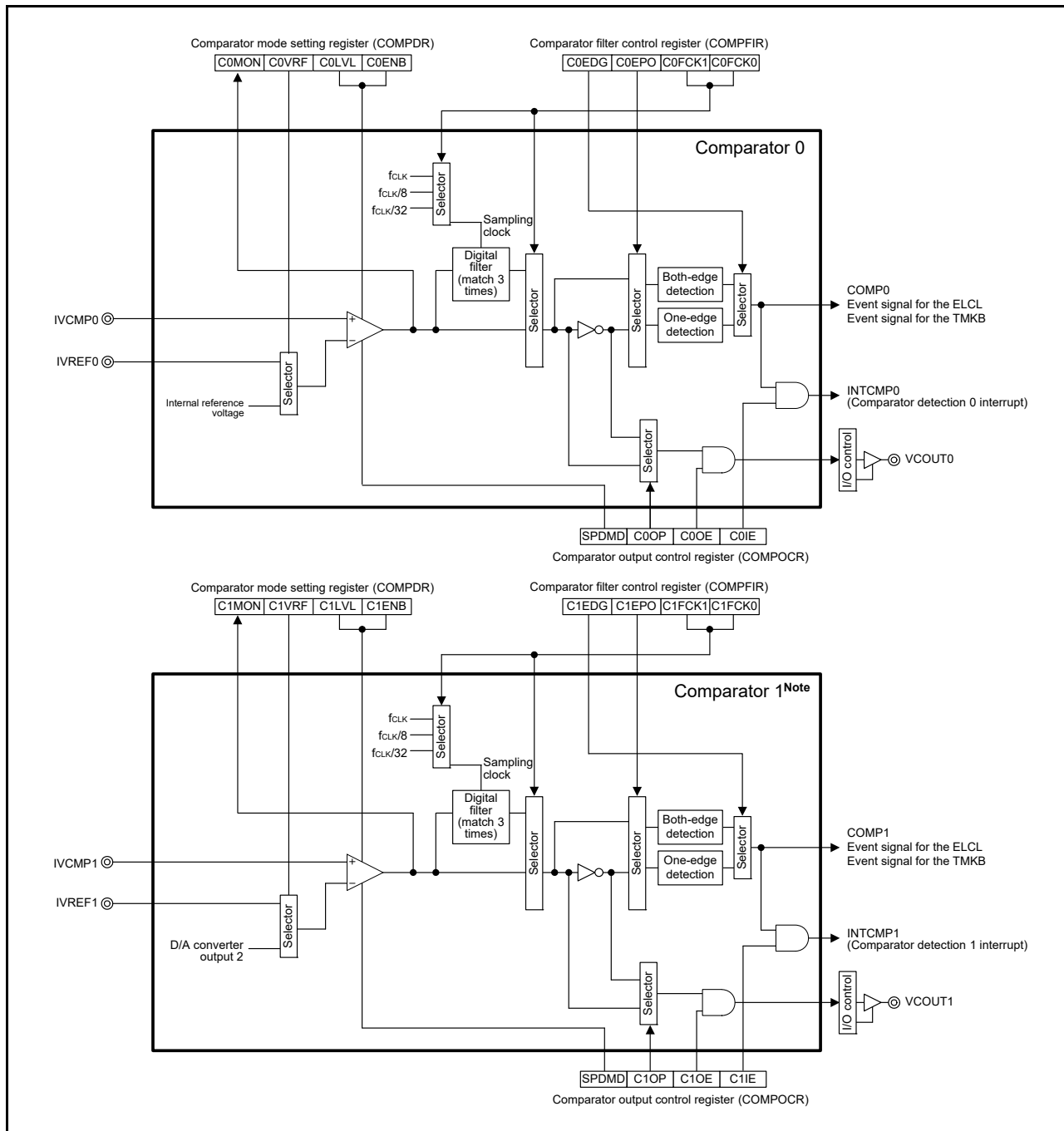
Note 1. For the internal reference voltage, see **Section 42 Electrical Characteristics**.

Note 2. The internal reference voltage and D/A converter output 2 are selectable for comparators 0 and 1, respectively.

18.2 Configuration of Comparator

Figure 18 - 1 shows the comparator block diagram.

Figure 18 - 1 Comparator Block Diagram



Note Comparator 1 is only present in the 52- to 100-pin products.

18.3 Registers for Controlling the Comparator

The following registers are used to control the comparator.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Comparator mode setting register (COMPMDR)
- Comparator filter control register (COMPFIR)
- Comparator output control register (COMPOCR)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- LCD port function registers (PFSEGx)

18.3.1 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 comparator is to be used, be sure to set bit 5 (CMPEN) 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 18 - 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>
<R> PER1	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUN
	CMPEN	Control of supply of an input clock to the comparator						
	0	Stops supply of an input clock. • The SFRs used by the comparator cannot be written. • When an SFR used by the comparator is read, the value returned is 00H or 0000H.						
	1	Enables supply of an input clock. • The SFRs used by the comparator can be read and written.						

Caution 1. Be sure to set the CMPEN bit to 1 before making the comparator settings. If CMPEN = 0, writing to the registers which control the comparator is ignored. Note, however, that writing to the following registers is valid.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- LCD port function registers (PFSEGx)

Caution 2. Be sure to set bit 1 to 0.

18.3.2 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 5 (CMPRES) to 1 to reset the comparator. 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 18 - 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	DACRES	SMSRES	CMPRES	TML32RES	0	0	0	CTSURES
	Control resetting of the comparator							
	0	The comparator is released from the reset state.						
	1	The comparator is in the reset state. The SFRs for use with the comparator are initialized.						

Caution Be sure to set bits 3 to 0 to 0.

18.3.3 Comparator mode setting register (COMPMDR)

The COMPMDR register is used to make the various settings of comparators 0 and 1, such as selection of the reference voltage, starting or stopping comparing operation, and also indicates the results of comparison. The COMPMDR register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the CnMON bit is read-only. The value of this register following a reset is 00H.

Figure 18 - 4 Format of Comparator Mode Setting Register (COMPMDR)

Address: F0340H

After reset: 00H

R/W: R/W

Symbol	7	6	5	<4>	3	2	1	<0>
COMPMDR	C1MON	C1VRF	C1LVL	C1ENB	C0MON	C0VRF	C0LVL	C0ENB
	C1MON	Comparator 1 monitor flag ^{Notes 1, 2}						
	0	IVCMP1 < comparator 1 reference voltage (IVREF1 or D/A converter output 2)						
	1	IVCMP1 > comparator 1 reference voltage (IVREF1 or D/A converter output 2)						
	C1VRF	Selection of comparator 1 reference voltage						
	0	Supply through the IVREF1 pin						
	1	Supply through D/A converter output 2						
	C1LVL	Selection of comparator 1 reference voltage range						
	0	0 to VDD - 1.4 V						
	1	1.4 V to VDD						
	C1ENB	Comparator 1 operation enable						
	0	Comparator 1 operation disabled						
	1	Comparator 1 operation enabled						
	C0MON	Comparator 0 monitor flag ^{Notes 1, 2}						
	0	IVCMP0 < comparator 0 reference voltage (IVREF0 or internal reference voltage)						
	1	IVCMP0 > comparator 0 reference voltage (IVREF0 or internal reference voltage)						
	C0VRF	Selection of comparator 0 reference voltage ^{Note 3}						
	0	Supply through the IVREF0 pin						
	1	Supply through the internal reference voltage						
	C0LVL	Selection of comparator 0 reference voltage range ^{Note 3}						
	0	0 to VDD - 1.4 V						
	1	1.4 V to VDD						
	C0ENB	Comparator 0 operation enable						
	0	Comparator 0 operation disabled						
	1	Comparator 0 operation enabled						

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

Note 1. The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.

Note 2. Bits 7 and 3 are read-only. The value written to these bits is ignored.

Note 3. When the internal reference voltage (C0VRF = 1) is to be used as the comparator 0 reference voltage, C0LVL = 0 and $V_{DD} \geq (\text{internal reference voltage (max.)} + 1.4 \text{ V})$ must be satisfied.

Caution **Be sure to set bits 7 to 4 to 0 in the 44- and 48-products.**

18.3.4 Comparator filter control register (COMPFIR)

The COMPFIR register is used to select the valid edge for use with the comparator interrupt signals and whether or not to use the digital filter. If rejecting noise is required, set the CnFCK1 and CnFCK0 bits to enable the digital filters. While a digital filter is enabled, the filter conveys the input level after detecting matches for three cycles of the sampling clock for use with the filter. The COMPFIR 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 18 - 5 Format of Comparator Filter Control Register (COMPFIR)

Address: F0341H
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0
C1EDG	Comparator 1 edge detection selection ^{Note 1}							
0	Interrupt request by comparator 1 one-edge detection							
1	Interrupt request by comparator 1 both-edge detection							
C1EPO	Comparator 1 edge polarity switching ^{Note 1}							
0	Interrupt request at comparator 1 rising edge							
1	Interrupt request at comparator 1 falling edge							
C1FCK1	C1FCK0	Comparator 1 filter selection ^{Notes 1, 3}						
0	0	No comparator 1 filter						
0	1	Comparator 1 filter enabled, sampling at fCLK						
1	0	Comparator 1 filter enabled, sampling at fCLK/8						
1	1	Comparator 1 filter enabled, sampling at fCLK/32						
C0EDG	Comparator 0 edge detection selection ^{Note 2}							
0	Interrupt request by comparator 0 one-edge detection							
1	Interrupt request by comparator 0 both-edge detection							
C0EPO	Comparator 0 edge polarity switching ^{Note 2}							
0	Interrupt request at comparator 0 rising edge							
1	Interrupt request at comparator 0 falling edge							
C0FCK1	C0FCK0	Comparator 0 filter selection ^{Notes 2, 3}						
0	0	No comparator 0 filter						
0	1	Comparator 0 filter enabled, sampling at fCLK						
1	0	Comparator 0 filter enabled, sampling at fCLK/8						
1	1	Comparator 0 filter enabled, sampling at fCLK/32						

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

- Note 1.** Changing the C1FCK0 and C1FCK1 bit field or the C1EPO or C1EDG bit may lead to the generation of an event signal for the ELCL and a comparator 1 interrupt request. Only change these bits while the output from comparator 1 is not selected as an input signal to the ELCL and the comparator 1 interrupt is masked. After that, to use the comparator 1 interrupt, clear the CMPIF1 bit to 0 and then clear the CMPMK1 bit to 0 to enable the interrupt. If the C1FCK1 and C1FCK0 bits are changed from 00B (no comparator 1 filter) to a value other than 00B (comparator 1 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal for the ELCL.
- Note 2.** Changing the C0FCK1 and C0FCK0 bit field or the C0EPO or C0EDG bit may lead to the generation of an event signal for the ELCL and a comparator 0 interrupt request. Only change these bits while the output from comparator 0 is not selected as an input signal to the ELCL and the comparator 0 interrupt is masked. After that, to use the comparator 0 interrupt, clear the CMPIF0 bit to 0 and then clear the CMPMK0 bit to 0 to enable the interrupt. If the C0FCK1 and C0FCK0 bits are changed from 00B (no comparator 0 filter) to a value other than 00B (comparator 0 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal for the ELCL.
- Note 3.** When using the comparators in STOP mode, make the settings for non-use of the filters (C0FCK1 and C0FCK0 = 00B, C1FCK1 and C1FCK0 = 00B).

Caution Be sure to set bits 7 to 4 to 0 in the 44- and 48-products.

18.3.5 Comparator output control register (COMPOCR)

The COMPOCR register is used to select items including the response speed of the comparators, control of the VCOUTn outputs, and enabling or disabling the interrupt request signals. The COMPOCR 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 18 - 6 Format of Comparator Output Control Register (COMPOCR)

Address: F0342H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
COMPOCR	SPDMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE
SPDMD								
Comparator speed selection ^{Note 1}								
0								
Comparator low-speed mode								
1								
Comparator high-speed mode								
C1OP								
VCOUT1 output polarity selection								
0								
Comparator 1 output is output to VCOUT1								
1								
Inverted comparator 1 output is output to VCOUT1								
C1OE								
VCOUT1 pin output enable								
0								
Comparator 1 VCOUT1 pin output disabled								
1								
Comparator 1 VCOUT1 pin output enabled								
C1IE								
Comparator 1 interrupt request enable ^{Note 2}								
0								
Comparator 1 interrupt request disabled								
1								
Comparator 1 interrupt request enabled								
C0OP								
VCOUT0 output polarity selection								
0								
Comparator 0 output is output to VCOUT0								
1								
Inverted comparator 0 output is output to VCOUT0								
C0OE								
VCOUT0 pin output enable								
0								
Comparator 0 VCOUT0 pin output disabled								
1								
Comparator 0 VCOUT0 pin output enabled								
C0IE								
Comparator 0 interrupt request enable ^{Note 3}								
0								
Comparator 0 interrupt request disabled								
1								
Comparator 0 interrupt request enabled								

Note 1. When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0, 1) in the COMPMDR register to 0 in advance.

Note 2. If the C1IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 0 (CMP1F1) in the interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), clear bit 0 (CMP1F1) in the interrupt request flag register 2H (IF2H) to 0 before using an interrupt.

Note 3. If the C0IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 0 (CMP0F0) in the interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), clear bit 0 (CMP0F0) in the interrupt request flag register 2H (IF2H) to 0 before using an interrupt.

Caution Be sure to set bits 6 to 4 to 0 in the 44- and 48-products.

18.3.6 Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator

Set the following registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

18.4 Operation

Comparator 0 and comparator 1 operate independently. Their setting methods and operations are the same. **Table 18 - 1** lists the procedure for setting comparator associated registers.

Table 18 - 1 Procedure for Setting Comparator Associated Registers

Step	Register	Bit	Setting Value
1	PRR1	CMPRES	0 (releasing comparator i from the reset state)
2	PER1	CMPEN	1 (enabling input clock supply)
3	PMCAxx	PMCA42 to PMCA45	Select the function of pins IVCMPi and IVREFi. • Set the PMCA42 to PMCA45 bits to 1 (analog input). • Set the PM42 to PM45 bits to 1 (input mode).
	PMxx	PM42 to PM45	
4	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode, 1: High-speed mode). Note 1
5	COMPMDR	CiVRF	Select the reference voltage. (For i = 0, 0: Input to the IVREF0 pin, 1: Internal reference voltage (For i = 1, 0: Input to the IVREF1 pin, 1: D/A converter output 2)
		CiLVL	Select the reference input voltage range. (0: 0 to VDD - 1.4 V, 1: 1.4 V to VDD)
6	COMPMDR	CiENB	1 (operation enabled)
7	Wait for comparator stabilization time tCMP.		
8	COMPFIR	CiFCK1 and CiFCK0	Select whether the digital filter is used or not and the sampling clock.
9		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).
10	COMPOCR	CiOP, CiOE	Set the VCOUTi output (select the polarity and set output enabled or disabled).
		CiIE	Set the interrupt request output enabled or disabled.
11	Select the port logic output pin: VCOUTi. Refer to 18.4.4 Comparator i output .		
12	PR02H	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.
13	MK2H	CMPMKi	When using an interrupt: Select the interrupt masking.
14	IF2H	CMPIFi	When using an interrupt: 0 (no interrupt requested: initialization) Note 2

Note 1. Comparator 0 and comparator 1 cannot be set independently.

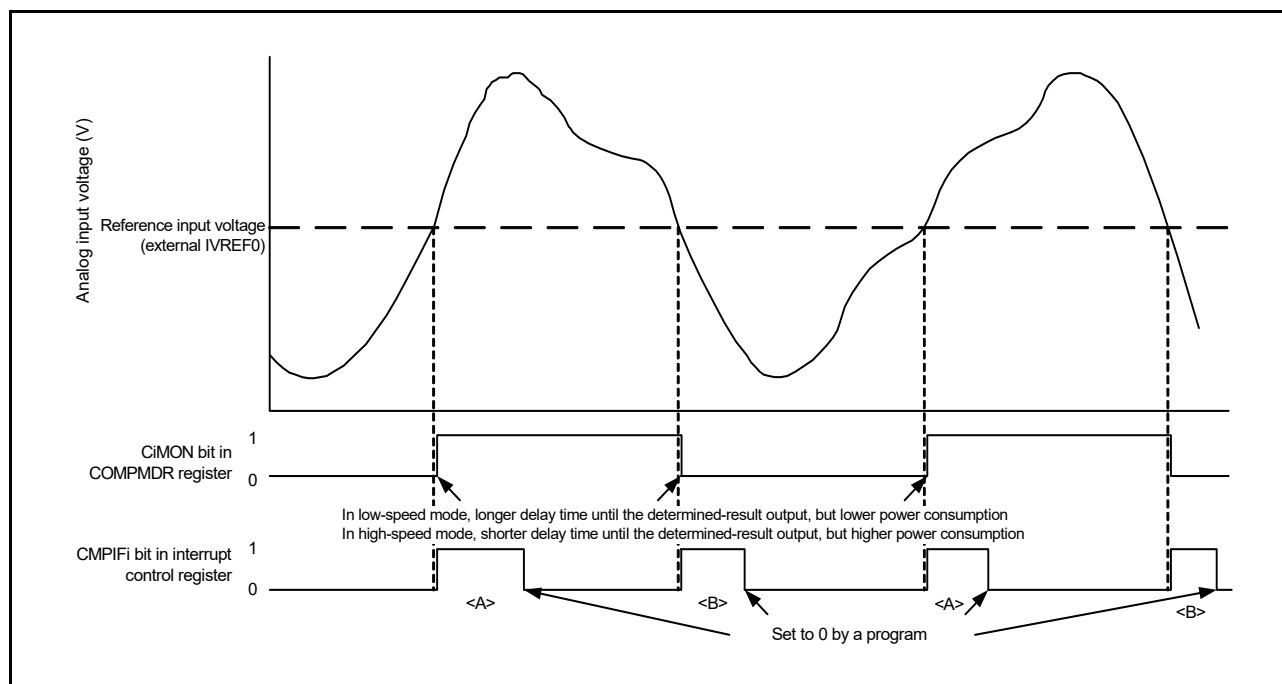
Note 2. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Caution To initialize all circuits of comparator i, set the CMPRES bit in the PRR1 register to 1.

Remark i = 0 in the 44- and 48-pin products
i = 0, 1 in the 52- to 100-pin products

Figure 18 - 7 shows an operation example of comparator i ($i = 0, 1$). In both low-speed mode and high-speed mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage. The input voltage to the IVREF0 pin is selected for use as the reference voltage.

Figure 18 - 7 Example of Comparator i Operation



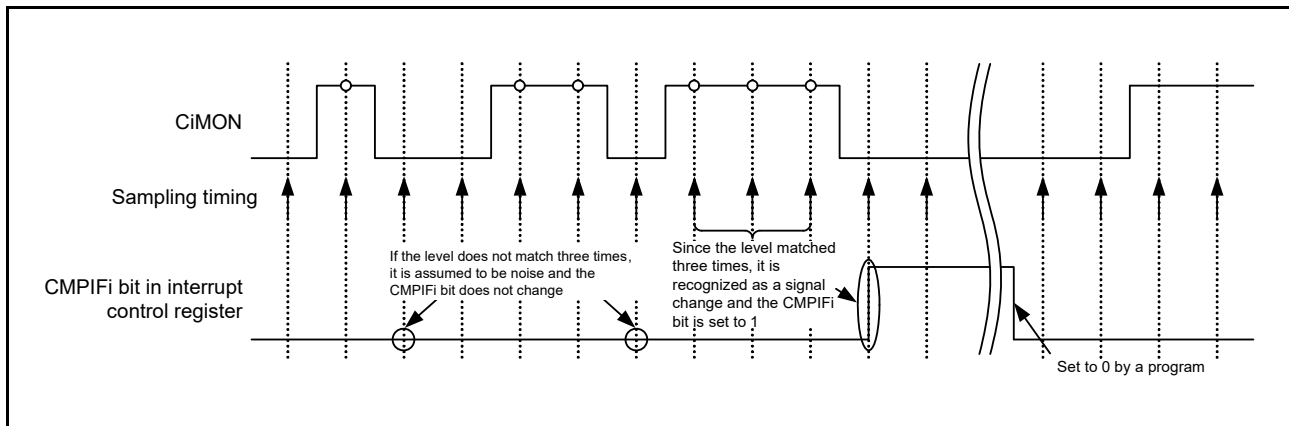
Caution The above diagram applies when CiFCK1 and CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPiFi changes as shown by <A> only. When CiEDG = 0 and CiEPO = 1 (falling edge), CMPiFi changes as shown by only.

Remark $i = 0$ in the 44- and 48-pin products
 $i = 0, 1$ in the 52- to 100-pin products

18.4.1 Digital Filter in Comparator i

Comparator i has a digital filter. The sampling clock can be selected by bits CiFCK1 and CiFCK0 in the COMPFIR register. The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock. **Figure 18 - 8** shows the example of digital filter and interrupt operations by comparator i.

Figure 18 - 8 Example of Digital Filter and Interrupt Operations by Comparator i



Caution The above operation example applies when bits CiFCK1 and CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).

Remark i = 0 in the 44- and 48-pin products
i = 0, 1 in the 52- to 100-pin products

18.4.2 Comparator i interrupts (i = 0, 1)

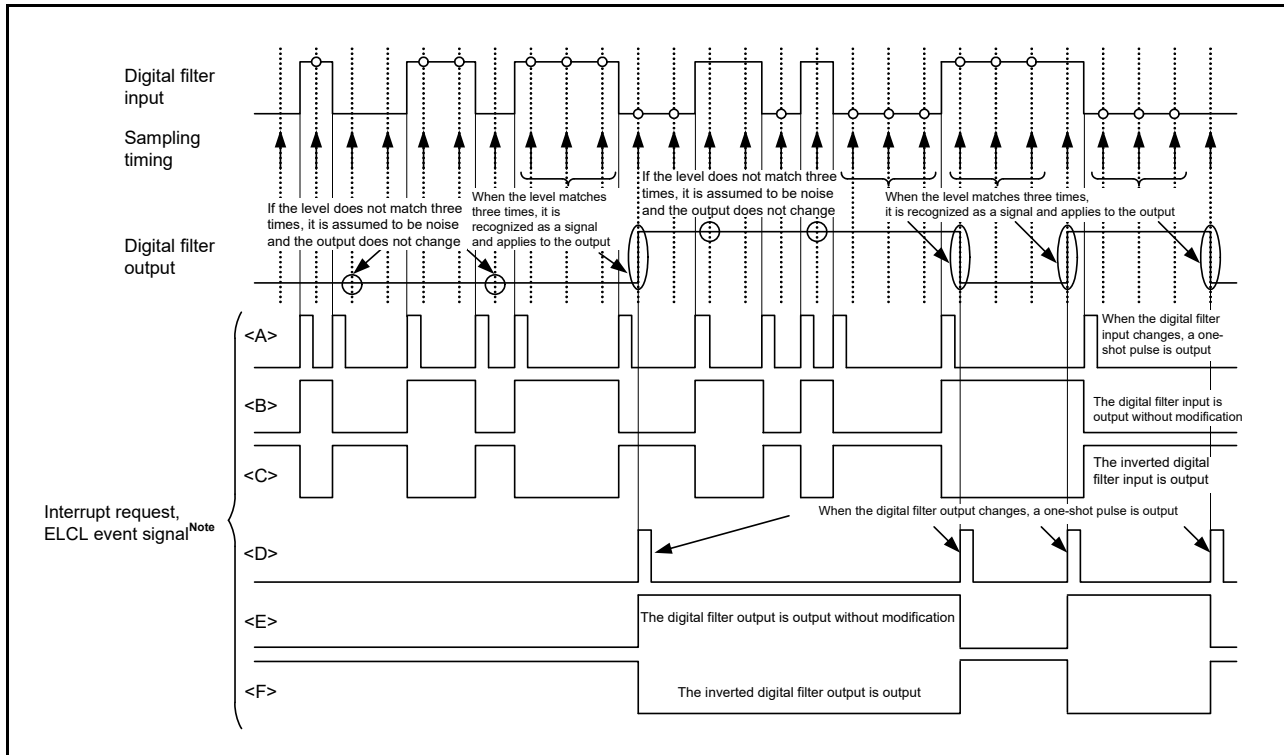
The comparator generates interrupt requests from two sources, comparator 0 and comparator 1. The comparator i interrupt each uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and a single vector. When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter. For details on the register setting and interrupt request generation, refer to **18.3.4 Comparator filter control register (COMPFIR)** and **18.3.5 Comparator output control register (COMPOCR)**.

Remark i = 0 in the 44- and 48-pin products
i = 0, 1 in the 52- to 100-pin products

18.4.3 Event signal output for the logic and event link controller (ELCL)

An event signal for the ELCL is generated by detecting the edge for the digital filter output set by the COMPFIR register, which is the same as the condition for interrupt request generation. However, unlike interrupt requests, the event signal for the ELCL is always output regardless of the setting of the CiIE bit in the COMPOCR register. Set the output signal select registers n (ELOSELn) and output signal enable register (ELOENCTL) of the ELCL to select the event output destination and to stop linking events.

Figure 18 - 9 Digital Filter and Interrupt Request/Event Signal Output for the ELCL



Note When the CiIE bit is 1, the same waveform is generated for an interrupt request and an ELCL event signal.
When the CiIE bit is 0, the value is fixed at 0 for an interrupt request only.

The waveforms of <A>, , and <C> are shown for an operation example when the CiFCK bits in the COMPFIR register are 00B (no digital filter). The waveforms <D>, <E>, and <F> are shown for an operation example when the CiFCK bits in the COMPFIR register are 01B, 10B, or 11B (digital filter enabled). <A> and <D> apply when the CiEDG bit is set to 1 (both edges), and <E> when the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), and <C> and <F> when the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge).

Remark i = 0 in the 44- and 48-pin products
i = 0, 1 in the 52- to 100-pin products

18.4.4 Comparator i output

The comparison result from the comparator can be output to external pins. Bits CiOP and CiOE in the COMPOCR register can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled. For the correspondence between the register setting and the comparator output, refer to **18.3.5 Comparator output control register (COMPOCR)**. To output the comparator comparison result to the VCOUTi output pin, use the following procedure to set the ports. Note that the ports are set to input after reset.

- <1> Set the mode for the comparator (steps 1 to 9 as listed in **Table 18 - 1 Procedure for Setting Comparator Associated Registers**).
- <2> Select the polarity of the VCOUT0 or VCOUT1 output and enable the output (step 11 as listed in **Table 18 - 1 Procedure for Setting Comparator Associated Registers**).
- <3> Set the corresponding port mode control register bit for the VCOUTi output pin to 0. Set the respective bits of the LCD port function register (PFSEGx) and LCD input switching control register (ISCLCD) to 0 and 1, respectively. The setting varies depending on the port pin. See **4.5.4 Examples of register settings for port and alternate functions**, for more details.
- <4> Set the corresponding port register bit for the VCOUTi output pin to 0.
- <5> Set the corresponding port mode register for the VCOUTi output pin to output (0) to start output from the pin.

Remark i = 0 in the 44- and 48-pin products
 i = 0, 1 in the 52- to 100-pin products

18.4.5 Stopping or supplying comparator clock

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

- <1> Set the CiIE bit (i = 0, 1) in the COMPOCR register to 0 (disabling a comparator interrupt).
- <2> Set the CiENB bit in the COMPMDR register to 0 (stopping the comparator).
- <3> Set the CMPIFi bit in the IF2H register to 0 (clearing any unnecessary interrupt before stopping the comparator).
- <4> Set the CMPEN bit in the PER1 register to 0.

When the clock is stopped by setting PER1, all the internal registers in the comparator are not initialized. The values of these registers are retained.

Caution When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, check the monitor flag (CiMON) of the comparator before enabling DTC activation as required.

- The comparator is set to an interrupt request on one-edge detection (CiEDG = 0), an interrupt request at the rising edge for the comparator (CiEPO = 0), and IVCMP > comparator i reference voltage
- The comparator is set to an interrupt request on one-edge detection (CiEDG = 0), an interrupt request at the falling edge for the comparator (CiEPO = 1), and IVCMP < comparator i reference voltage

Remark i = 0 in the 44- and 48-pin products
 i = 0, 1 in the 52- to 100-pin products

Section 19 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/L23 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.

<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	(CSI01)		(IIC01)
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	—	UART2 (supporting LIN-bus)	—
	1	CSI21		IIC21
	2	—	—	(IIC30)
	3	—		—

<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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	—	—	IIC30
	3	—		—

<52-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	—		—

<64-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<80-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<100-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

When UART0 is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 in channels 1 and 3 can be used.

Caution Most of the following descriptions in this section use the units and channels of the 100-pin products as an example.

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

19.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/L23 has the following features.

19.1.1 Simplified SPIs (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31)

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 **19.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) 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. fCLK/2 (CSI00 only)
Max. fCLK/4
During slave communication: Max. fMCK/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, simplified SPIs (CSIs) of following channels support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

- 44-pin products: CSI00
- 48- to 100-pin products: CSI00 and CSI20

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 42 Electrical Characteristics**.

Caution Use a general-purpose port pin to send a chip select signal when required.

19.1.2 UART (UART0 to UART3)

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 **19.6**

Operation of UART (UART0 to UART3) 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 and UART2 reception support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1).

[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 Only UART0 and UART2 support the 9-bit data length.

19.1.3 Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

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 **19.8 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)**

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

* The following functions are 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 **19.8.3 Data Reception, 2. Processing flow** for details.

Remark 1. To use an I²C bus of full function, see **Section 20 Serial Interface IICA (IICA)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.2 Configuration of Serial Array Unit

The serial array unit includes the following registers, and input and output pins.

Table 19 - 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, SCK10, SCK11, SCK20, SCK21, SCK30, SCK31 pins (for simplified SPI), SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCL30, SCL31 pins (for simplified I ² C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31 pins (for simplified SPI), RxD0, RxD1, RxD3 pins (for UART), RxD2 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31 pins (for simplified SPI), TxD0, TxD1, TxD3 pins (for UART), TxD2 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, SDA31 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 m (SSCm) • 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 mode registers (PMxx) • Port registers (Pxx) • Port input mode registers (PIMxx) • Port output mode registers (POMxx) • Port mode control A registers (PMCAxx) • Port mode control T registers (PMCTxx) • Port mode control E registers (PMCExx) • Port function output enable registers (PFOEx) • LCD port function registers (PFSEGx) • 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.

- mn = 00, 01, 10, 11: lower 9 bits
- Other than above: lower 8 bits

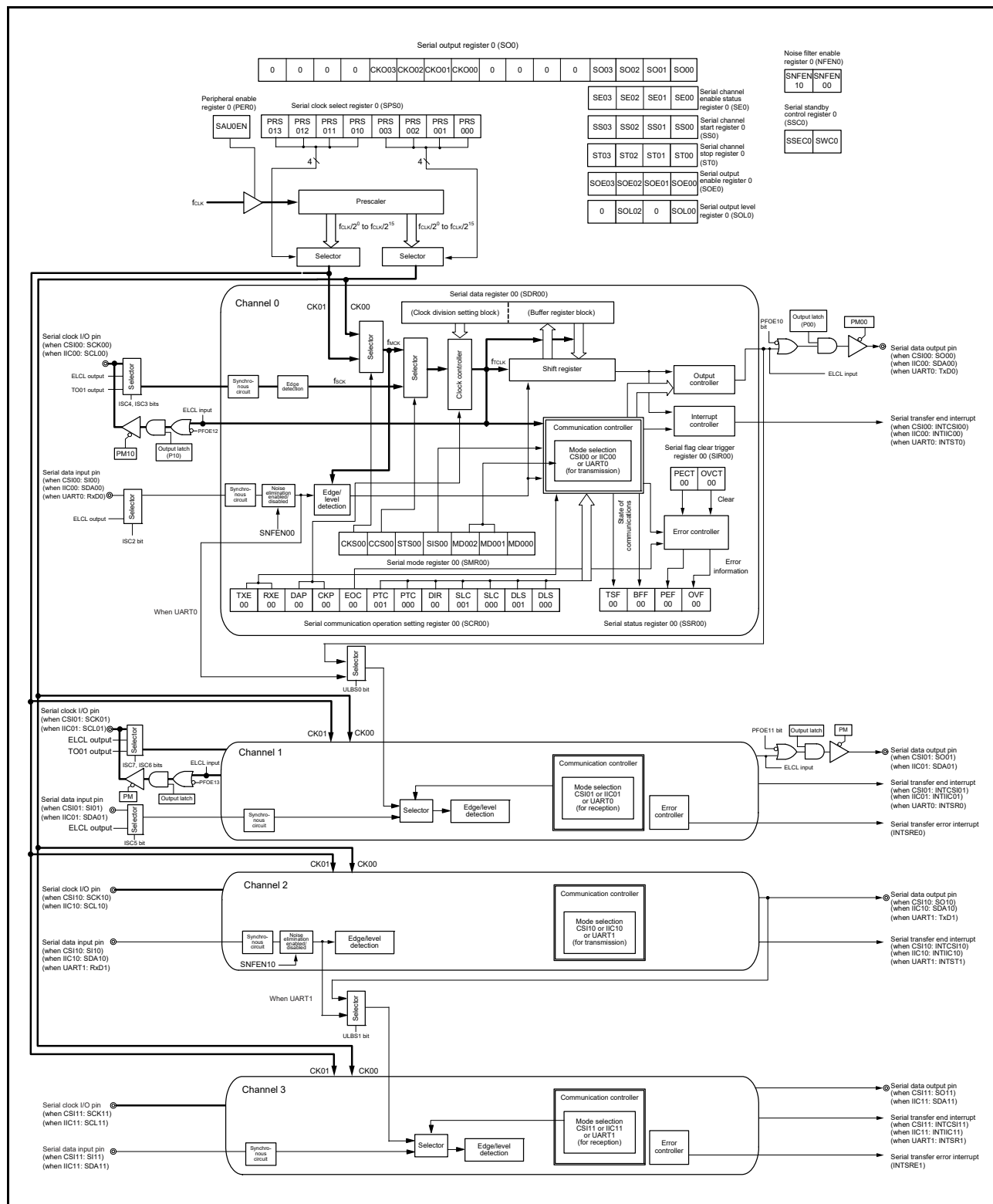
Note 2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- 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)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31)
q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

Figure 19 - 1 shows the block diagram of serial array unit 0.

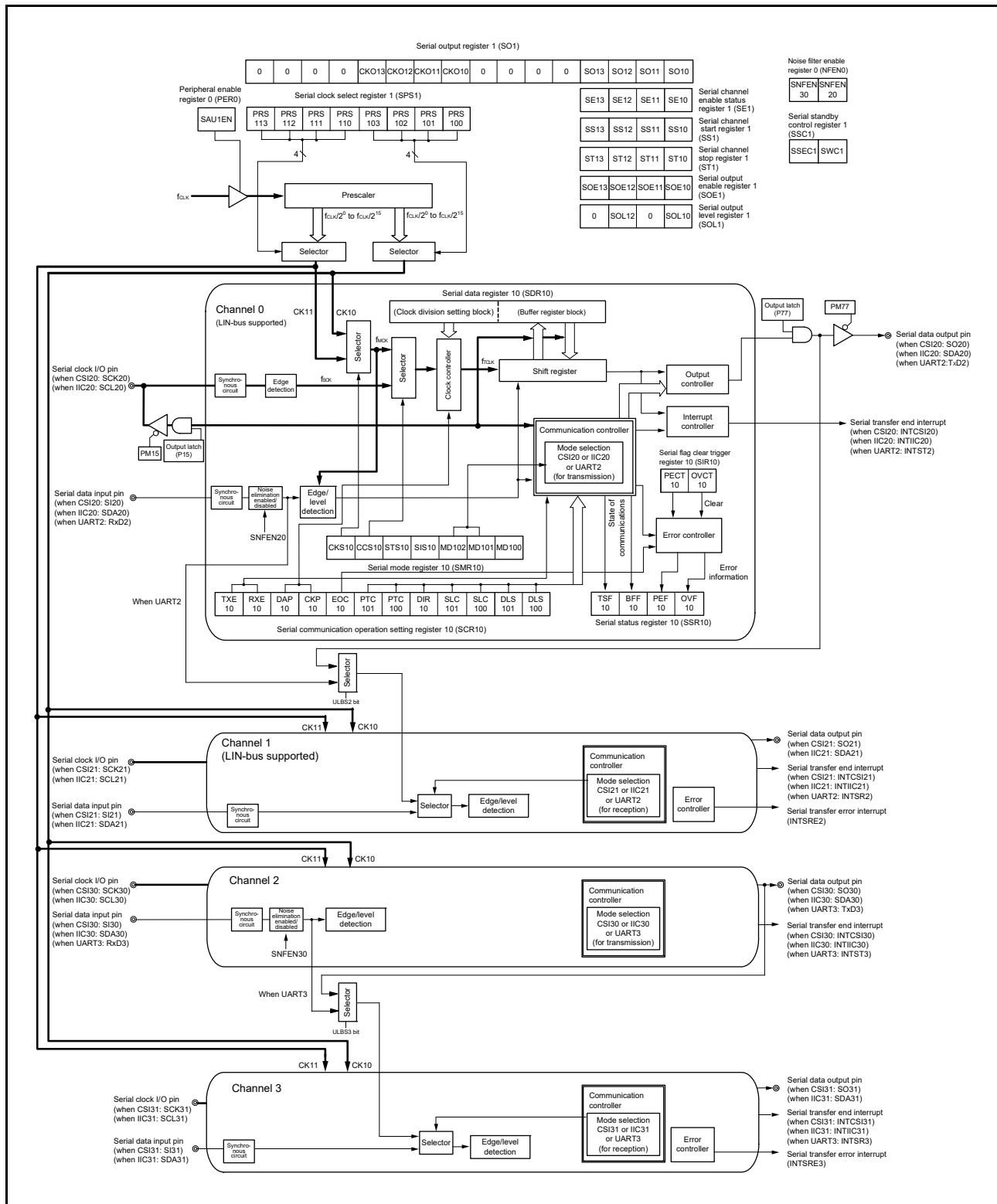
Figure 19 - 1 Block Diagram of Serial Array Unit 0



Remark The pins of serial array unit 0 when PIORx = 0 are shown in the above figure.

Figure 19 - 2 shows the block diagram of serial array unit 1.

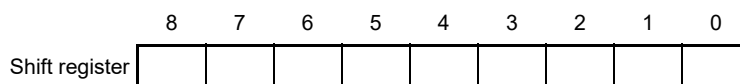
Figure 19 - 2 Block Diagram of Serial Array Unit 1



Remark The pins of serial array unit 1 when PIORx = 0 are shown in the above figure.

19.2.1 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 8 to 0) are used^{Note}. 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).



Note A 9-bit data length is only supported for UART0 and UART2.

19.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn)

SDRmn is a 16-bit transmission/reception data register 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 operation 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 1 and 0 (DLSmn1, DLSmn0) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 6 to 0 of the SDRmn register)
- 8-bit data length (stored in bits 7 to 0 of the SDRmn register)
- 9-bit data length (stored in bits 8 to 0 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. Only UART0 and UART2 support 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 8 to 0 in bit portions that exceed the data length.

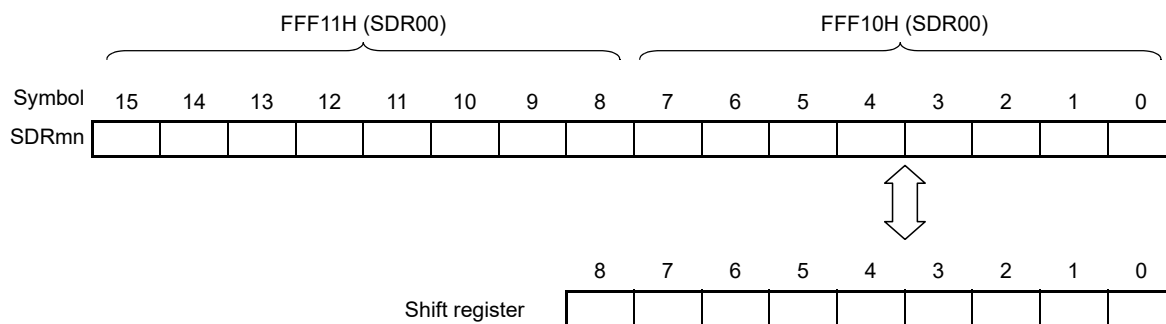
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31)
q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

Figure 19 - 3 Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H

R/W: R/W



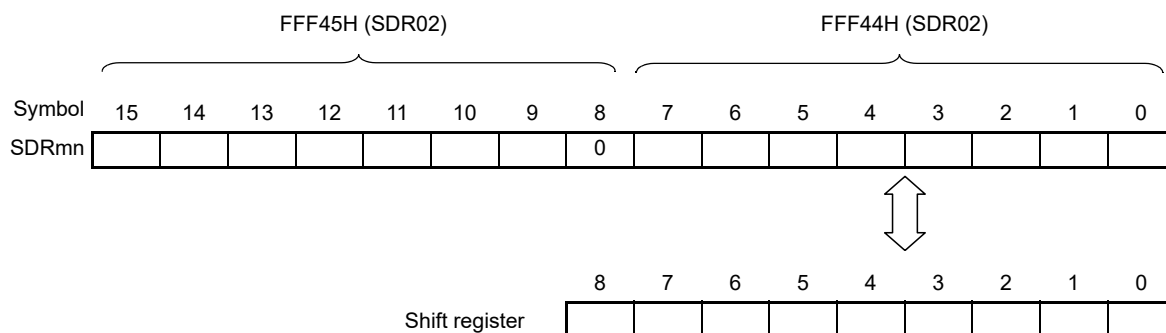
Remark For the function of the higher 7 bits of the SDRmn register, see **19.3 Registers for Controlling the Serial Array Unit**.

Figure 19 - 4 Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 12, 13)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)^{Note}

After reset: 0000H

R/W: R/W



Note This register is only present in the 52- to 100-pin products.

Caution Be sure to clear bit 8 to 0.

Remark For the function of the higher 7 bits of the SDRmn register, see **19.3 Registers for Controlling the Serial Array Unit**.

19.3 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 m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- LCD port function registers (PFSEGx)
- UART loopback select register (ULBS)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.3.1 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 19 - 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	IICA1EN	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.

Caution 1. Be sure to set the SAUmEN bit to 1 before making the serial array unit m settings. If SAUmEN = 0, writing to the registers for controlling serial array unit m is ignored.

- 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 m (SSCm)

Note, however, that writing to the following registers is valid.

- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable register 1 (PFOE1)
- LCD port function registers (PFSEGx)

Caution 2. Be sure to clear bit 1 to 0.

19.3.2 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 this register following a reset is 00H.

Figure 19 - 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	IICA1RES	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 Be sure to clear bits 7 and 1 to 0.

19.3.3 Serial clock select register m (SPSm)

SPSm is a 16-bit register that is used to select two types of operation clocks (CKm0, 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 19 - 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 operation 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

19.3.4 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 operation 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 19 - 8 Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03),
F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13)^{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 Note 2

7	6	5	4	3	2	1	0
0	SISmn0 Note 2	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (fmck) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register

Operation 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 operation 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 operation 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 19 - 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	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 to SMR03, SMR10, SMR11, and SMR12 are present in all products.
SMR13 is only present in the 52- to 100-pin products.

Note 2. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 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, SMR10, or SMR12 register) 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, 10, 11, 20, 21, 30, 31),
q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

19.3.5 Serial communication operation setting register mn (SCRmn)

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 19 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03),
F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13)**Note 1**

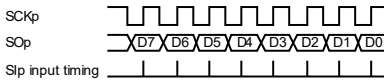
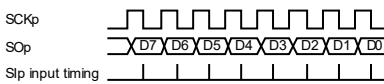
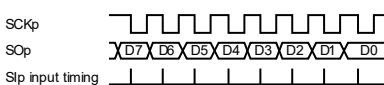
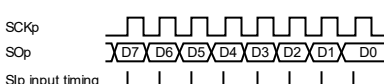
After reset: 0087H

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 Note 2	SLCmn0	0	1	DLSmn1 Note 3	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	1
0	1	SCKp	2
1	0	SCKp	3
1	1	SCKp	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 3))
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 19 - 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 (mn = 00, 02, 10, 12 only)
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 I ² C 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 8 to 0 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 6 to 0 of the SDRmn register)
1	1	8-bit data length (stored in bits 7 to 0 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 to SCR03, SCR10, SCR11, and SCR12 are present in all products.

SCR13 is only present in the 52- to 100-pin products.

Note 2. This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.

Note 3. This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

Note 4. An INTSREx error interrupt might occur unless the setting of EOCmn in CSImn is 0.

Note 5. 0 is always appended regardless of the data contents.

Caution Be sure to clear bits 11, 6, and 3 to 0 (Also clear bit 5 of the SCR01, SCR03, SCR11, or SCR13 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, 10, 11, 20, 21, 30, 31)

19.3.6 Serial data register mn (SDRmn)

SDRmn is a transmission/reception data register of channel n. Bits 8 to 0 (lower 9 bits) of SDR00, SDR01, SDR10, and SDR11 or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR12, and SDR13^{Note} 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 operation clock (fMCK). If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operation 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.

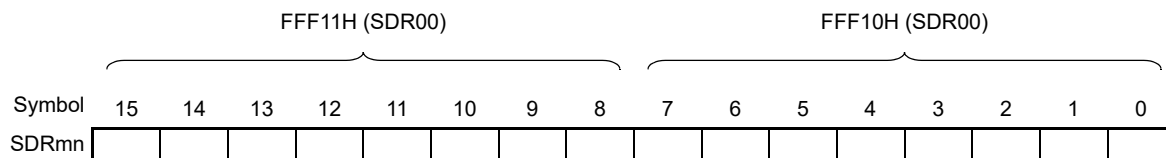
Note This register is only present in the 52- to 100-pin products.

Figure 19 - 10 Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

After reset: 0000H

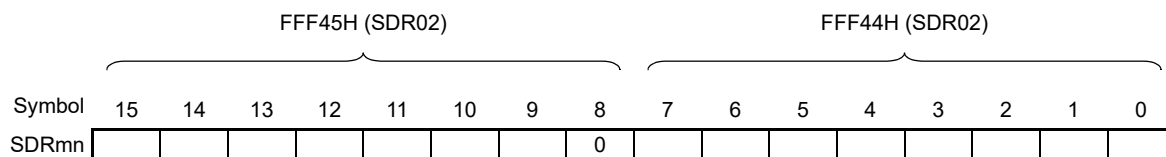
R/W: R/W



Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)^{Note}

After reset: 0000H

R/W: R/W



SDRmn[15:9]							Transfer clock setting by dividing the operation 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

Note This register is only present in the 52- to 100-pin products.

Caution 1. Be sure to clear bit 8 of the SDR02, SDR03, SDR12, and SDR13 registers 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 19.2 Configuration of Serial Array Unit.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.3.7 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is used to trigger clearing of each error flag of channel n. Setting each bit (FECTmn, PECTmn, and OVCTmn) of this register to 1 clears the corresponding bit (FEFmn, PEFmn, or OVFmn) of serial status register mn to 0. Each bit of this register is cleared once 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 19 - 11 Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03),
F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13)^{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 Note 2	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 to SIR03, SIR10, SIR11, and SIR12 are present in all products.
SIR13 is only present in the 52- to 100-pin products.

Note 2. This bit is only present in the SIR01, SIR03, SIR11, and SIR13 registers.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, or SIR12 register) to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. The value read from the SIRmn register is always 0000H.

19.3.8 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 19 - 12 Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03),
F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)^{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 Note 2	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 19 - 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 to SSR03, SSR10, SSR11, and SSR12 are present in all products.

SSR13 is only present in the 52- to 100-pin products.

Note 2. This bit is only present in the SSR01, SSR03, SSR11, and SSR13 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 (SWCm = 1), the OVFmn and BFFmn flags will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.3.9 Serial channel start register m (SSm)

The SSm register is used to trigger starting of communication/counting by each channel. Writing 1 to a bit (SSmn) of this register sets the corresponding bit (SEmn) of serial channel enable status register m (SEm) to 1 (enabling operation). Each bit of this register is cleared once the corresponding bit of the SEmn register is set to 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 an 1-bit or 8-bit memory manipulation instruction with SSmL. The value of each SSm register following a reset is 0000H.

Figure 19 - 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	SS13	SS12	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, bits 15 to 3 of the SS1 register for 44- and 48-pin products, and bits 15 to 4 of the SS1 register for 52- to 100-pin products 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 four cycles of fMCK have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. The value read from the SSm register is always 0000H.

19.3.10 Serial channel stop register m (STm)

The STm register is used to trigger stoppage of communication/counting by each channel. Writing 1 to a bit (STmn) of this register clears the corresponding bit (SEmn) of serial channel enable status register m (SEm) to 0 (stopping operation). Each bit of this register is cleared once the corresponding bit of the SEMn register is cleared to 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 19 - 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	ST13	ST12	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, bits 15 to 3 of the ST1 register for 44- and 48-pin products, and bits 15 to 4 of the ST1 register for 52- to 100-pin products 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.

19.3.11 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 in which 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 in which 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 SE_mL. The value of each SEm register following a reset is 0000H.

Figure 19 - 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	SE13	SE12	SE11	SE10

SE _m _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)

19.3.12 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 in which 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, in which 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 19 - 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	SOE13	SOE12	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, bits 15 to 3 of the SOE1 register for 44- and 48-pin products, and bits 15 to 4 of the SOE1 register for 52- to 100-pin products to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.3.13 Serial output register m (SOM)

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 CKOMn 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 CKOMn 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 CKOMn 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 CKOMn 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 19 - 17 Format of Serial Output Register m (SOM)

Address: F0128H, F0129H (SO0)

After reset: 0F0FH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO0	0	0	0	0	CKO03	CKO02	CKO01	CKO00
	7	6	5	4	3	2	1	0
	0	0	0	0	SO03	SO02	SO01	SO00

Address: F0168H, F0169H (SO1)

After reset: 0F0FH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO1	0	0	0	0	CKO13	CKO12	CKO11	CKO10
	7	6	5	4	3	2	1	0
	0	0	0	0	SO13	SO12	SO11	SO10

CKOMn	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 and 7 to 4 of the SO0 register to 0. Be sure to clear bits 15 to 11 and 7 to 3 of the SO1 register for 44- and 48-pin products and bits 15 to 12 and 7 to 4 of the SO1 register for 52- to 100-pin products to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

19.3.14 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 19 - 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

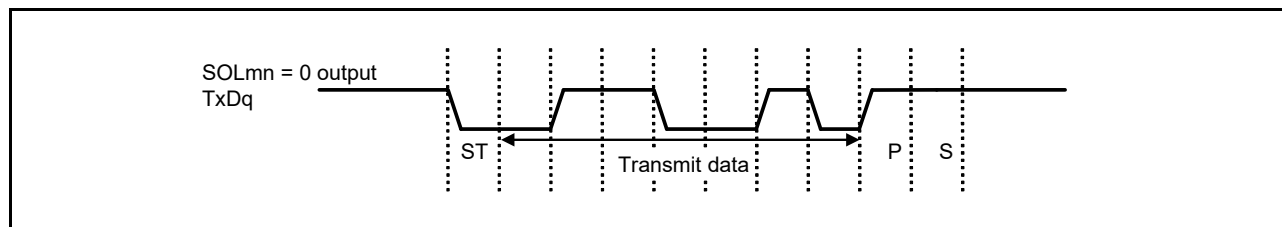
Symbol	15	14	13	12	11	10	9	8
SOL1	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	SOL12	0	SOL10
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode							
0	Communication data is output as is.							
1	Communication data is inverted and output.							

Caution Be sure to clear bits 15 to 3 and 1 of the SOL0 register, bits 15 to 1 of the SOL1 register for 44- and 48-pin products, and bits 15 to 3 and 1 of the SOL1 register for 52- to 100-pin products to 0.

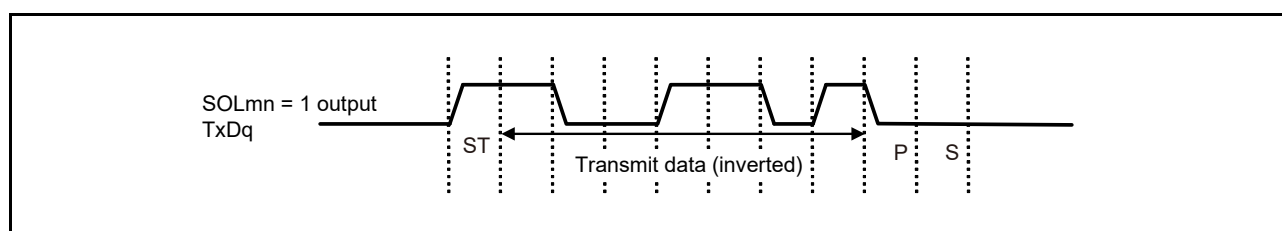
Figure 19 - 19 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 19 - 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)

19.3.15 Serial standby control register m (SSCm)

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 SSC1 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI20 or UART2 serial data. The SSCm register can be set by a 16-bit memory manipulation instruction. The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL. The value of each SSCm register following a reset is 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- When using CSI00, CSI20: Up to 1 Mbps
- When using UART0, UART2: Up to 115.2 kbps (when setting FWKUP = 1, fCLK = fIH (32 MHz))

Figure 19 - 20 Format of Serial Standby Control Register m (SSCm)

Address: F0138H, F0139H (SSC0), F0178H, F0179H (SSC1)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SSCm	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SSECm	SWCm
SSECm	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/INTSRE2).							
1	Disable the generation of error interrupts (INTSRE0/INTSRE2).							
• The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0. • Setting SSECm, SWCm = 1, 0 is prohibited.								
SWCm	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 SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.								

Table 19 - 2 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

19.3.16 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. When bit 2 is set to 1, the ELCL output signal can be selected as the serial data input source of CSI00. Bits 3 and 4 can be used to select the SCK00 pin input, ELCL output signal, or TO01 output signal as the serial clock input source of CSI00. When bit 5 is set to 1, the ELCL output signal can be selected as the serial data input source of CSI01. Bits 6 and 7 can be used to select the SCK01 pin input, ELCL output signal, 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 19 - 21 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
	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	ELCL output signal					
	1	0	TO01 output signal					
	1	1	Setting prohibited					
	ISC5	Switch of the serial data input source of CSI01 Note 2						
	0	Input signal of the SI01 pin (normal operation)						
	1	ELCL output signal						
	ISC4	ISC3	Switch of the serial clock input source of CSI00 Note 3					
	0	0	Input signal of the SCK00 pin (normal operation)					
	0	1	ELCL output signal					
	1	0	TO01 output signal					
	1	1	Setting prohibited					
	ISC2	Switch of the serial data input of CSI00/UART0 Note 4						
	0	Input signal of the SI00/RxD0 pin (normal operation)						
	1	ELCL output signal						
	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 1, set the ISC5 bit to 0.

Note 3. When UART mode or simplified I²C mode is to be selected for channel 0, set the ISC4 and ISC3 bits to 0.

Note 4. When UART mode or simplified I²C mode is to be selected for channel 0, set the ISC2 bit to 0.

19.3.17 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 operation 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 operation 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 19 - 22 Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD3 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the other than RxD3 pin.	

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.

19.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- LCD port function registers (PFSEGx)

For details, 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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.14 Port function output enable registers (PFOEx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting registers that are used to control the port functions, see **4.5.4 Examples of register settings for port and alternate functions**.

When connecting 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**.

19.3.19 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 19 - 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	ULBS7	ULBS6	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0
ULBSn	Selection of the UART loopback function							
0	Inputs the states of the RxD0 to RxD3 pins of serial array unit UART0 to UART3 to the reception shift register.							
1	Loops back output from the transmission shift register to the reception shift register.							

Remark n = 0 to 3

19.4 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.

19.4.1 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 19 - 24 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.

	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCWEN x	IICA1EN x	ADCEN x	IICA0EN x	SAU1EN 0/1	SAU0EN 0/1	0 x	TAU0EN x

Control of SAUm input clock
 0: Stops supply of input clock
 1: Supplies input clock

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 mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Caution 2. Be sure to clear bit 1 to 0.

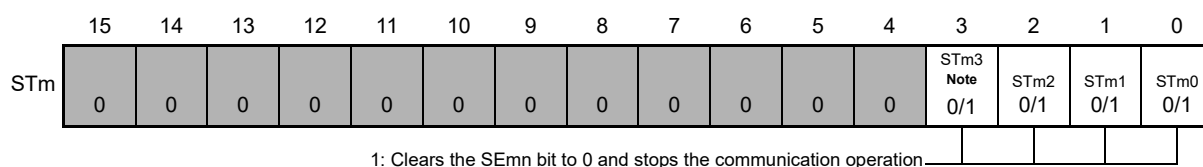
Remark x: Bits not used with serial array units (depending on the settings of other peripheral functions)
 0/1: Set to 0 or 1 depending on the usage of the user

19.4.2 Stopping the Operation by Channels

The stopping of the operation by channels is set using each of the following registers.

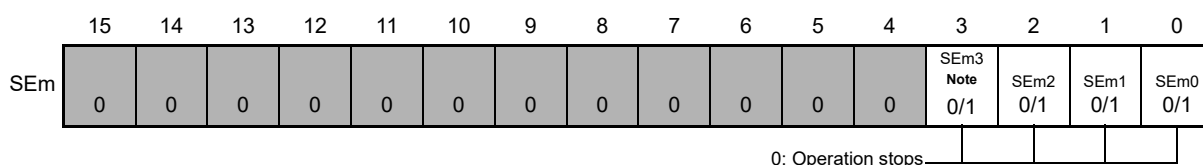
Figure 19 - 25 Each Register Setting When Stopping the Operation by Channels

- a) Serial channel stop register m (STm): The STm register is used to trigger stoppage of communication/counting by each channel.



* STmn is cleared once SEmn is cleared to 0.

- b) Serial channel enable status register m (SEm): This register indicates whether data transmission/reception operation of each channel is enabled or stopped.



* SEm is a read-only status register. Operation of each channel can be stopped by using the STm register.

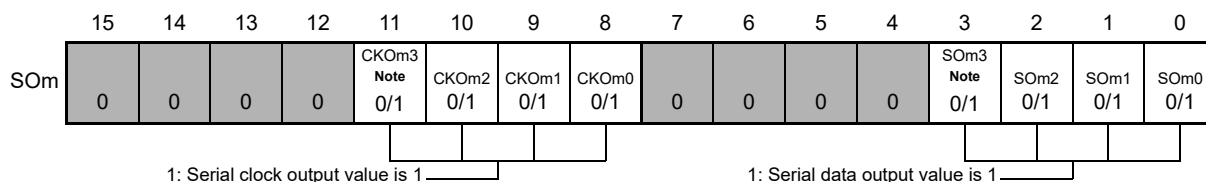
For channel n, in which serial output is stopped, the SOMn bit value of the SOM register can be set by software.

- 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, in which serial output is stopped, the SOMn bit value of the SOM register can be set by software.

- d) Serial output register m (SOM): 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 and SOM bits to 1.

Note For serial array unit 1, this bit is only present in the 52- to 100-pin products.

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

19.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) 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, simplified SPIs (CSIs) of following channels support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

- 44-pin products: CSI00
- 48- to 100-pin products: CSI00, CSI20

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 42 Electrical Characteristics**.

Caution Use a general-purpose port pin to send a chip select signal when required.

The channels supporting simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) are channels 0 to 3 of SAU0 and channels 0 to 3 of SAU1.

<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	(CSI01)		(IIC01)
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	—	UART2 (supporting LIN-bus)	—
	1	CSI21		IIC21
	2	—	—	(IIC30)
	3	—		—

<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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	—	—	IIC30
	3	—		—

<52-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	—		—

<64-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<80-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<100-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx).
For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) performs the following seven types of communication operations.

- Master transmission (See **19.5.1.**)
- Master reception (See **19.5.2.**)
- Master transmission/reception (See **19.5.3.**)
- Slave transmission (See **19.5.4.**)
- Slave reception (See **19.5.5.**)
- Slave transmission/reception (See **19.5.6.**)
- SNOOZE mode function (See **19.5.7.**)

19.5.1 Master Transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21	SCK30, SO30	SCK31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

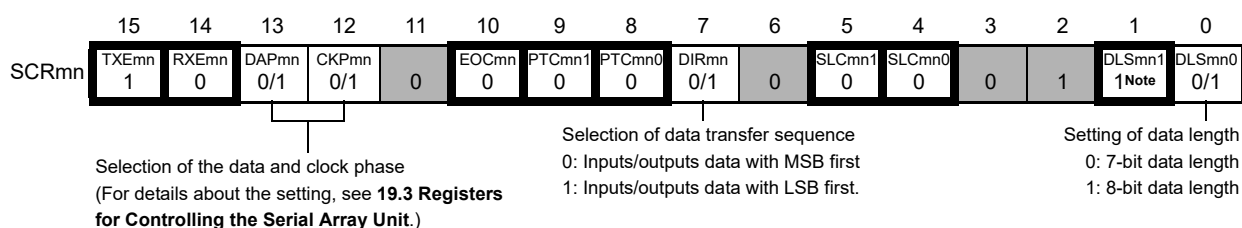
1. Register setting

Figure 19 - 26 Example of Contents of Registers for Master Transmission of Simplified SPI
(CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

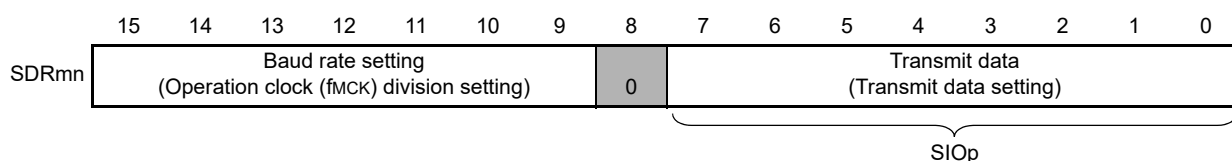
a) Serial mode register mn (SMRmn)



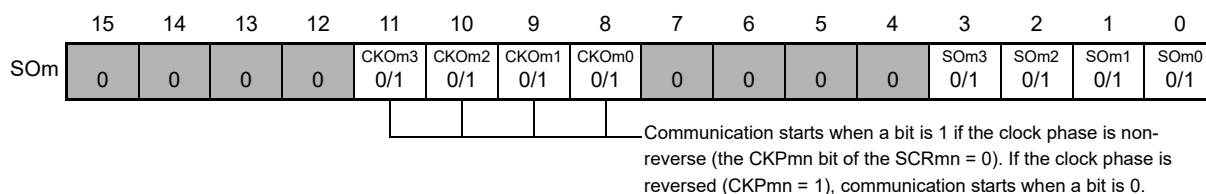
b) Serial communication operation setting register mn (SCRmn)



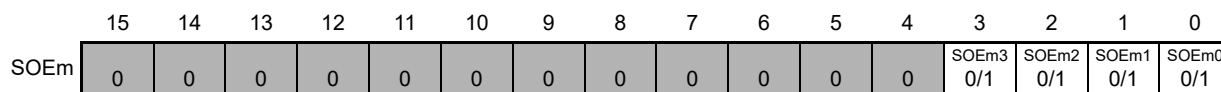
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



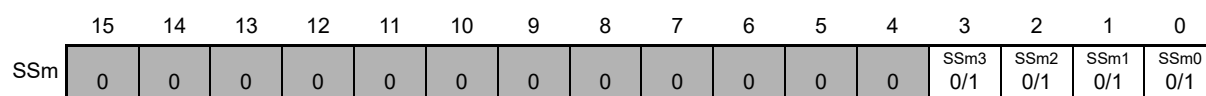
d) Serial output register m (SOM): Set only the bit of the target channel.



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.



Note This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

(Remarks continue on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

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 19 - 27 Initial Setting Procedure for Master Transmission

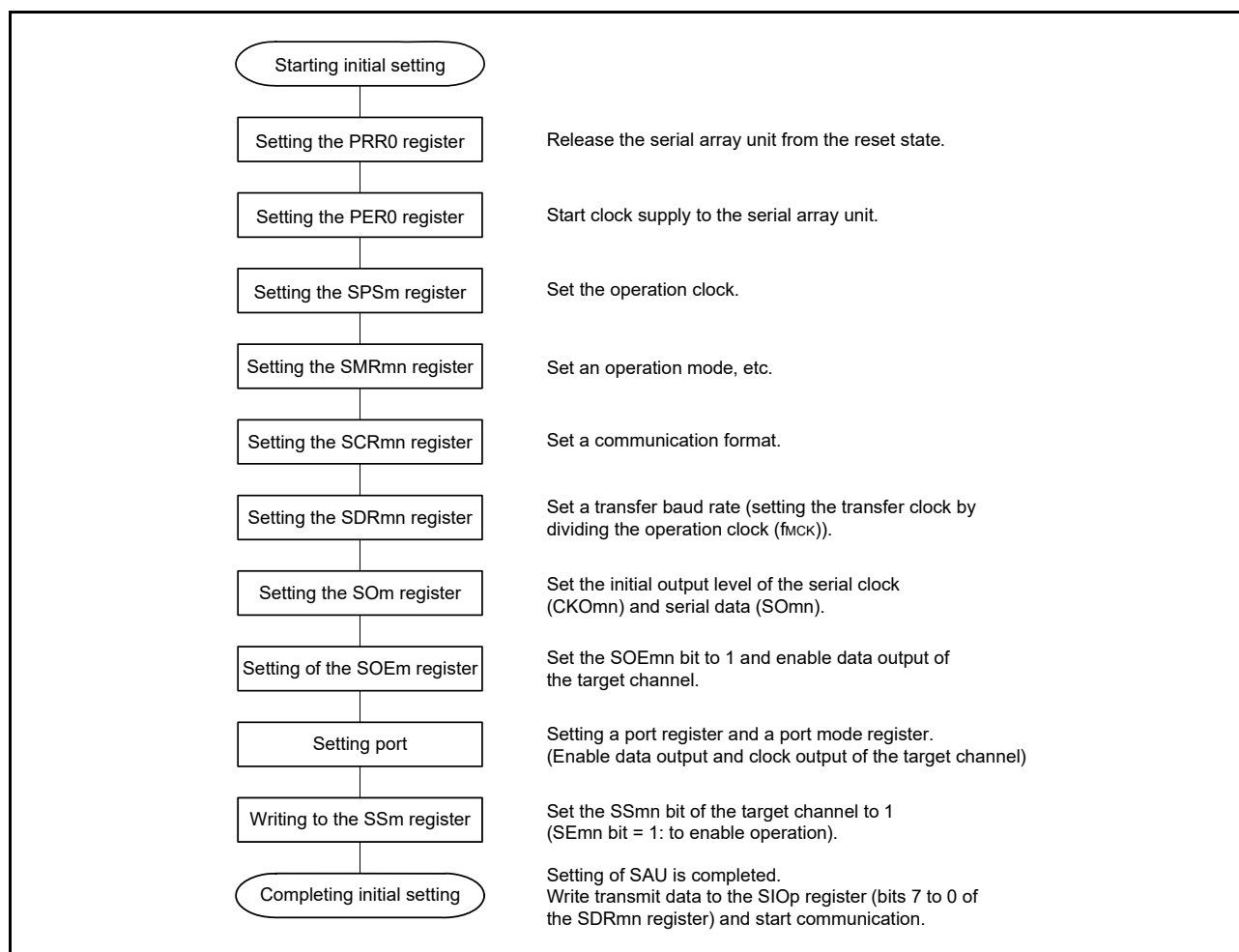


Figure 19 - 28 Procedure for Stopping Master Transmission

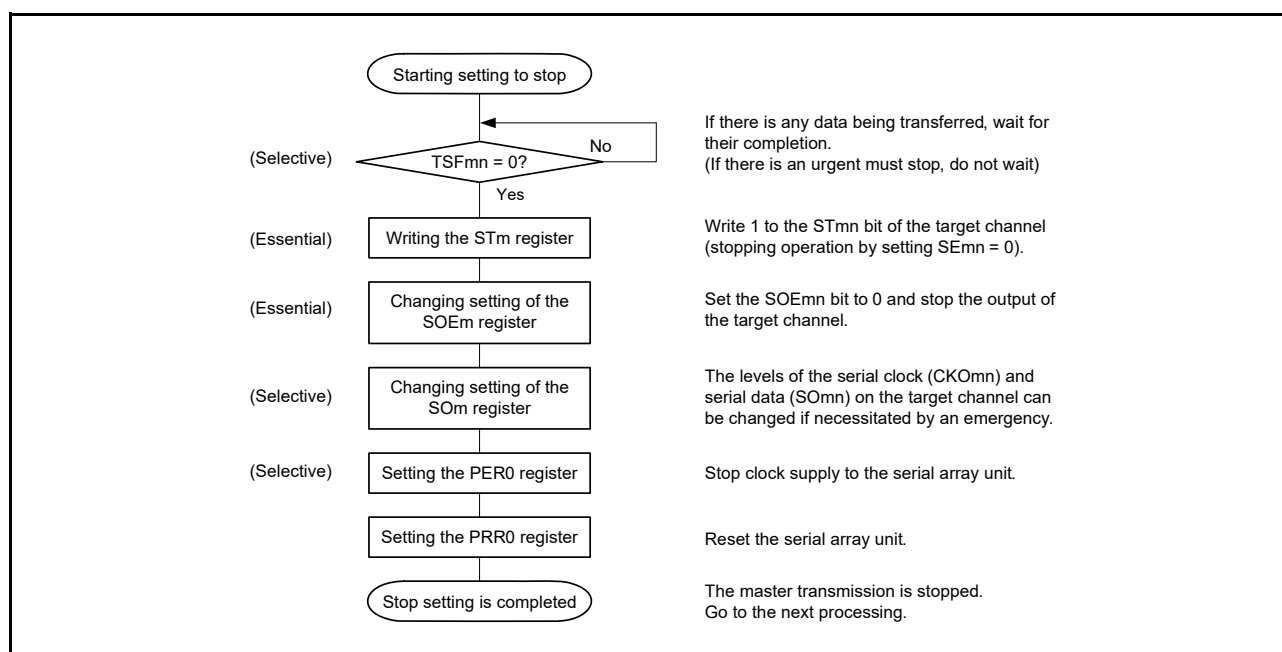
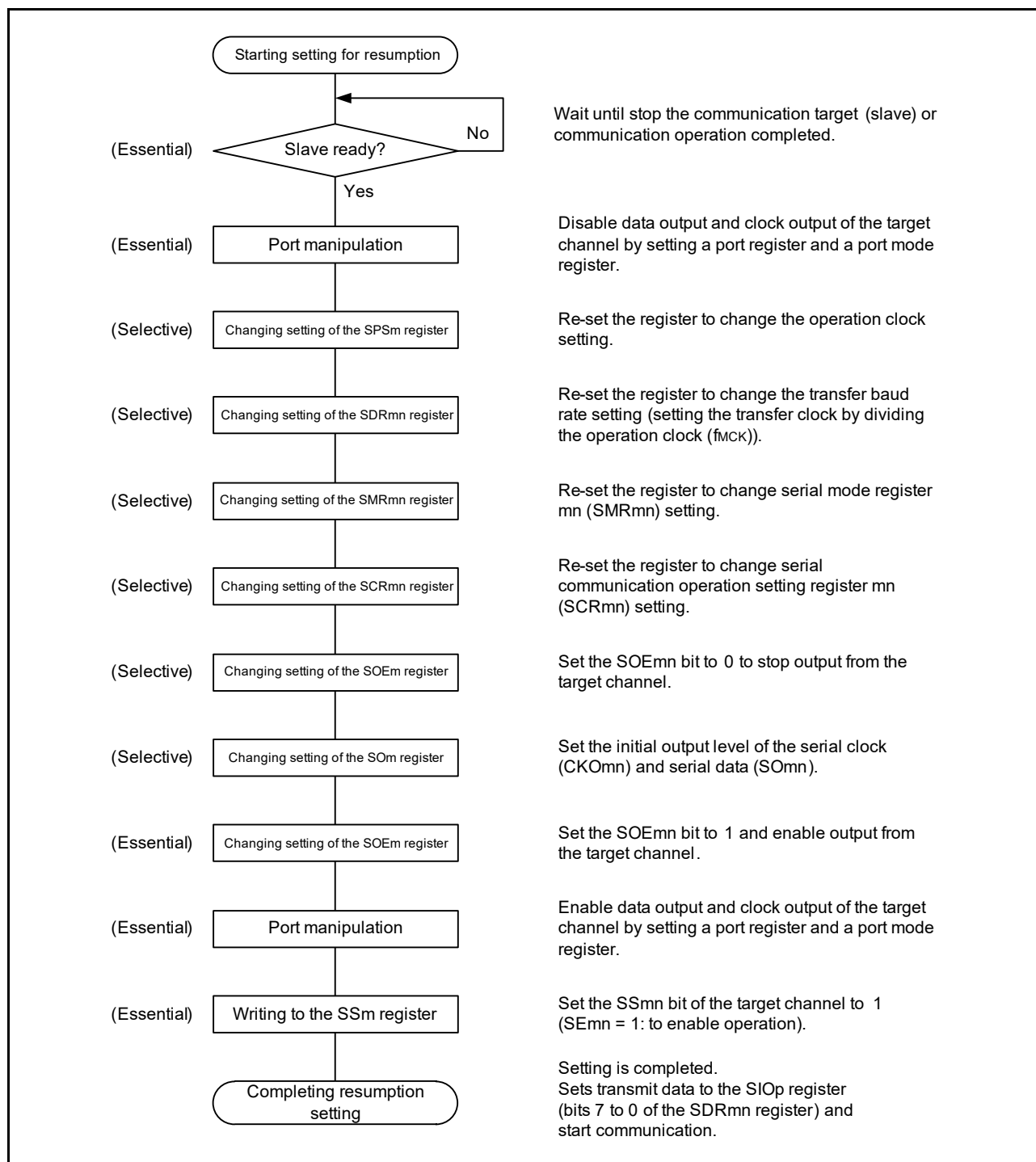


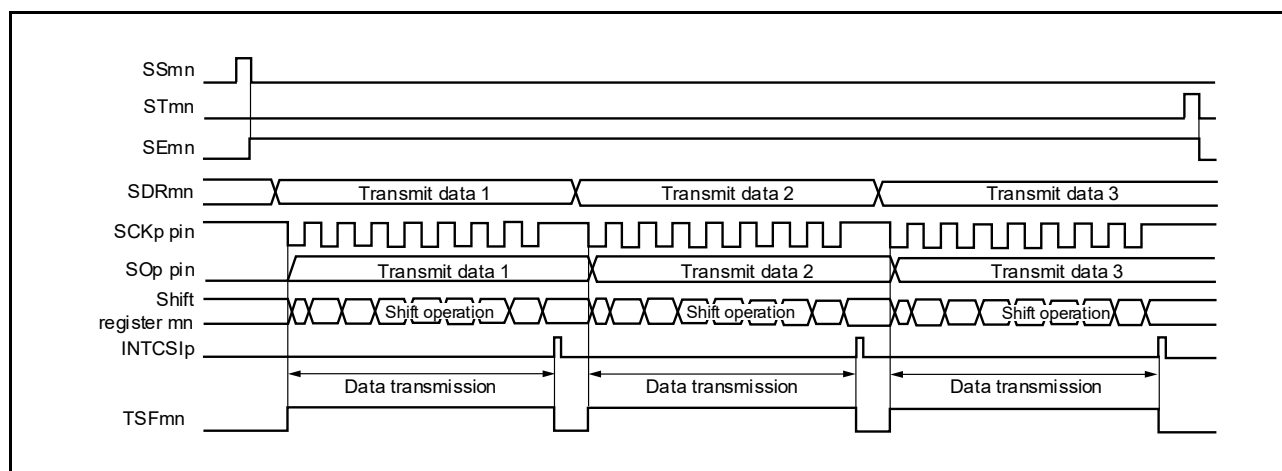
Figure 19 - 29 Procedure for Resuming Master Transmission



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 with initialization instead of restarting the communication.

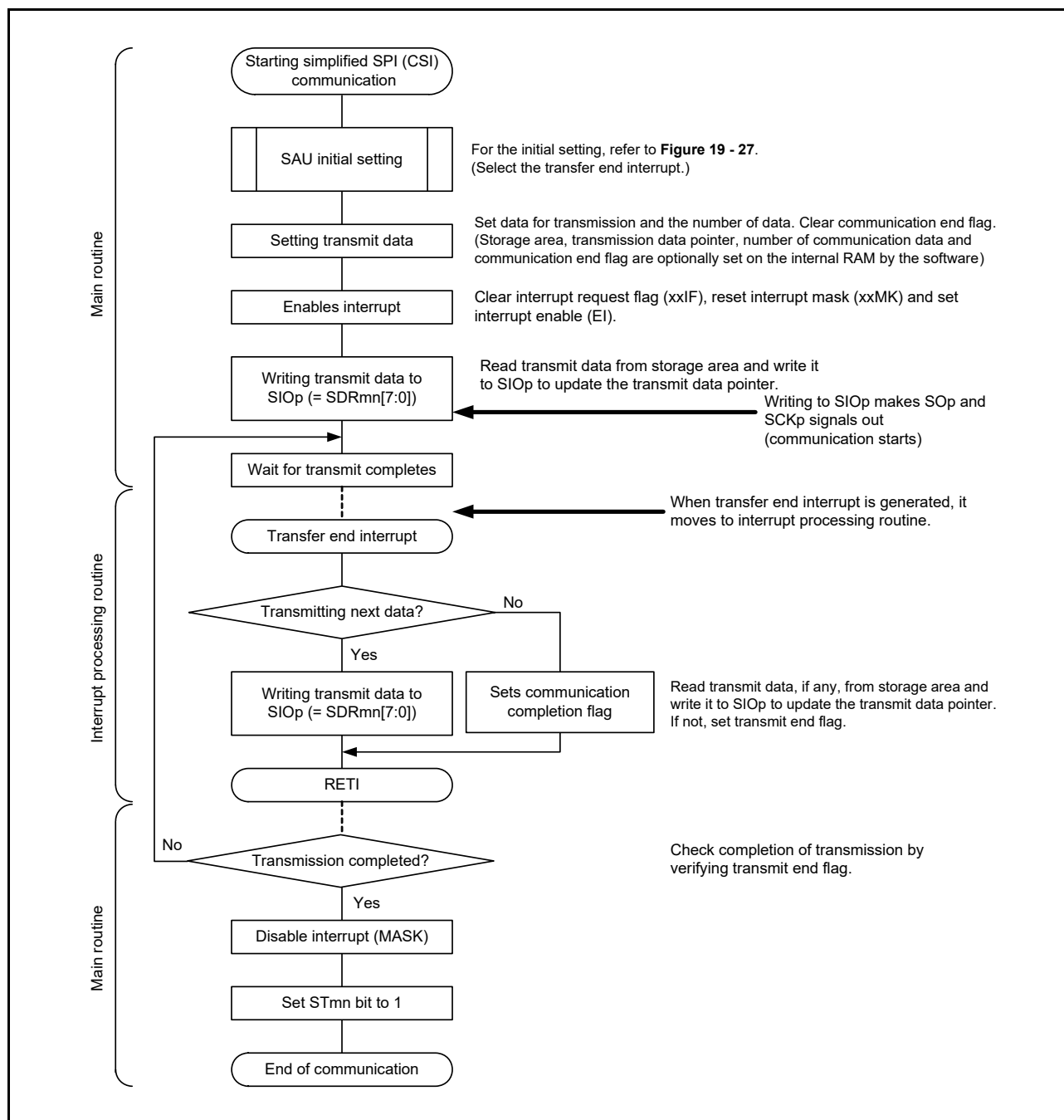
3. Processing flow (in single-transmission mode)

Figure 19 - 30 Timing Chart of Master Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



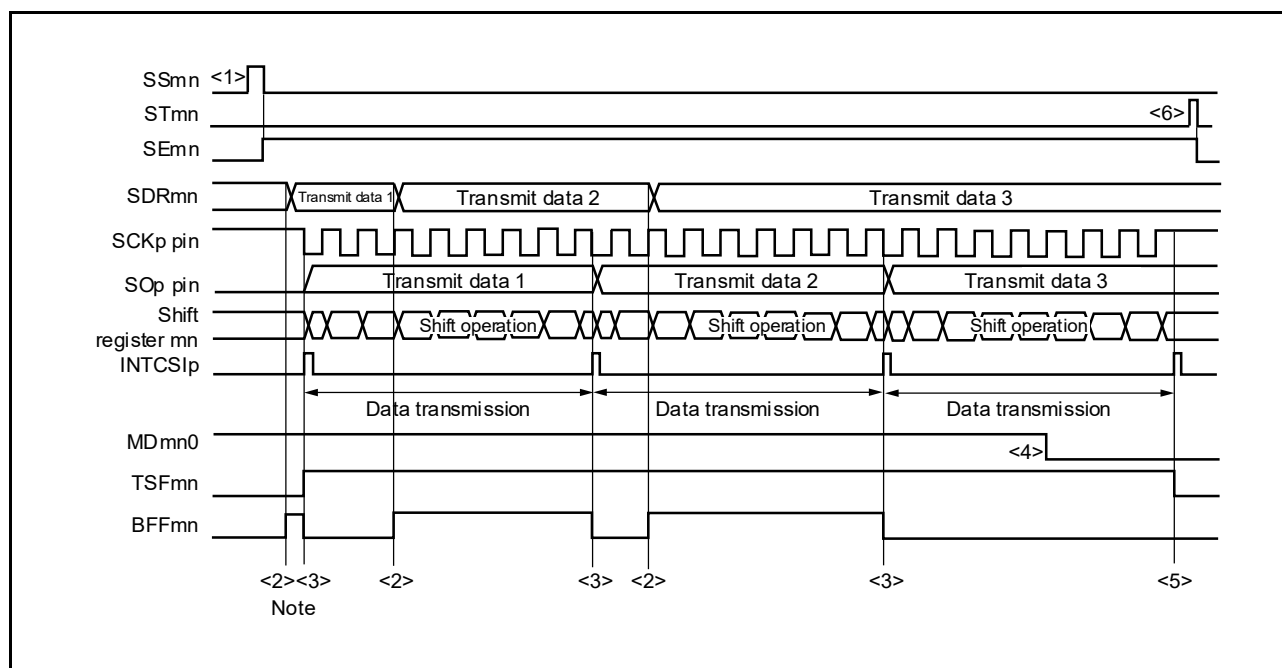
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 31 Flowchart of Master Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 19 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

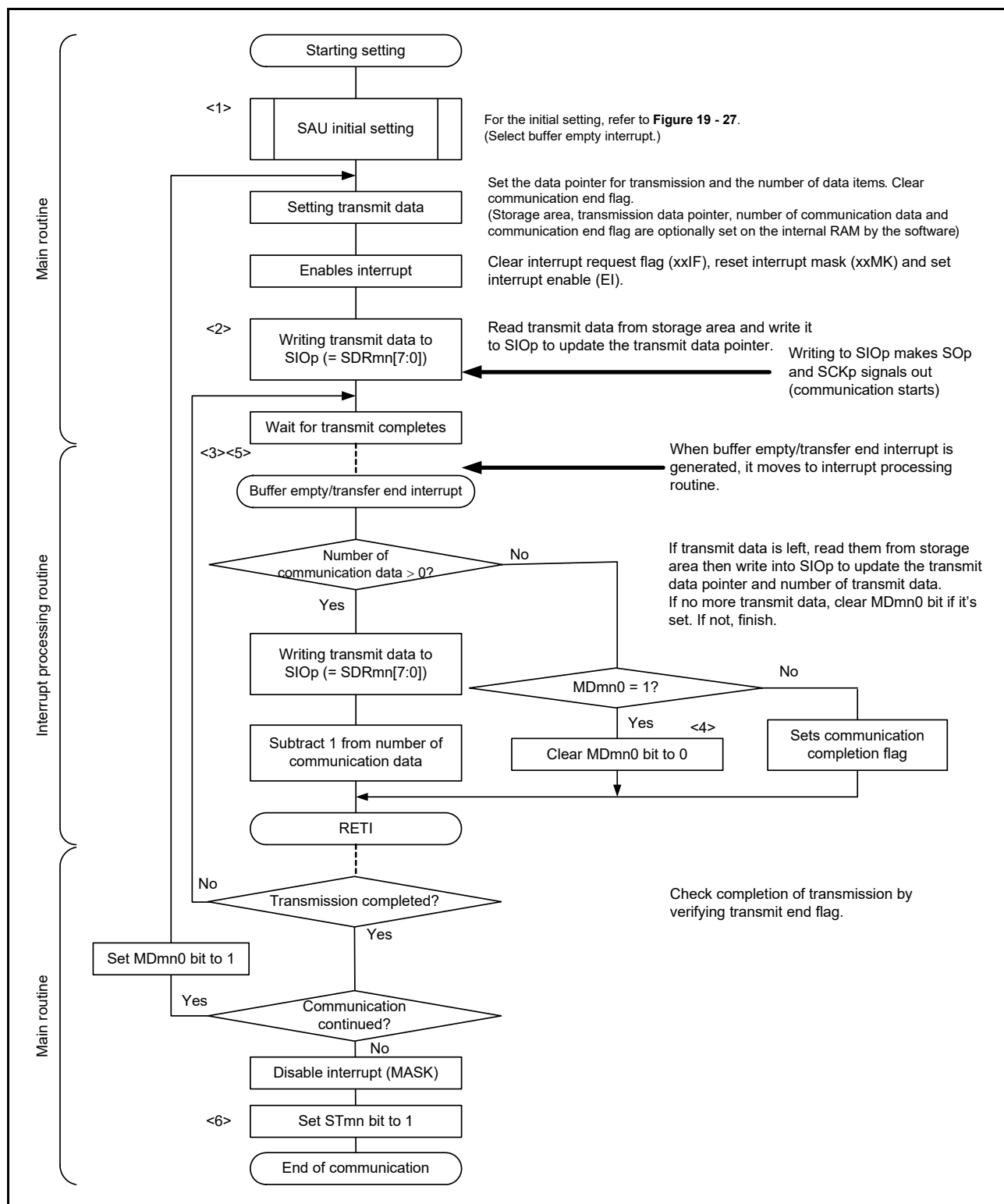


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 33 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 19 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

19.5.2 Master Reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21	SCK30, SI30	SCK31, SI31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.							
Error detection flag	Overflow 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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

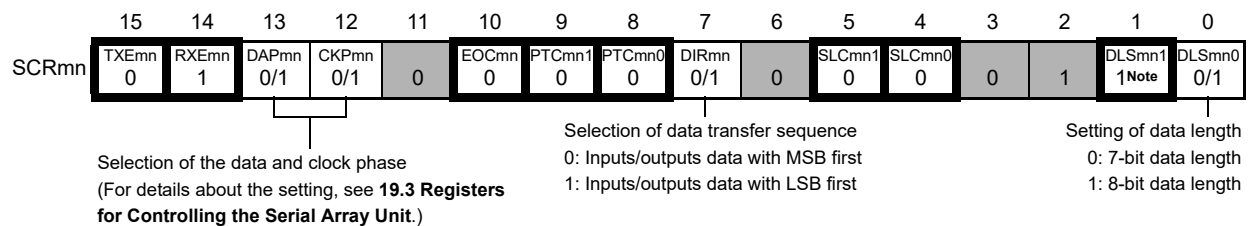
1. Register setting

Figure 19 - 34 Example of Contents of Registers for Master Reception of Simplified SPI
(CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

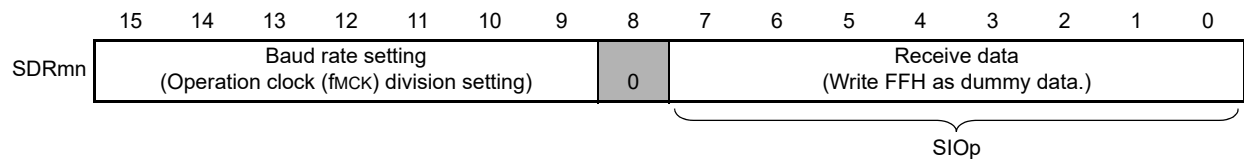
a) Serial mode register mn (SMRmn)



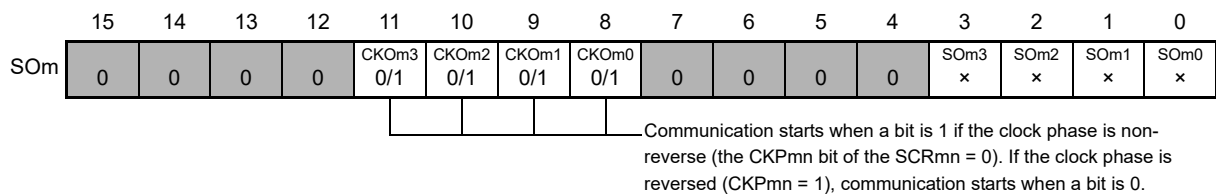
b) Serial communication operation setting register mn (SCRmn)



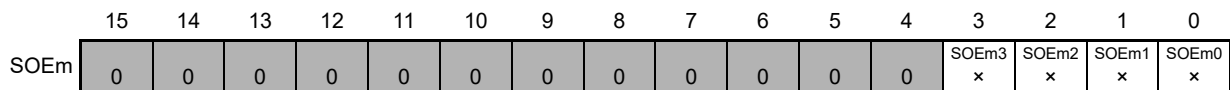
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



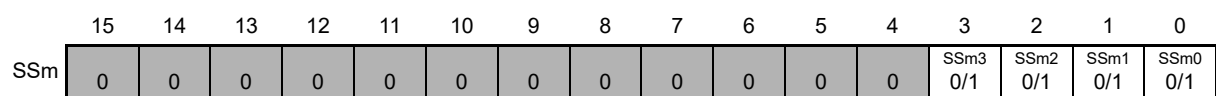
d) Serial output register m (SOM): Set only the bit of the target channel.



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.



Note This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

(Remarks continue on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

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 19 - 35 Initial Setting Procedure for Master Reception

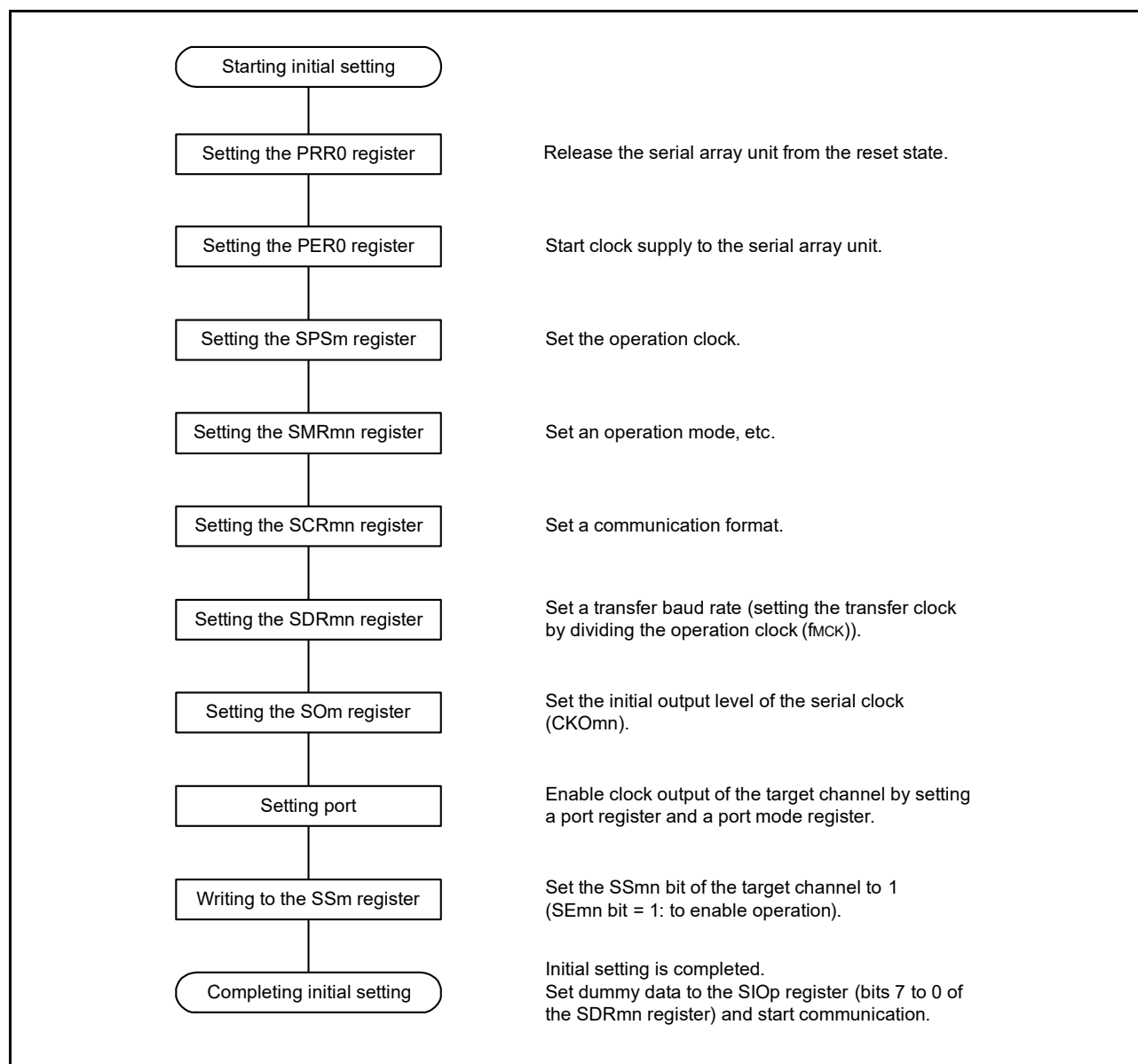


Figure 19 - 36 Procedure for Stopping Master Reception

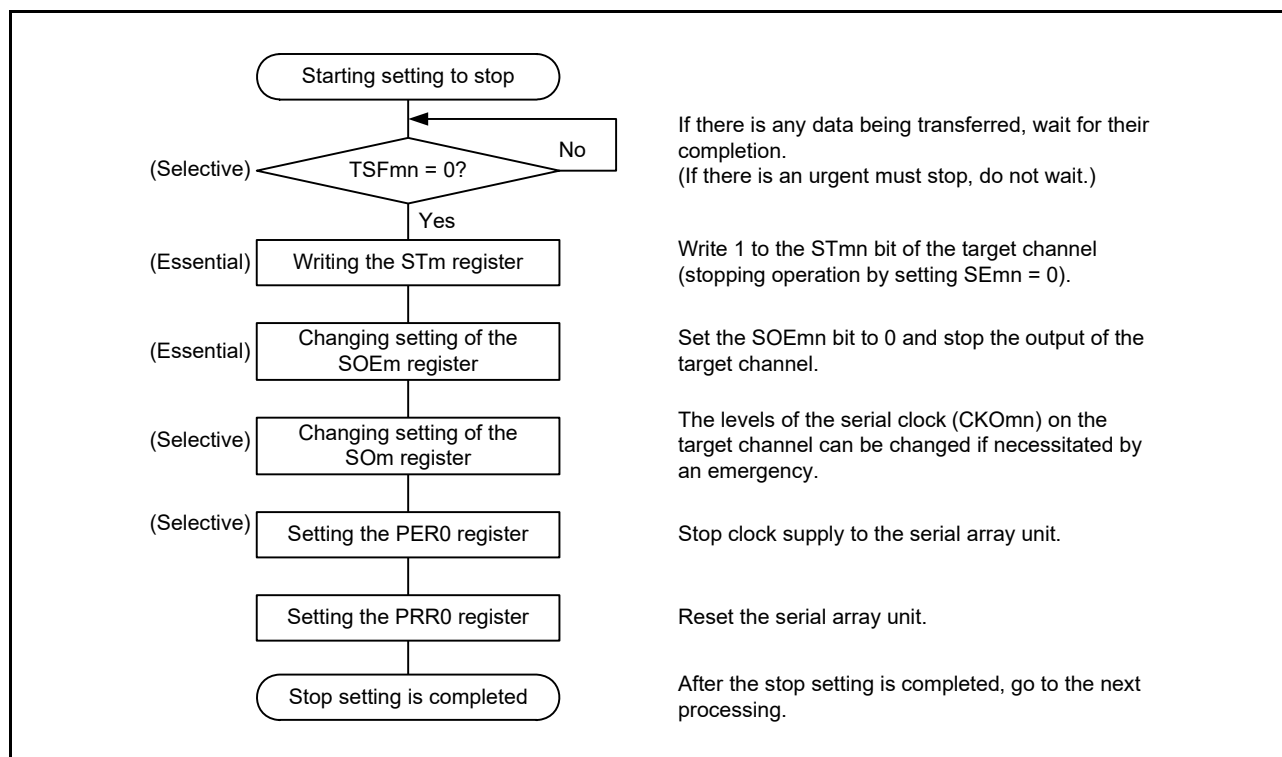
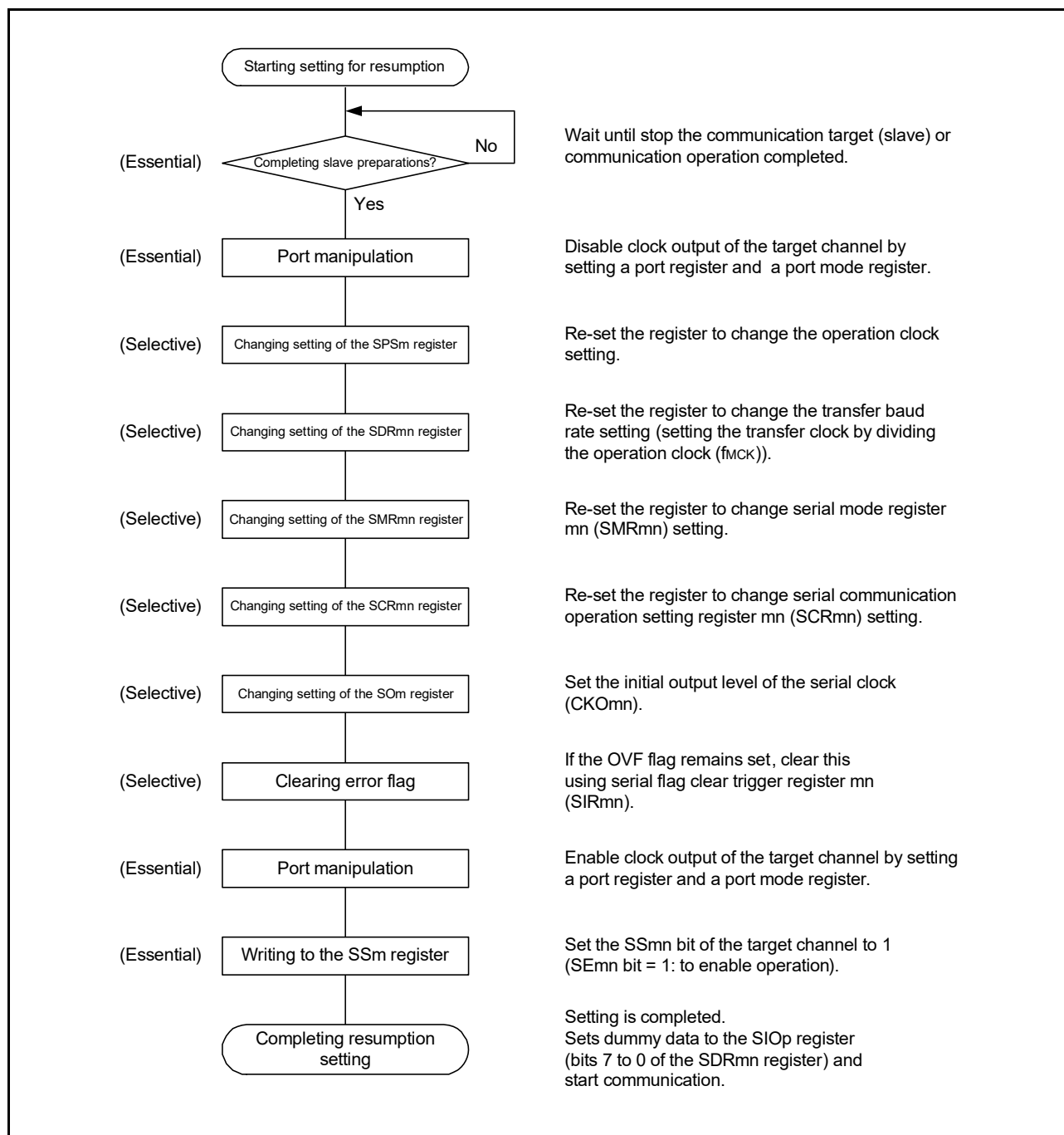


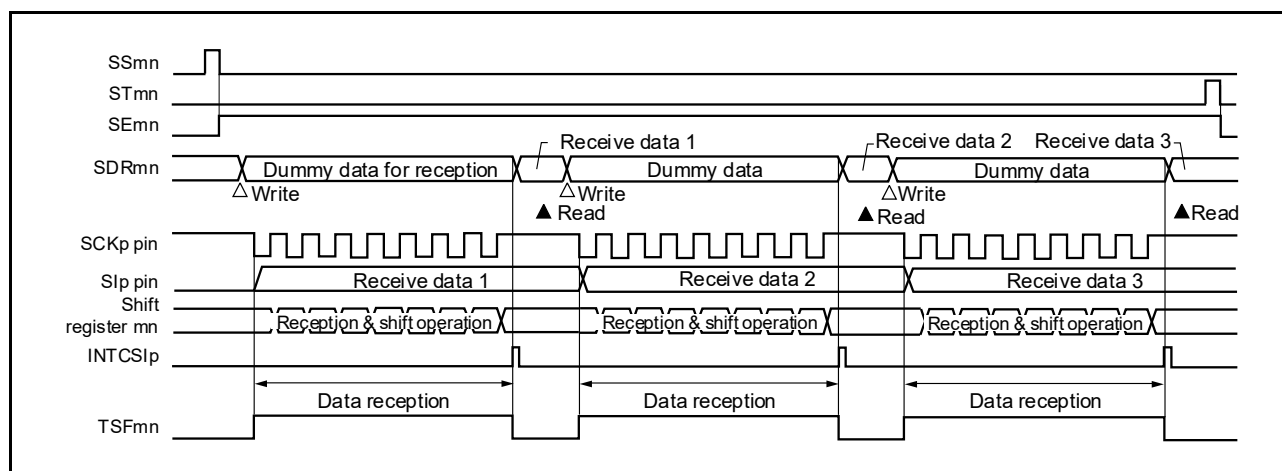
Figure 19 - 37 Procedure for Resuming Master Reception



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 with initialization instead of restarting the communication.

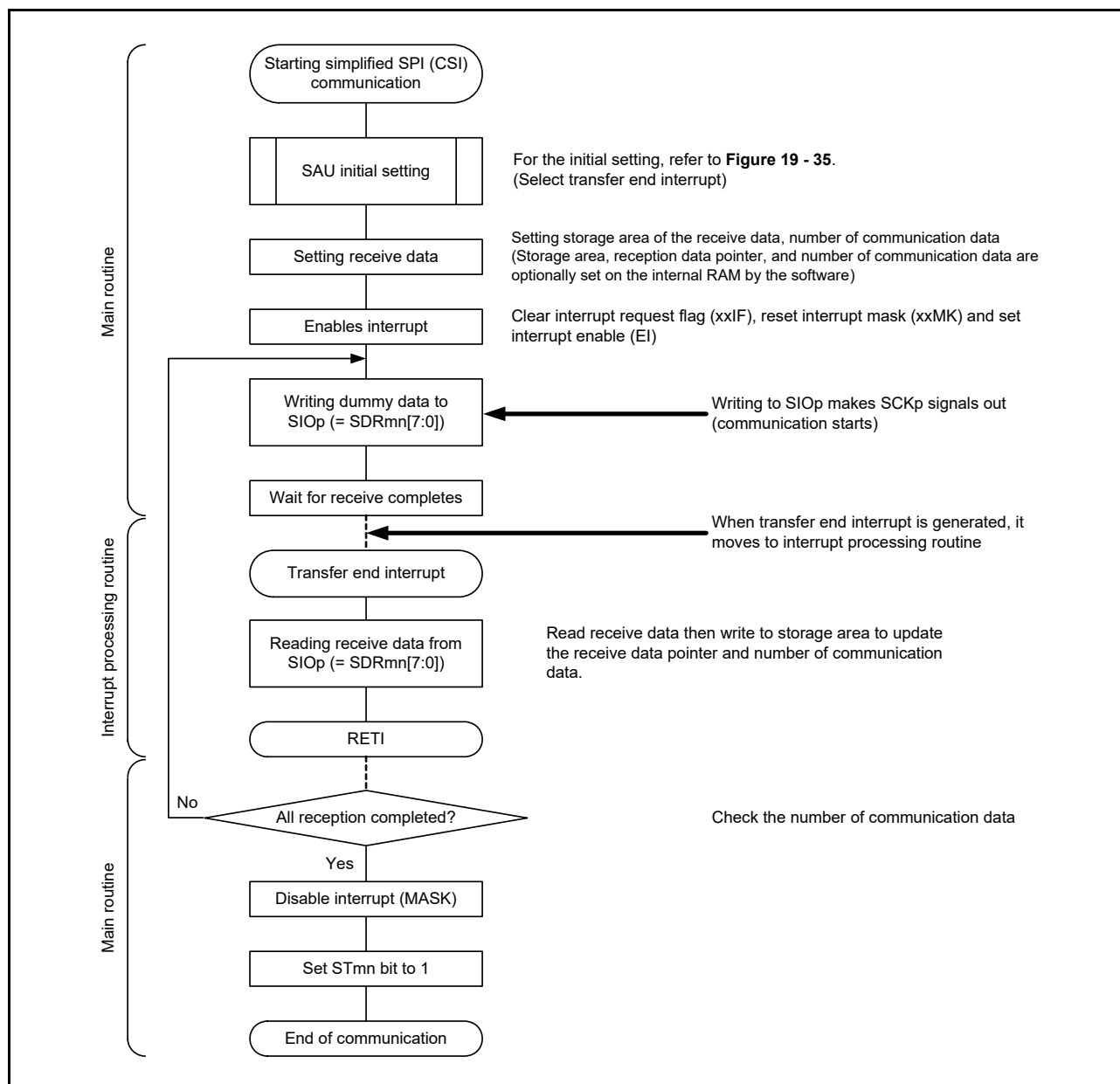
3. Processing flow (in single-reception mode)

Figure 19 - 38 Timing Chart of Master Reception (in Single-reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



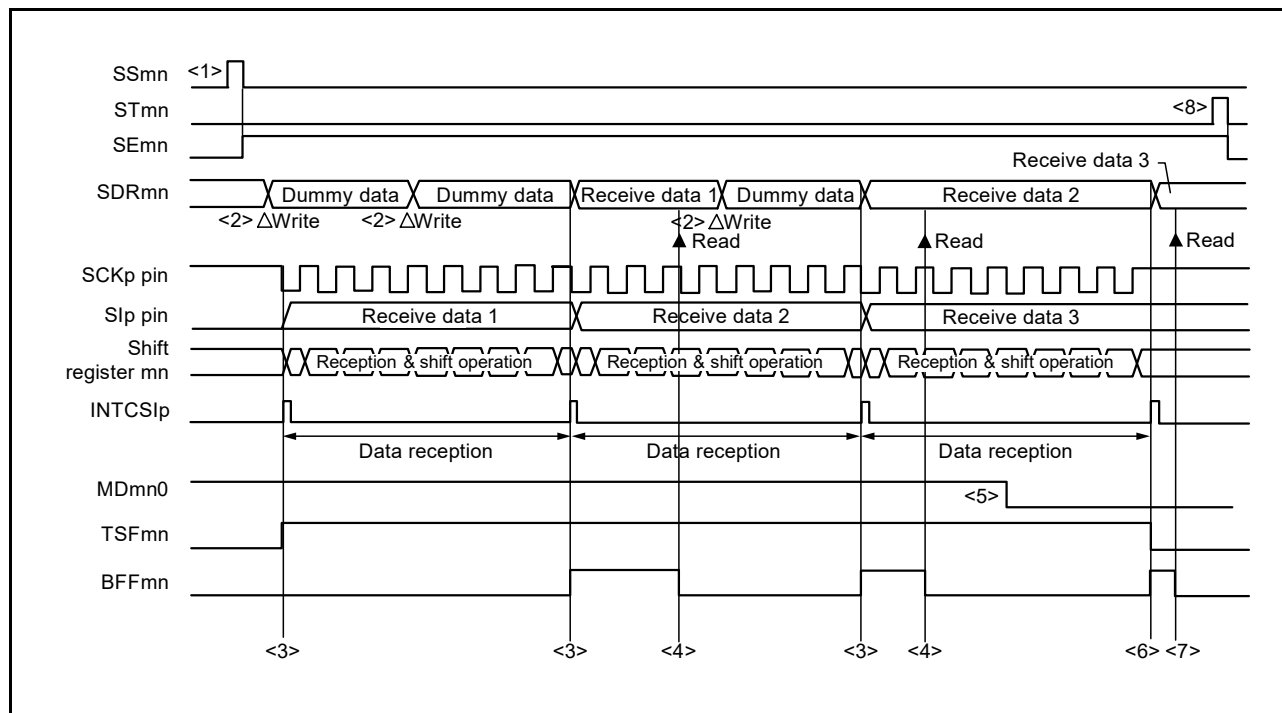
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 39 Flowchart of Master Reception (in Single-reception Mode)



4. Processing flow (in continuous reception mode)

Figure 19 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

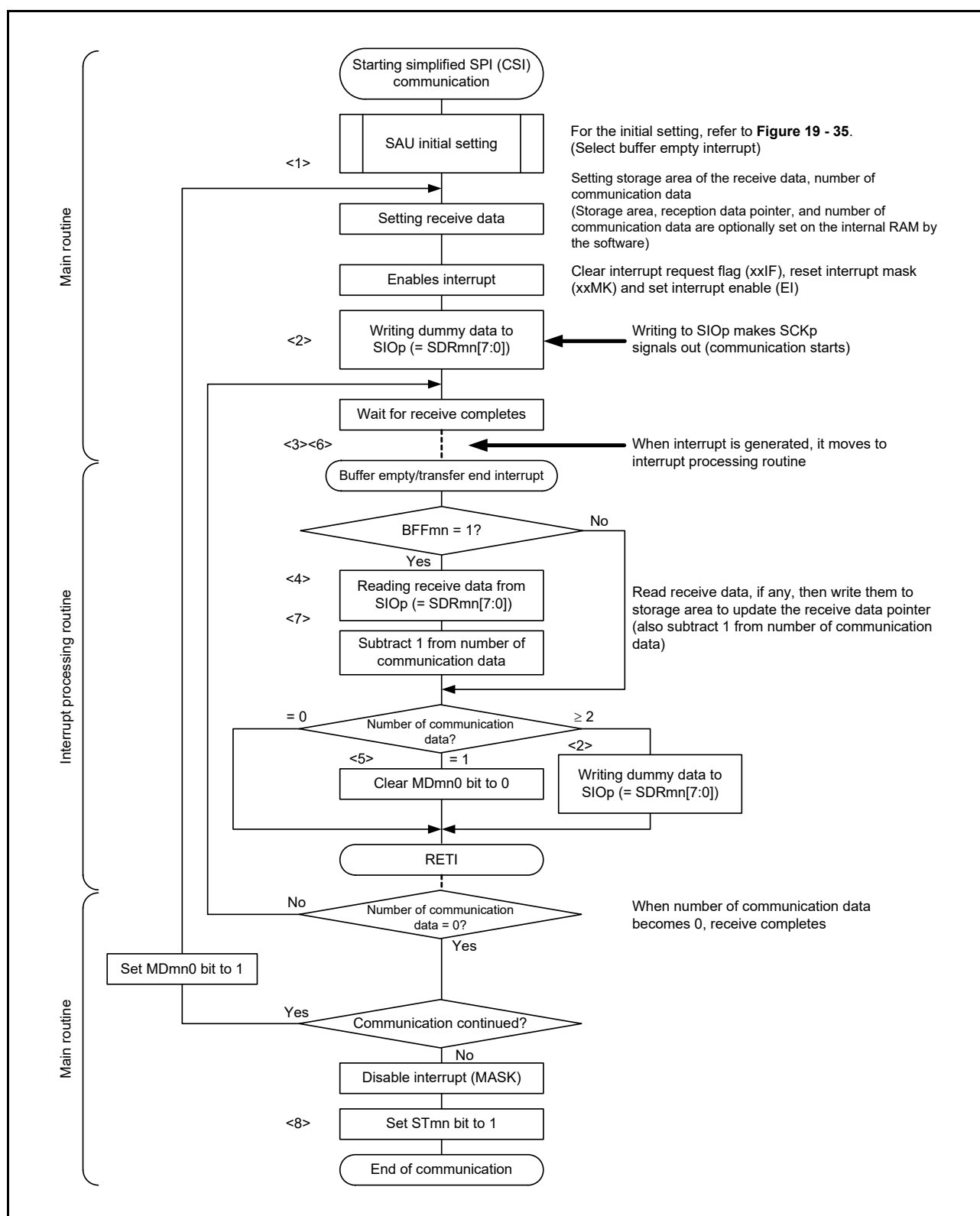


Caution The MDmn0 bit can be rewritten even during operation. However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 19 - 41 Flowchart of Master Reception (in Continuous Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 41 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.3 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	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21	SCK30, SI30, SO30	SCK31, SI31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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. fCLK/2 [Hz] (CSI00 only), fCLK/4 [Hz] Min. fCLK/(2 × 2 ¹⁵ × 128) [Hz] fCLK: 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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

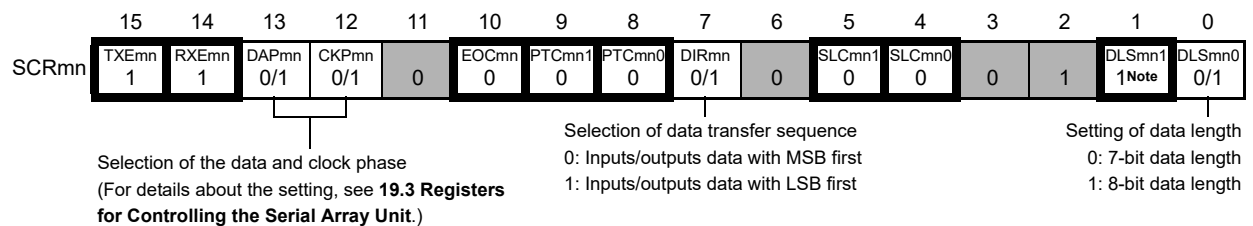
1. Register setting

Figure 19 - 42 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

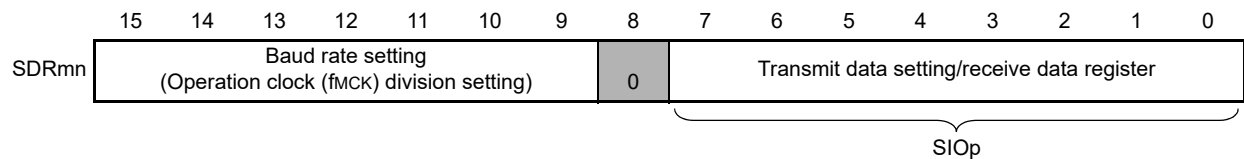
a) Serial mode register mn (SMRmn)



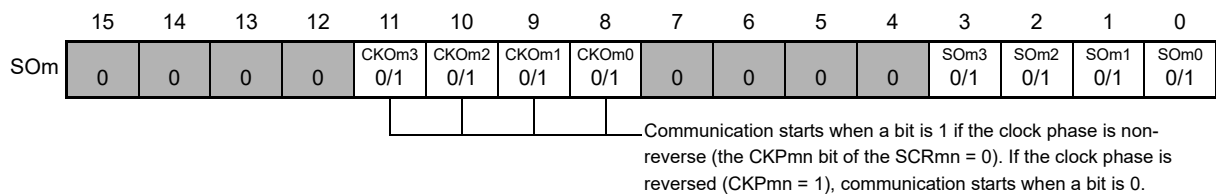
b) Serial communication operation setting register mn (SCRmn)



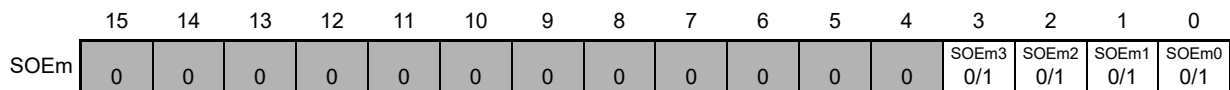
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



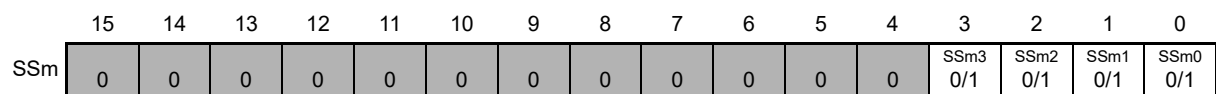
d) Serial output register m (SOM): Set only the bit of the target channel.



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.



Note This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

(Remarks continue on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

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 19 - 43 Initial Setting Procedure for Master Transmission/Reception

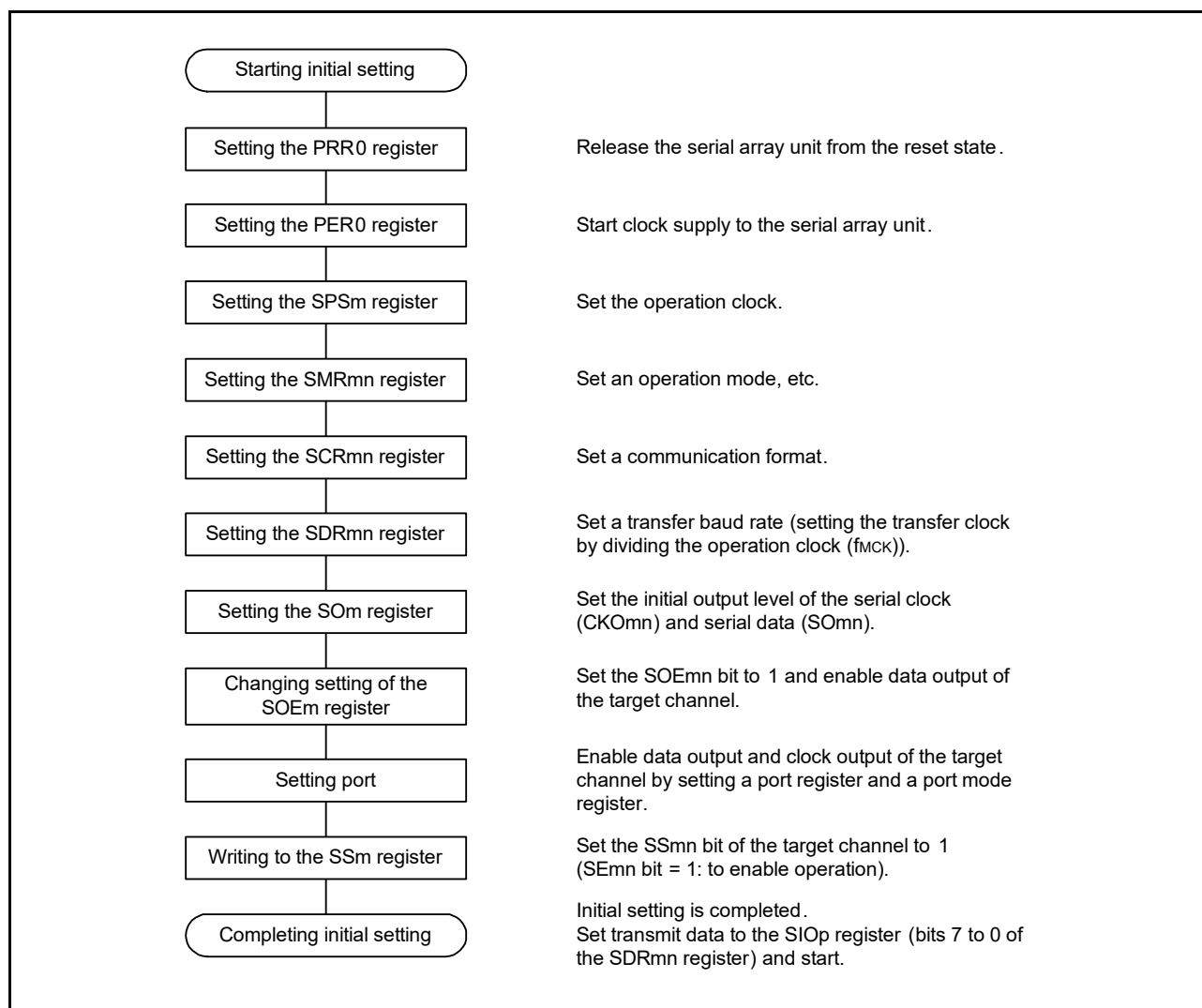


Figure 19 - 44 Procedure for Stopping Master Transmission/Reception

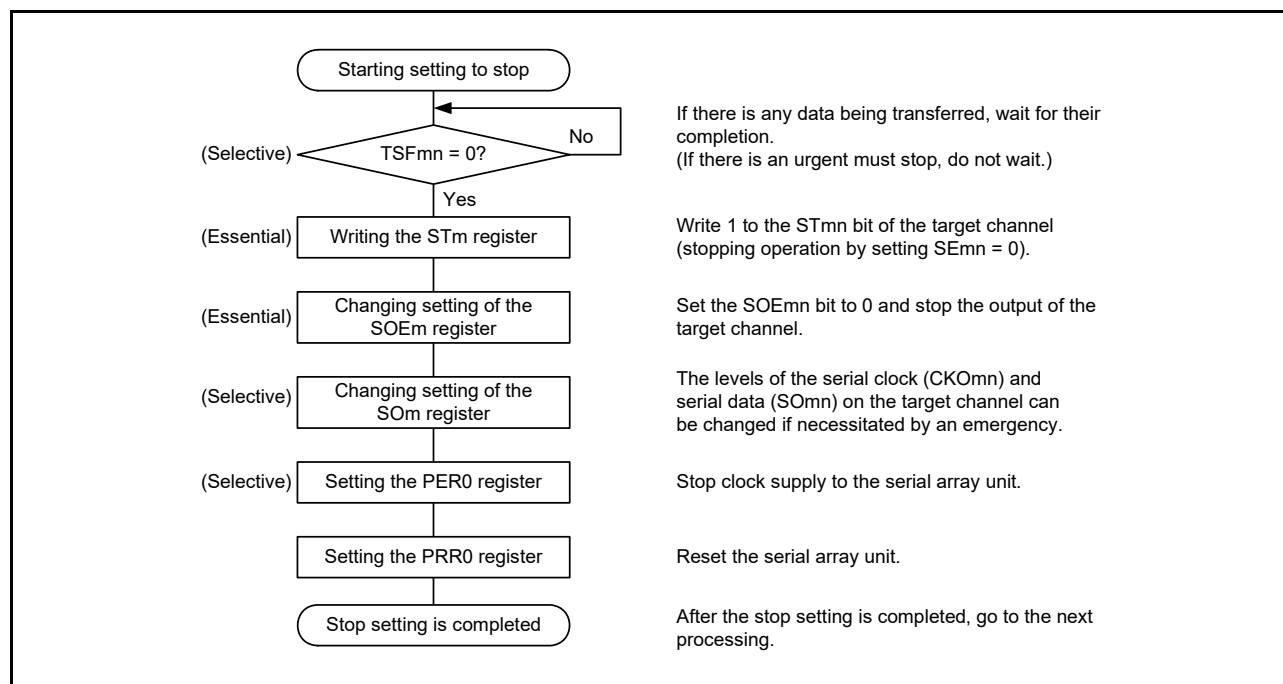
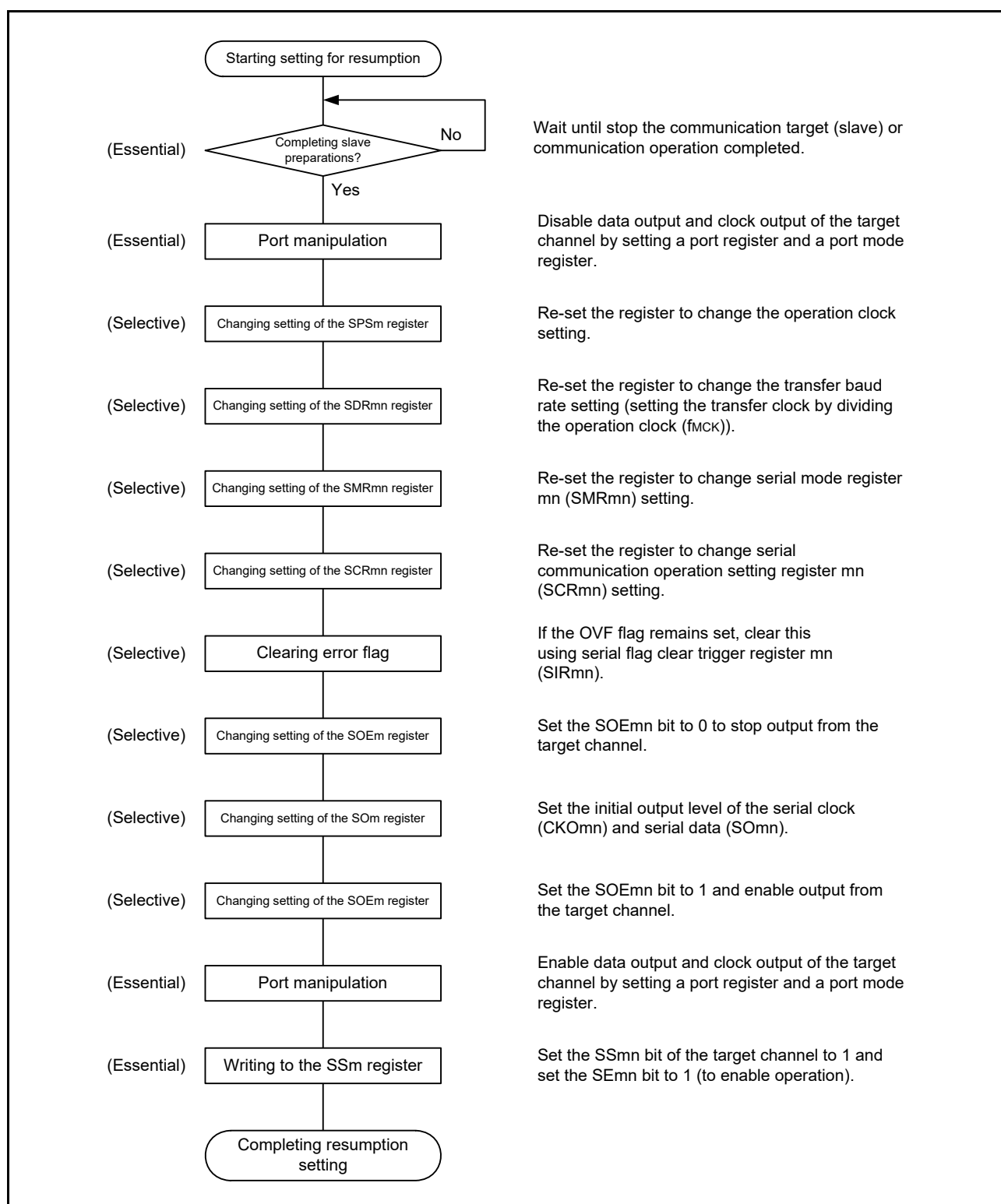
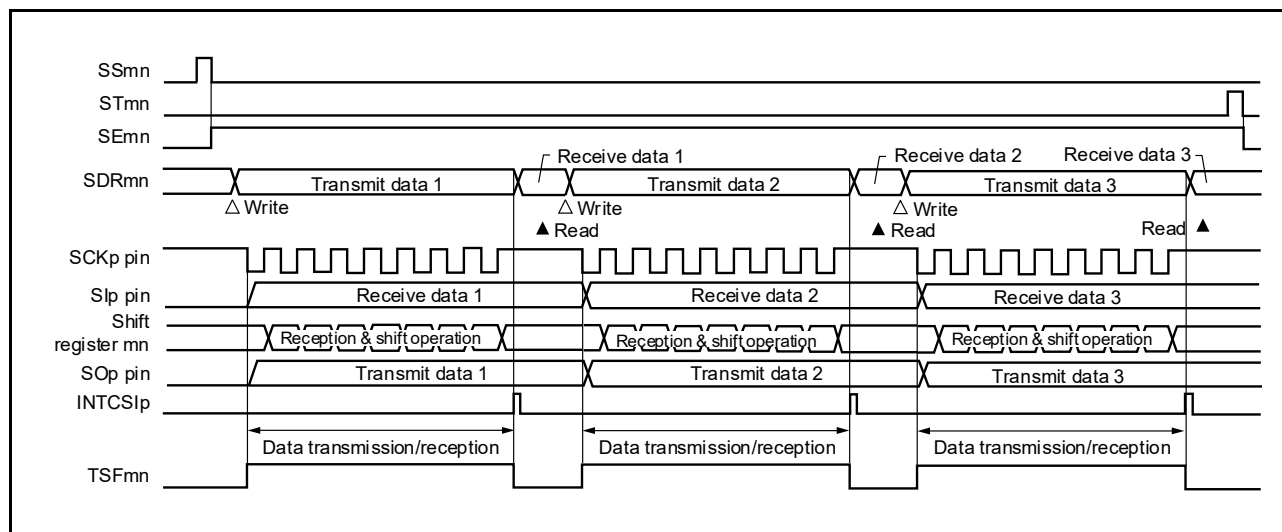


Figure 19 - 45 Procedure for Resuming Master Transmission/Reception



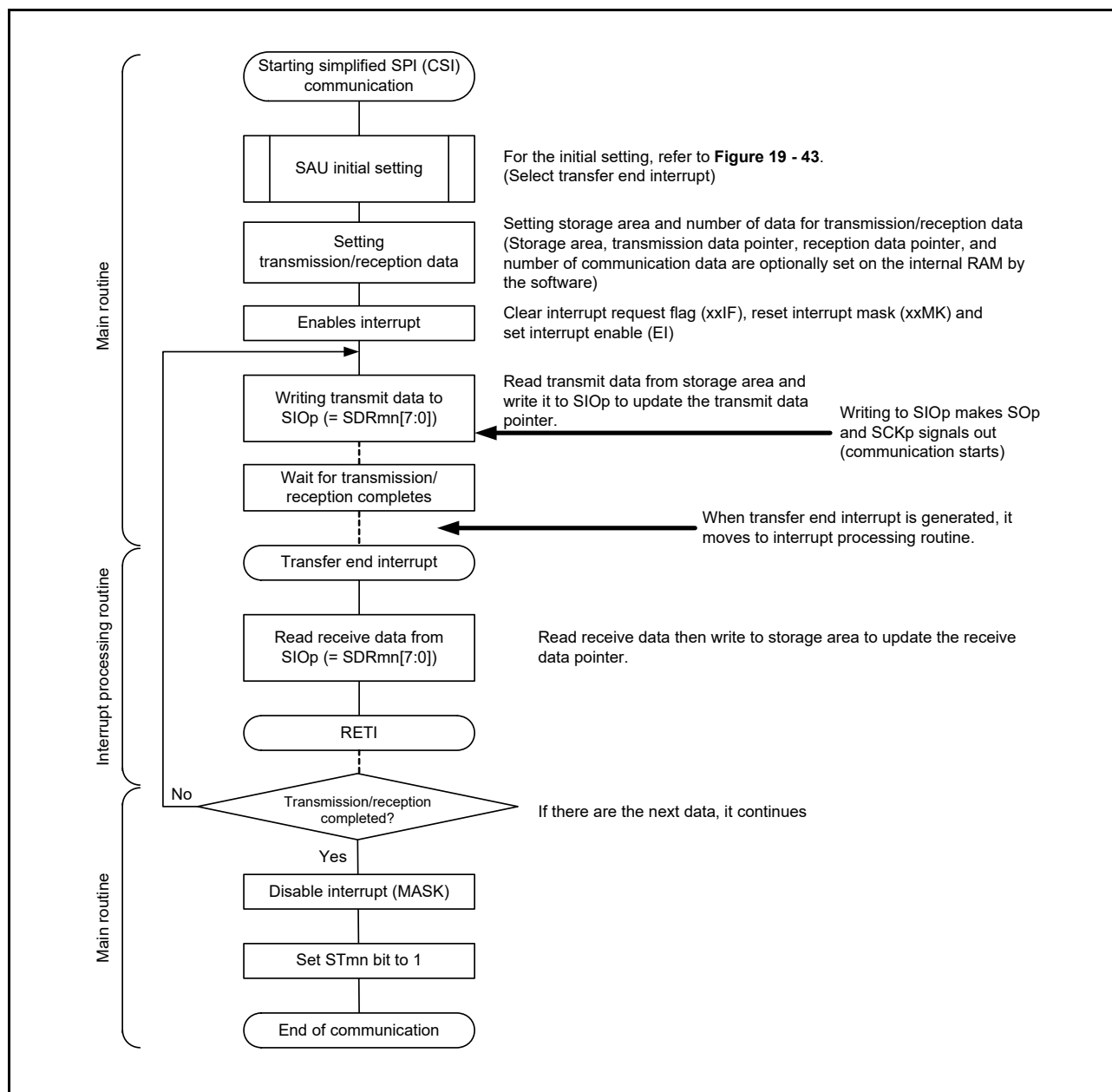
3. Processing flow (in single-transmission/reception mode)

Figure 19 - 46 Timing Chart of Master Transmission/Reception (in Single-transmission/reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



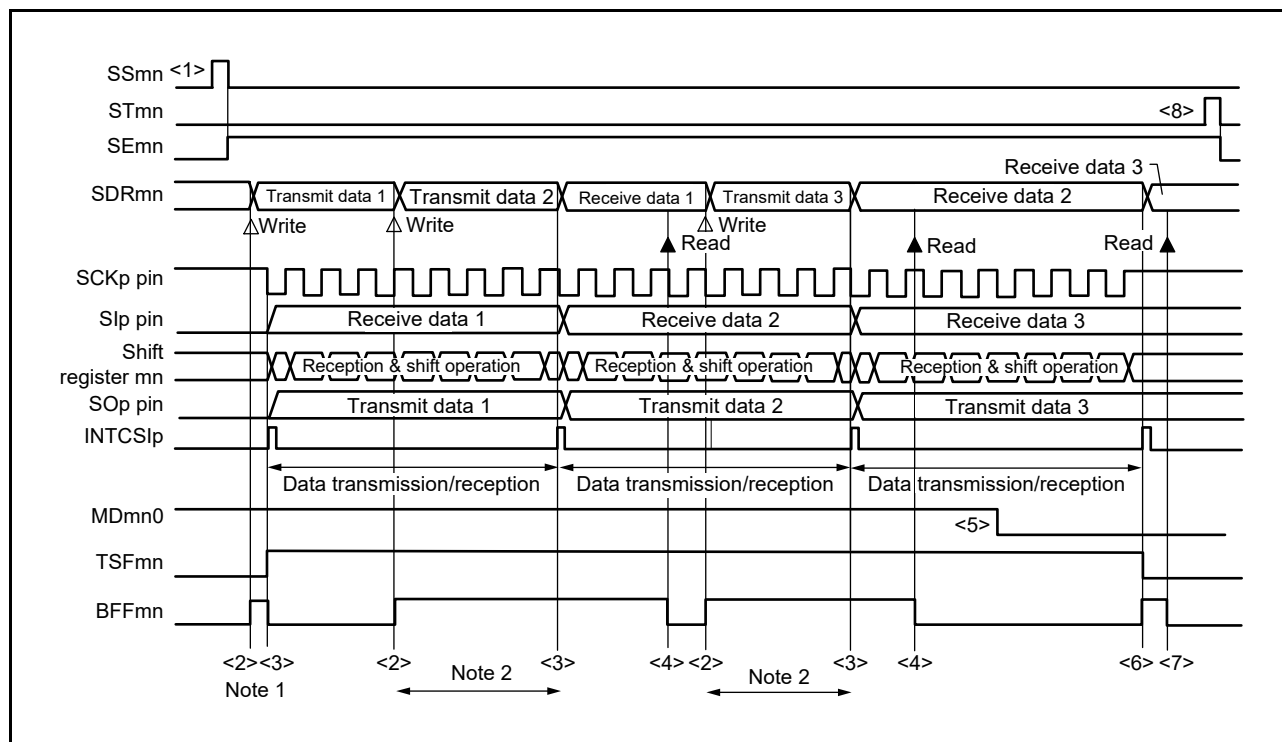
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

Figure 19 - 47 Flowchart of Master Transmission/Reception (in Single-transmission/reception Mode)



4. Processing flow (in continuous transmission/reception mode)

Figure 19 - 48 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.

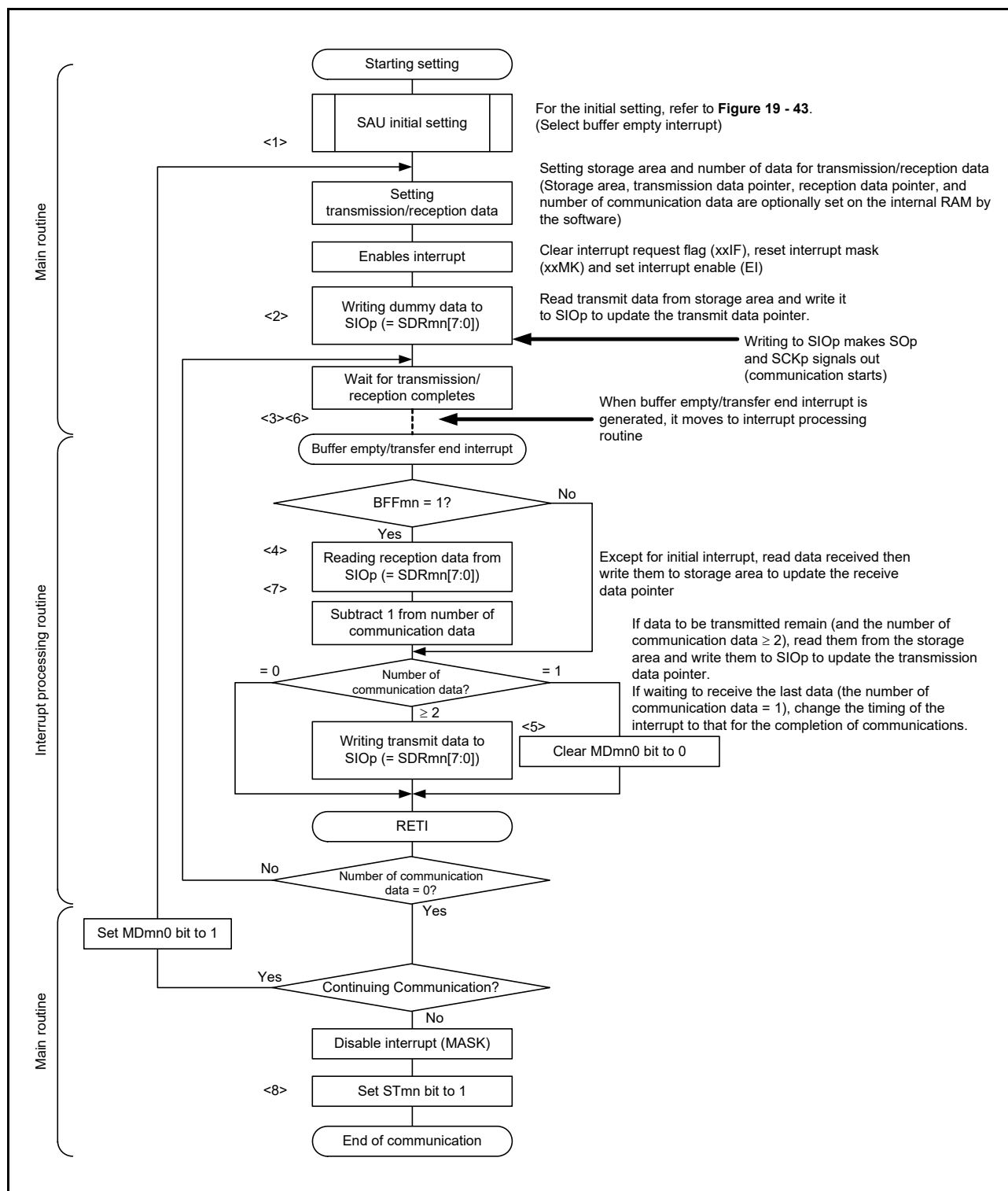
Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

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 19 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.4 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	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21	SCK30, SO30	SCK31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 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 42 Electrical Characteristics**.

Remark 1. f_{MCK} : Operation clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

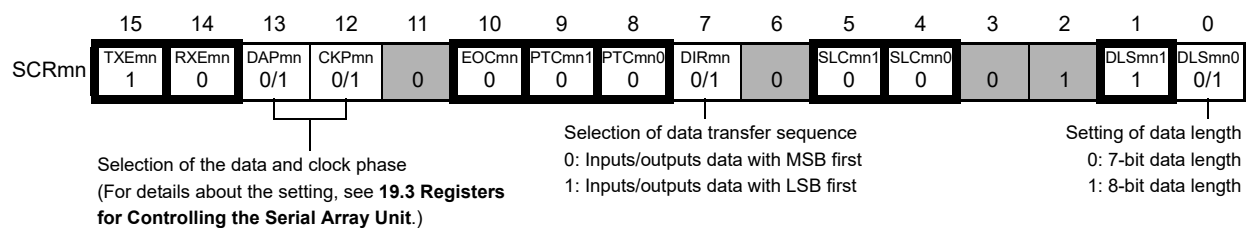
1. Register setting

Figure 19 - 50 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

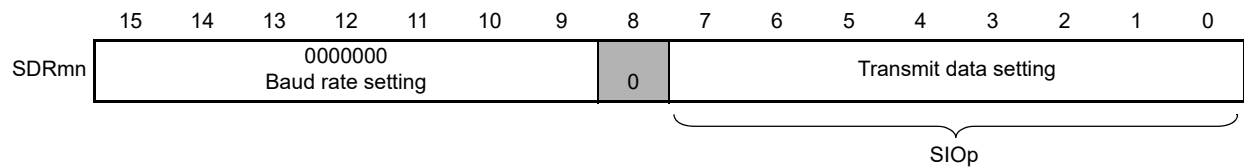
a) Serial mode register mn (SMRmn)



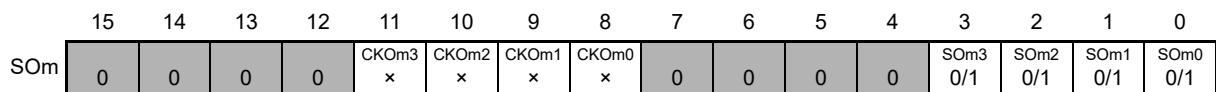
b) Serial communication operation setting register mn (SCRmn)



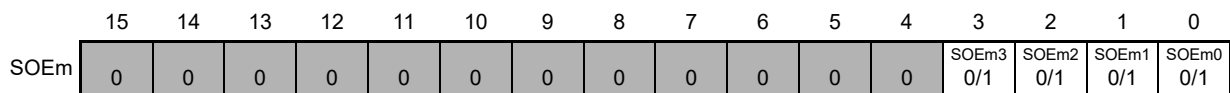
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



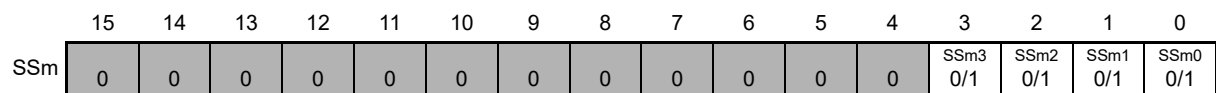
d) Serial output register m (SOM): Set only the bit of the target channel.



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.



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), 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)
x: 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 19 - 51 Initial Setting Procedure for Slave Transmission

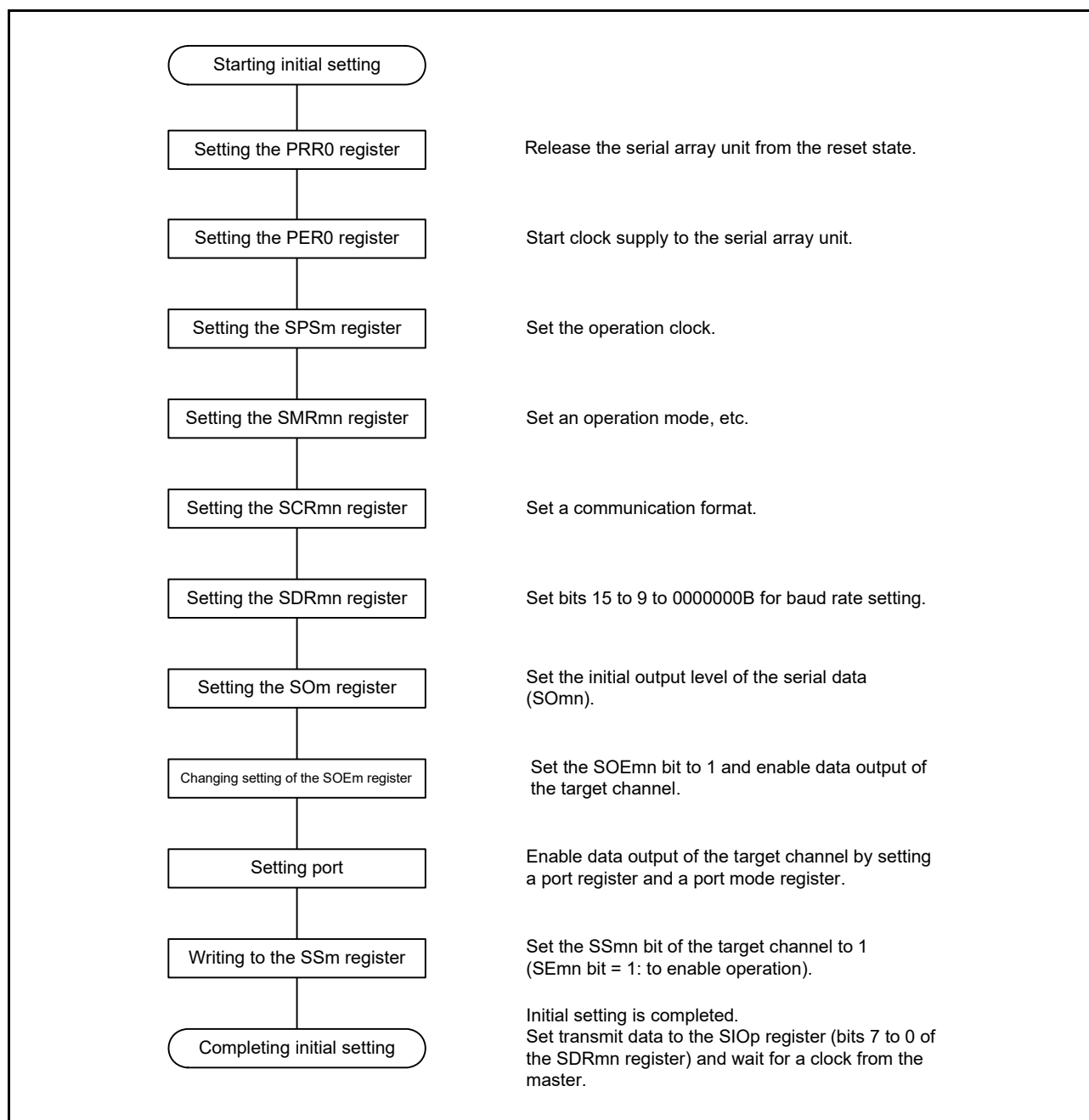


Figure 19 - 52 Procedure for Stopping Slave Transmission

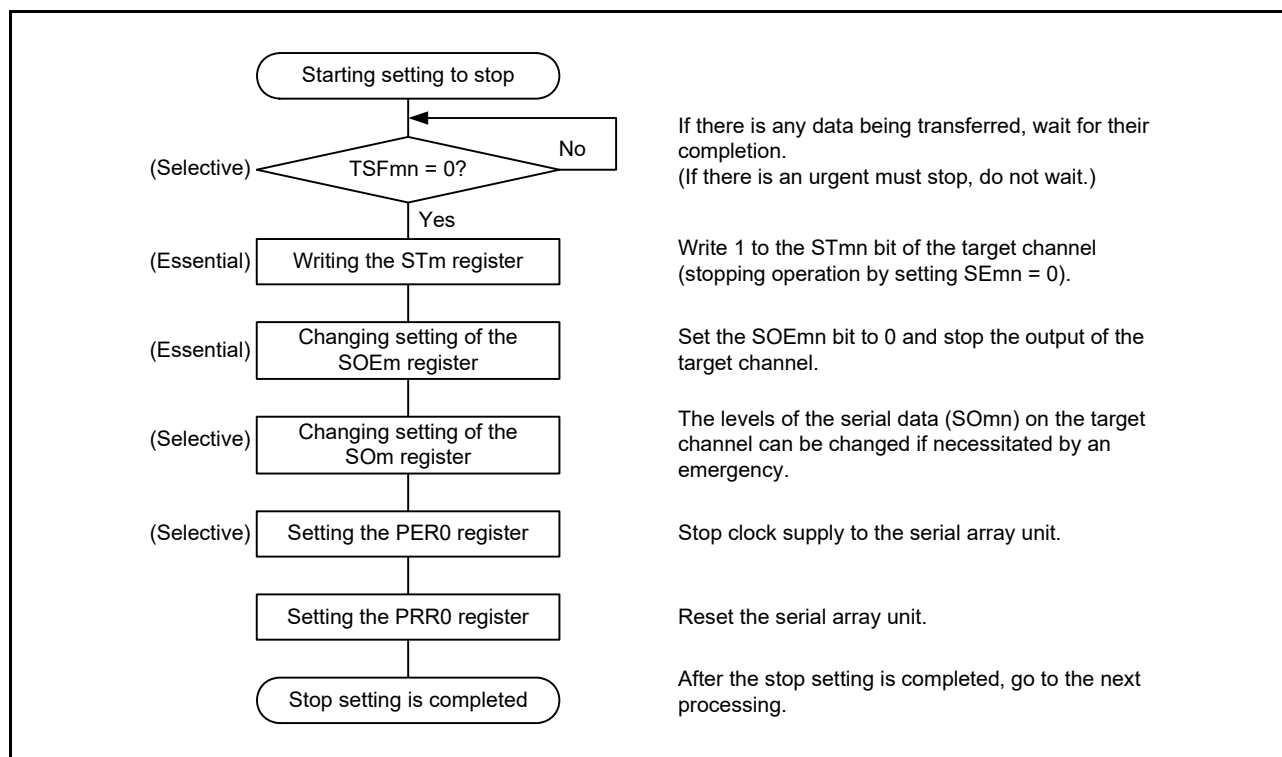
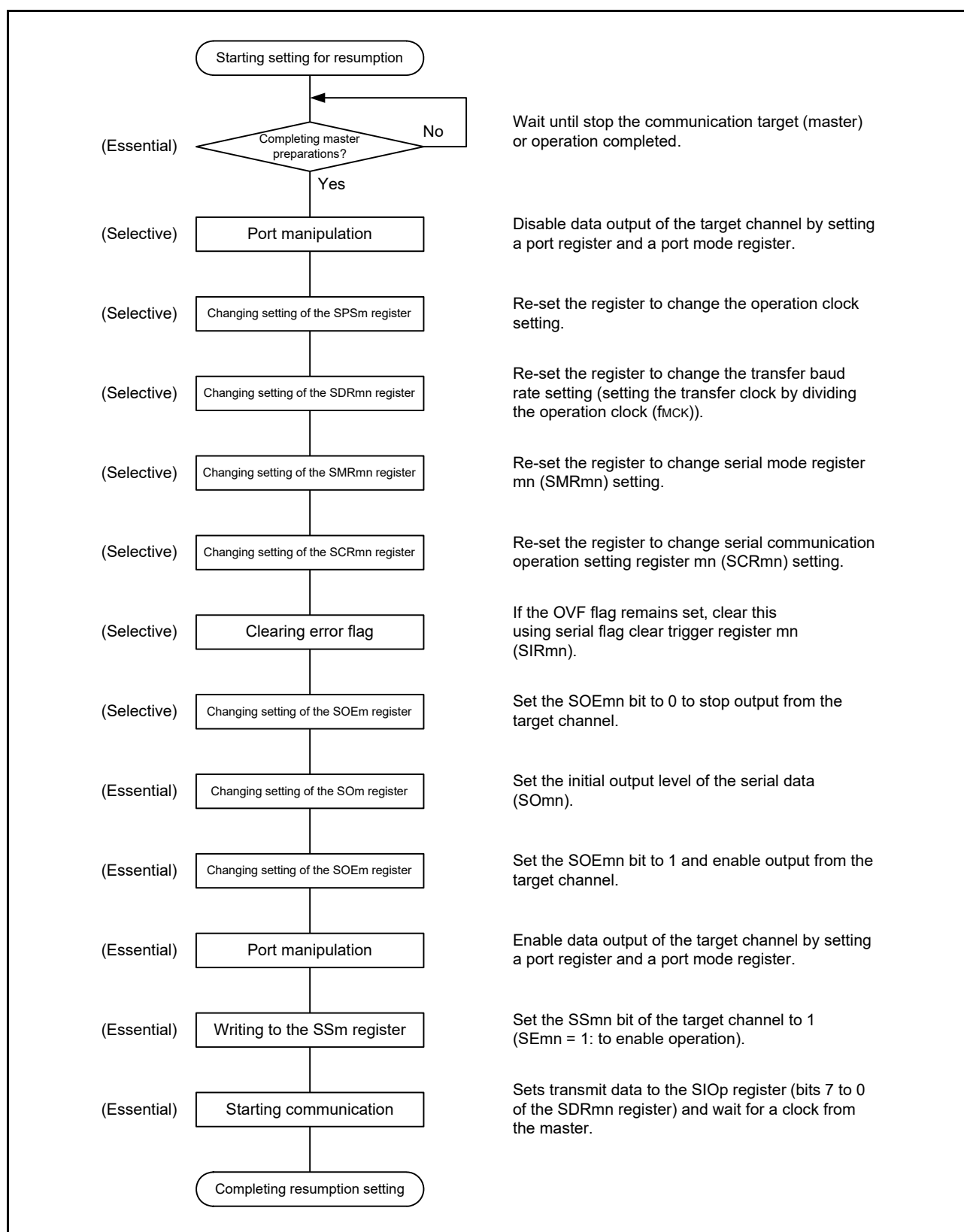


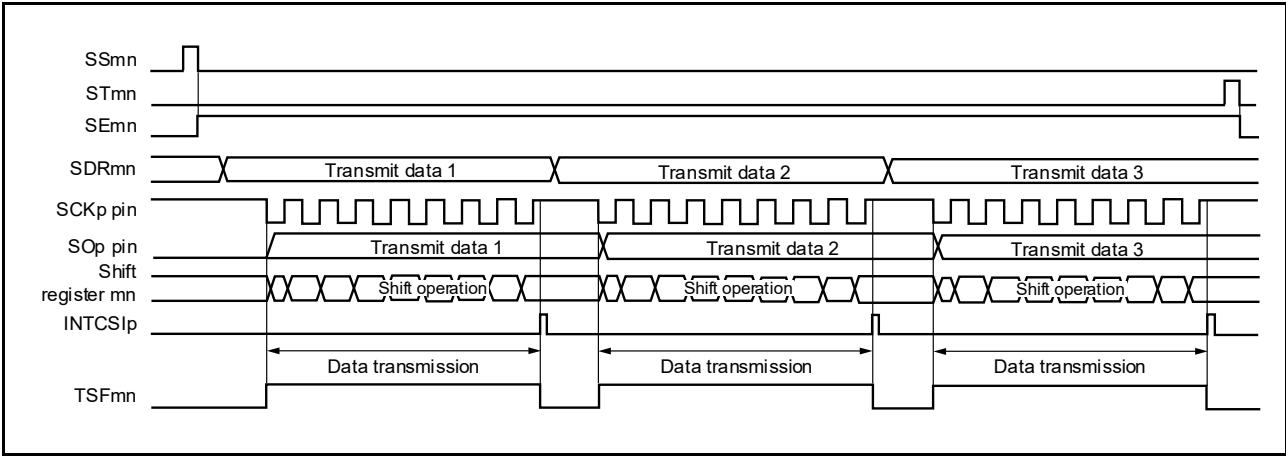
Figure 19 - 53 Procedure for Resuming Slave Transmission



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 with initialization instead of restarting the communication.

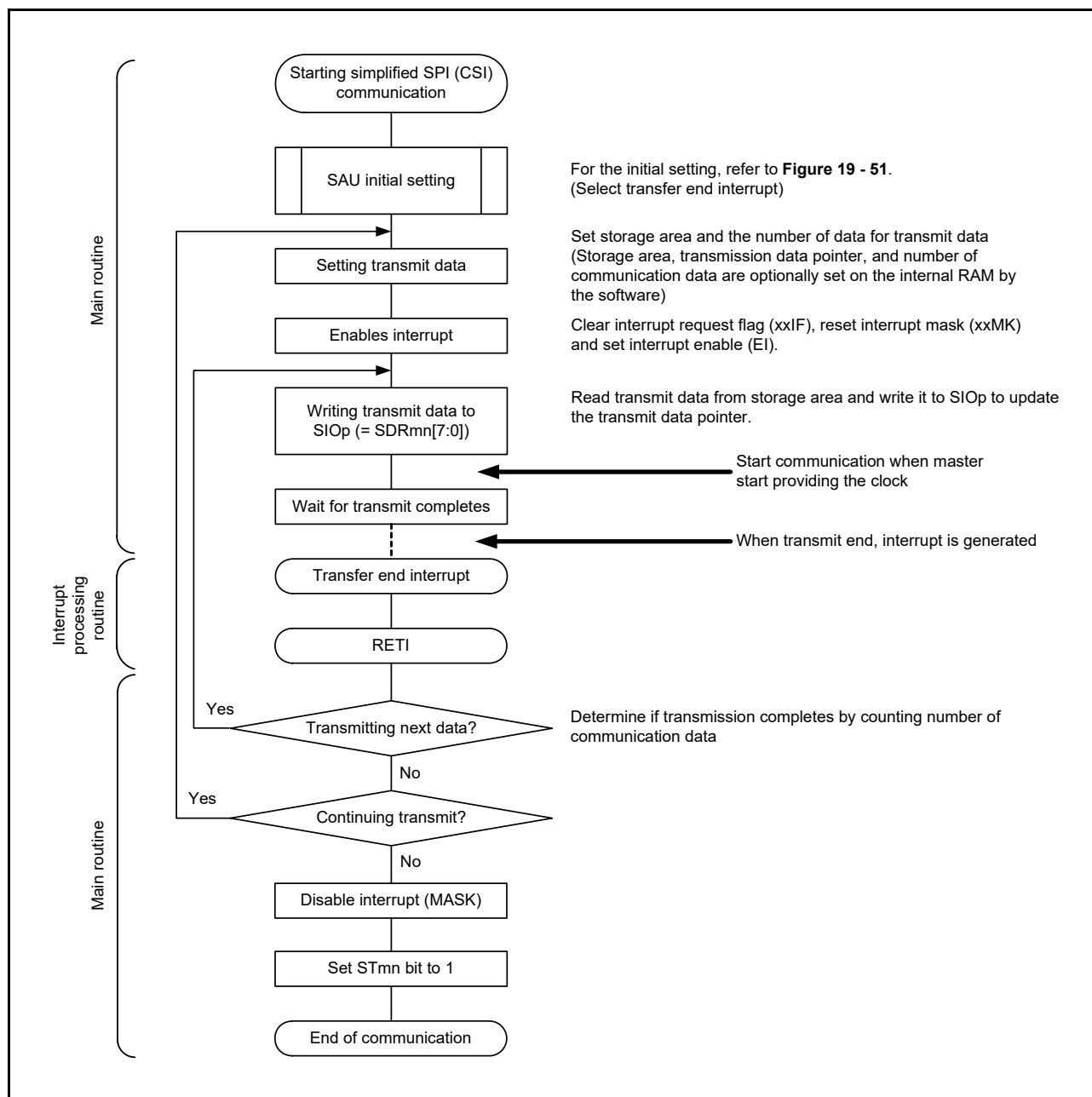
3. Processing flow (in single-transmission mode)

Figure 19 - 54 Timing Chart of Slave Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



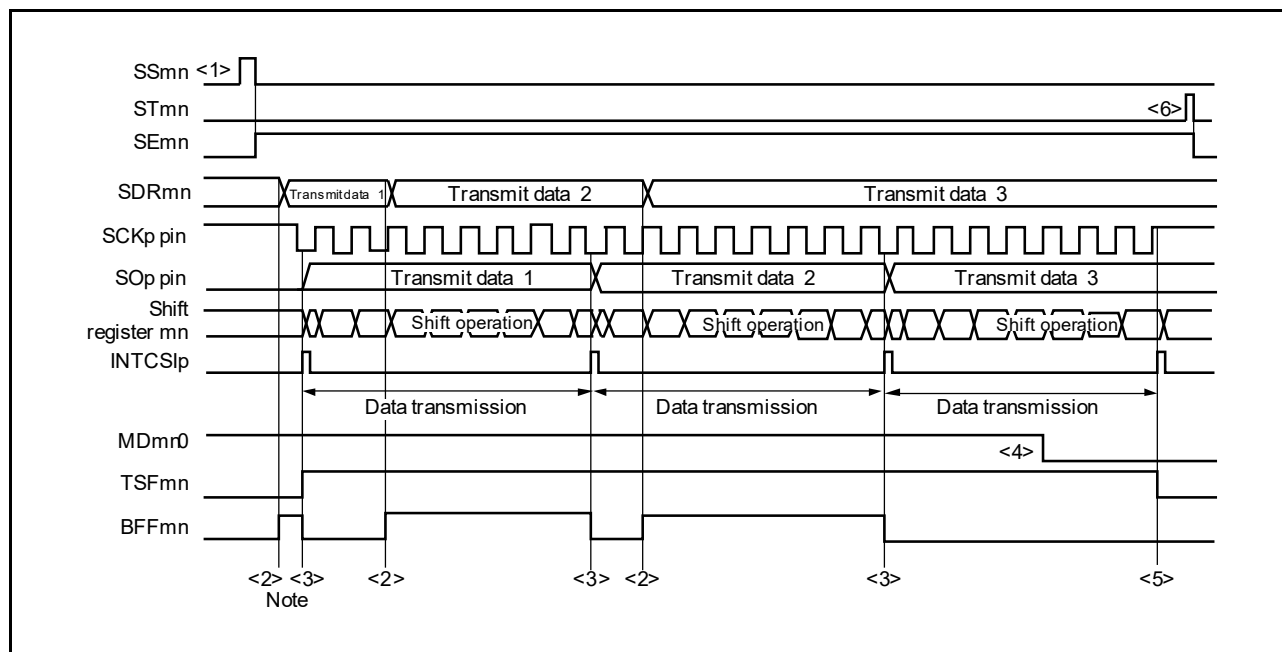
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 55 Flowchart of Slave Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 19 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

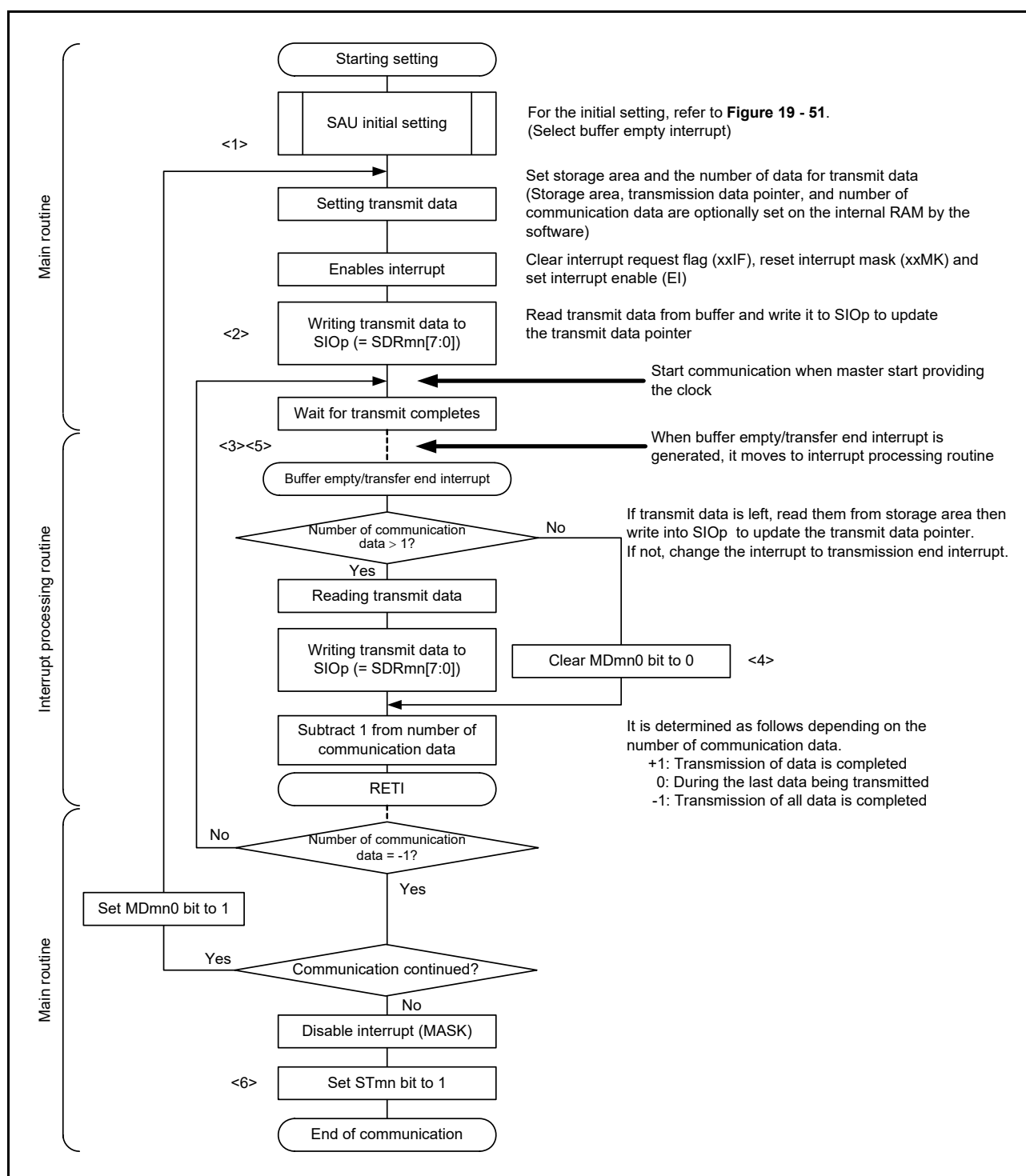


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 57 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 19 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

19.5.5 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	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21	SCK30, SI30	SCK31, SI31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 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 42 Electrical Characteristics**.

Remark 1. f_{MCK} : Operation clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

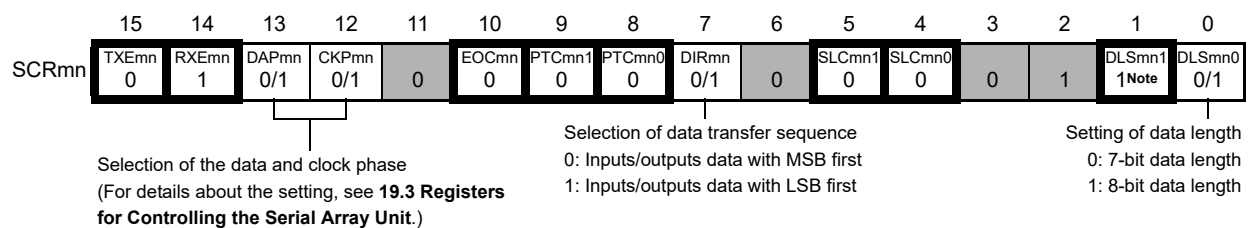
1. Register setting

Figure 19 - 58 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

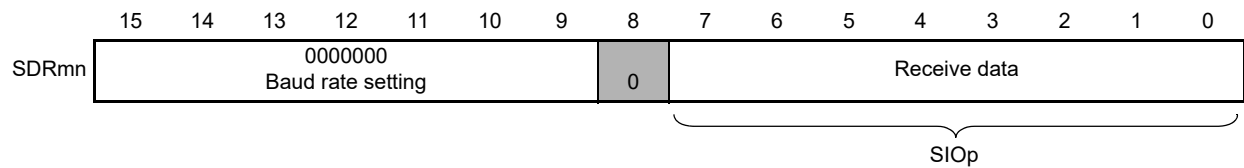
a) Serial mode register mn (SMRmn)



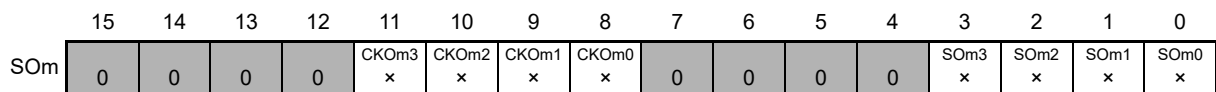
b) Serial communication operation setting register mn (SCRmn)



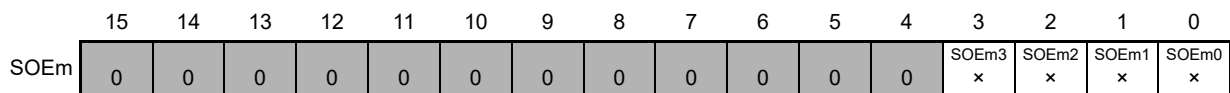
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



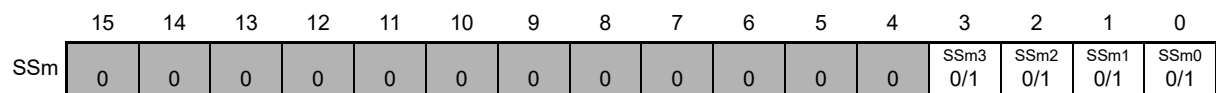
d) Serial output register m (SOM): This register is not used in this mode.



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.



Note This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. : Setting is fixed in the simplified SPI (CSI) slave reception mode, : Setting disabled (set to the initial value)
 x: 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 19 - 59 Initial Setting Procedure for Slave Reception

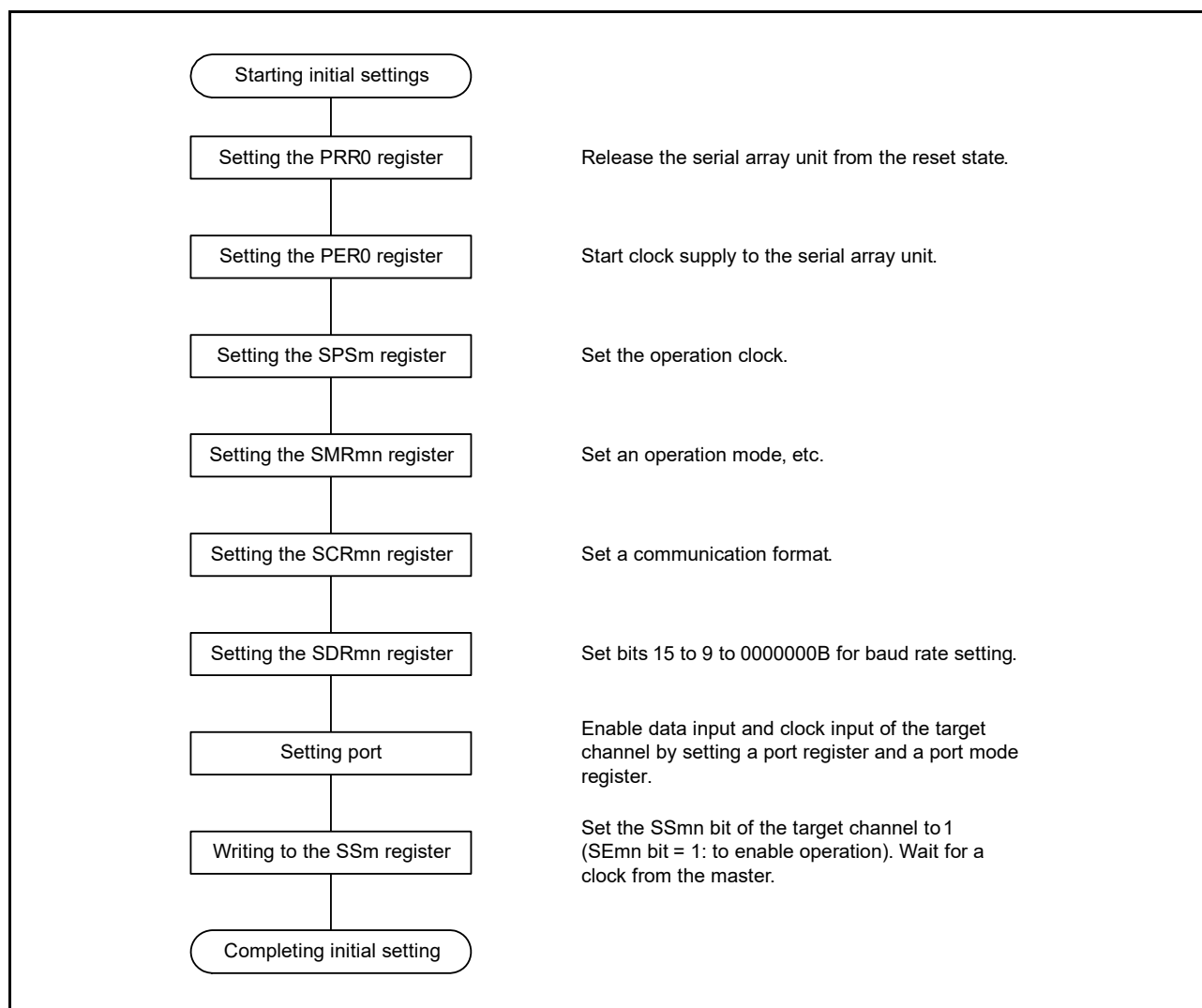


Figure 19 - 60 Procedure for Stopping Slave Reception

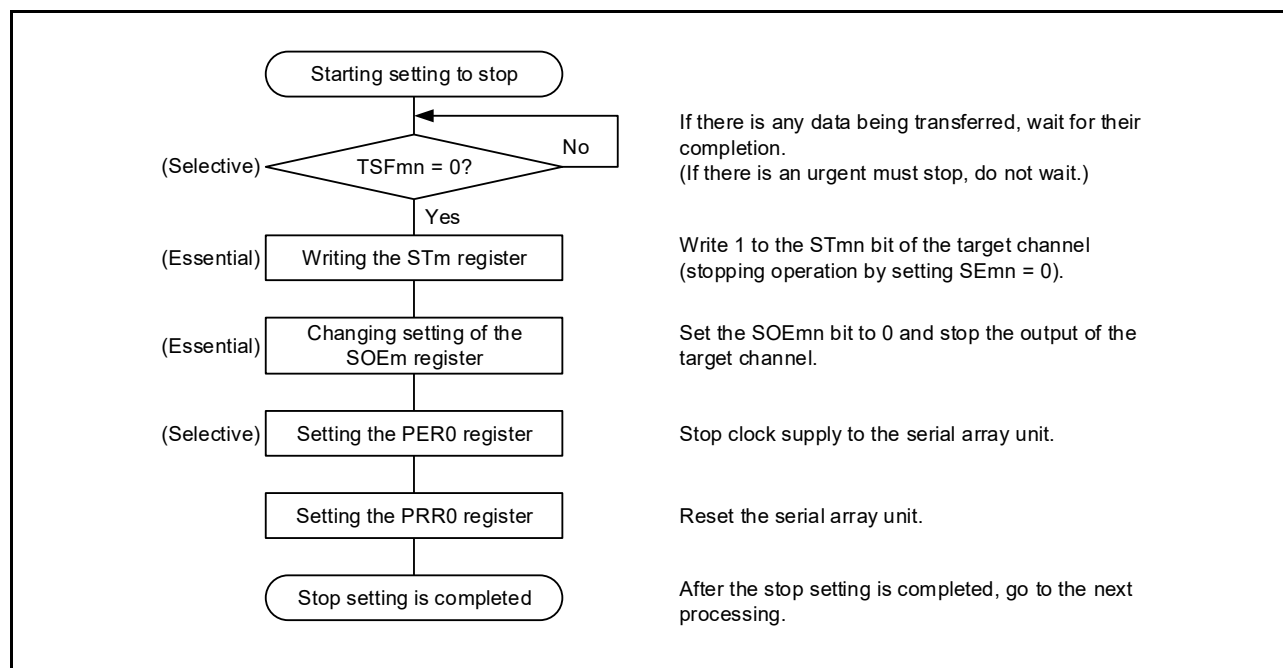
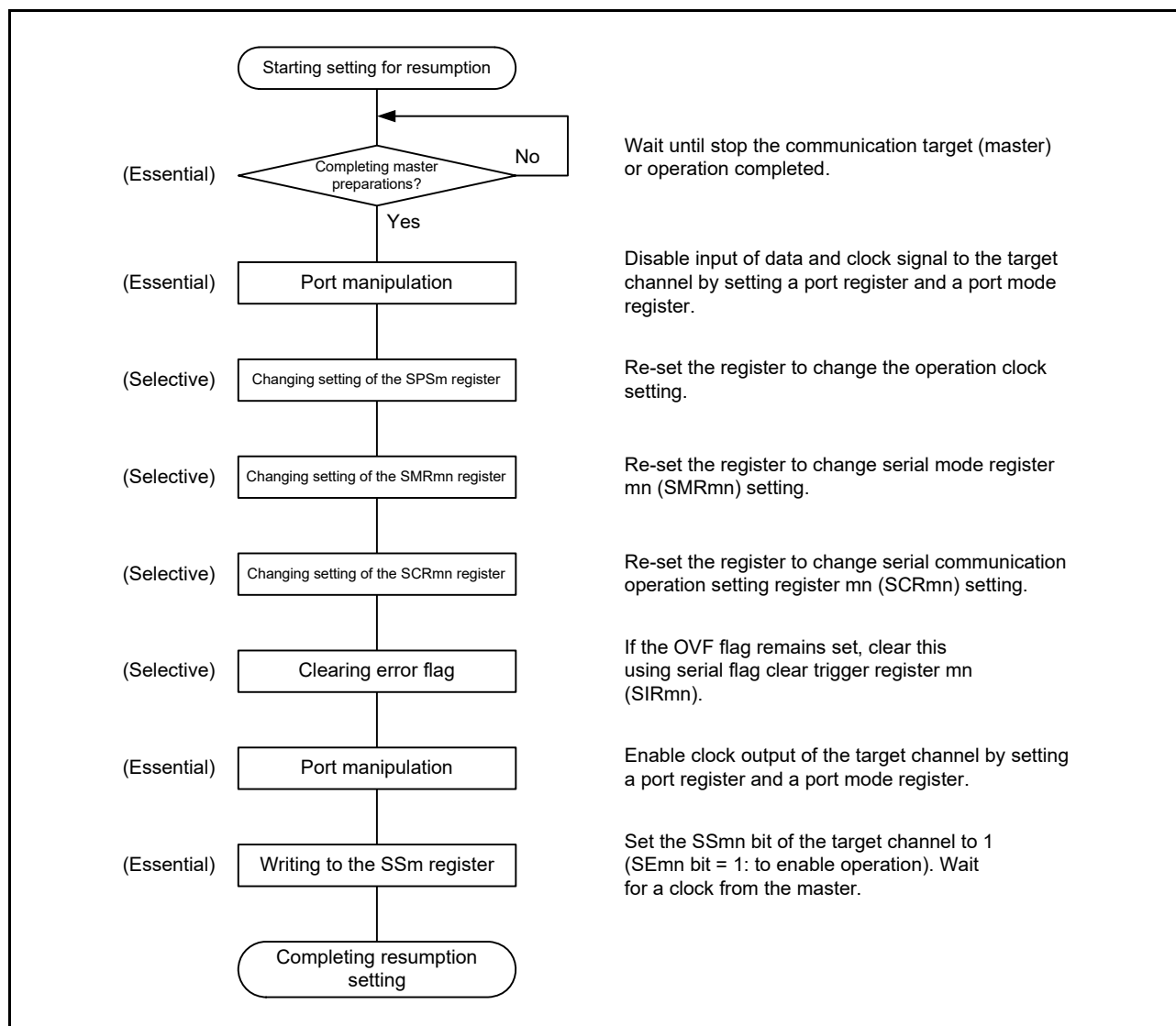


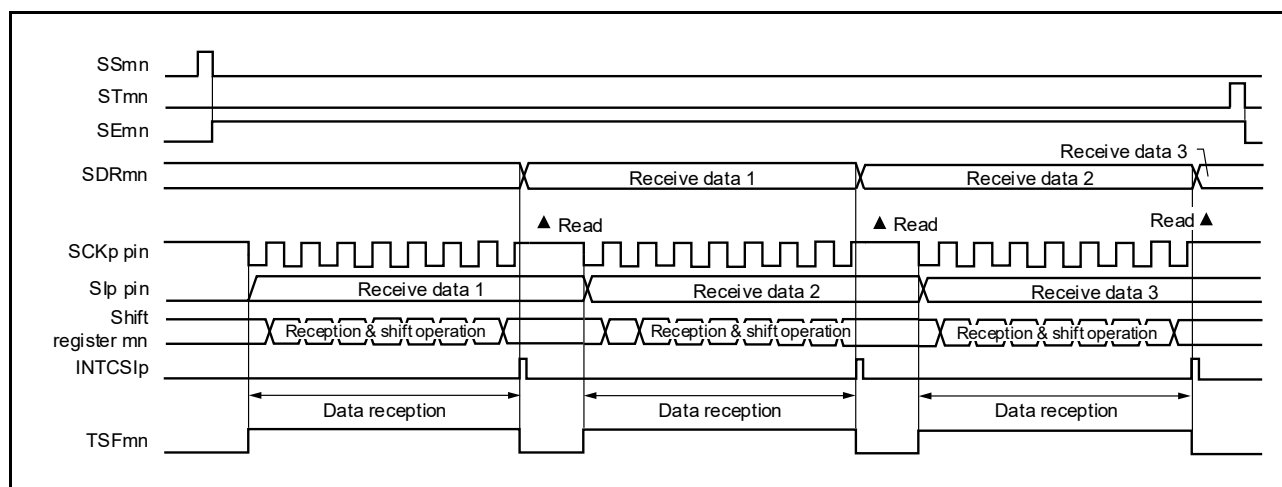
Figure 19 - 61 Procedure for Resuming Slave Reception



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 with initialization instead of restarting the communication.

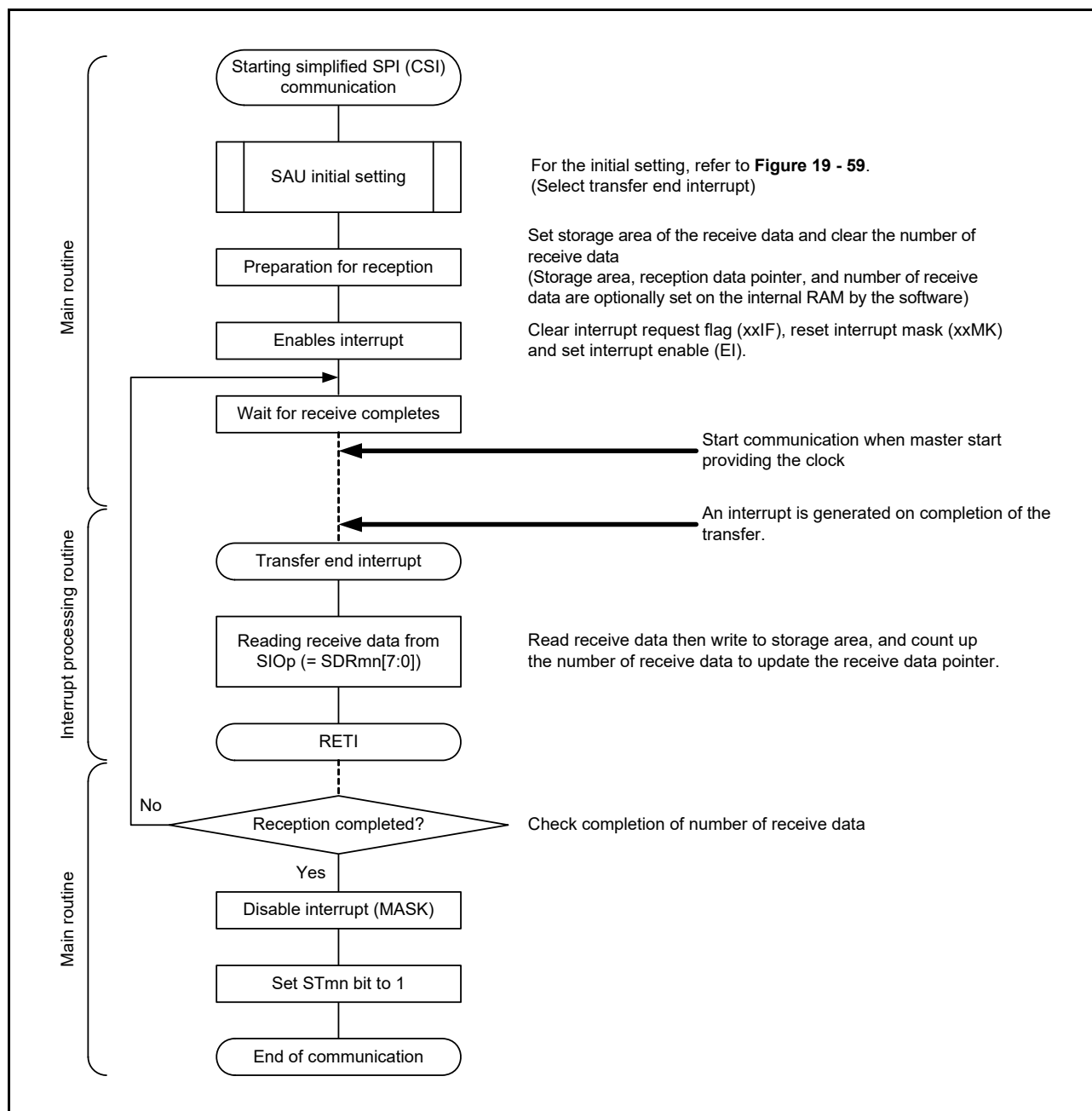
3. Processing flow (in single-reception mode)

Figure 19 - 62 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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 63 Flowchart of Slave Reception (in Single-reception Mode)



19.5.6 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	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21	SCK30, SI30, SO30	SCK31, SI31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 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 42 Electrical Characteristics**.

Remark 1. f_{MCK} : Operation clock frequency of target channel
 f_{SCK} : Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

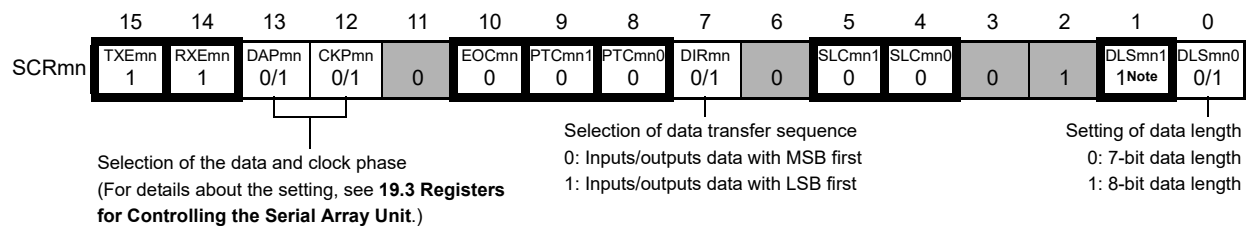
1. Register setting

Figure 19 - 64 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

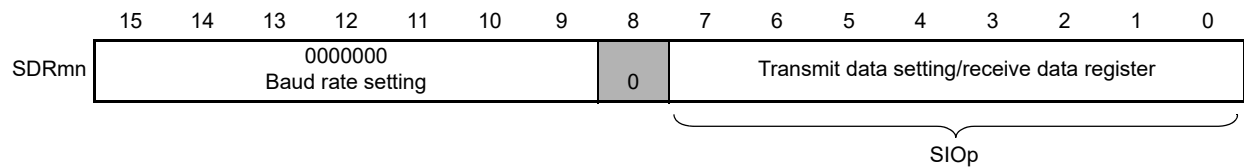
a) Serial mode register mn (SMRmn)



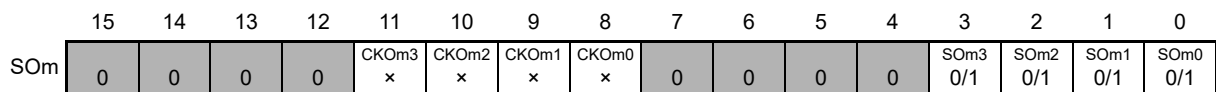
b) Serial communication operation setting register mn (SCRmn)



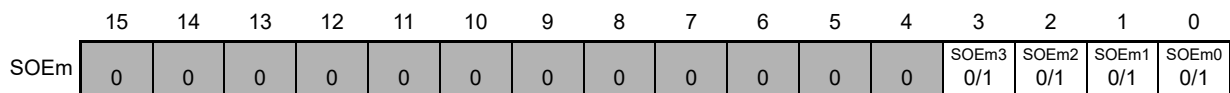
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



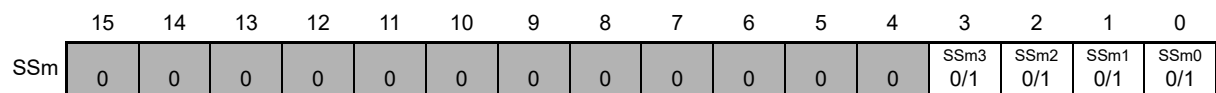
d) Serial output register m (SOM): Set only the bit of the target channel.



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.



Note This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(Remarks continue on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

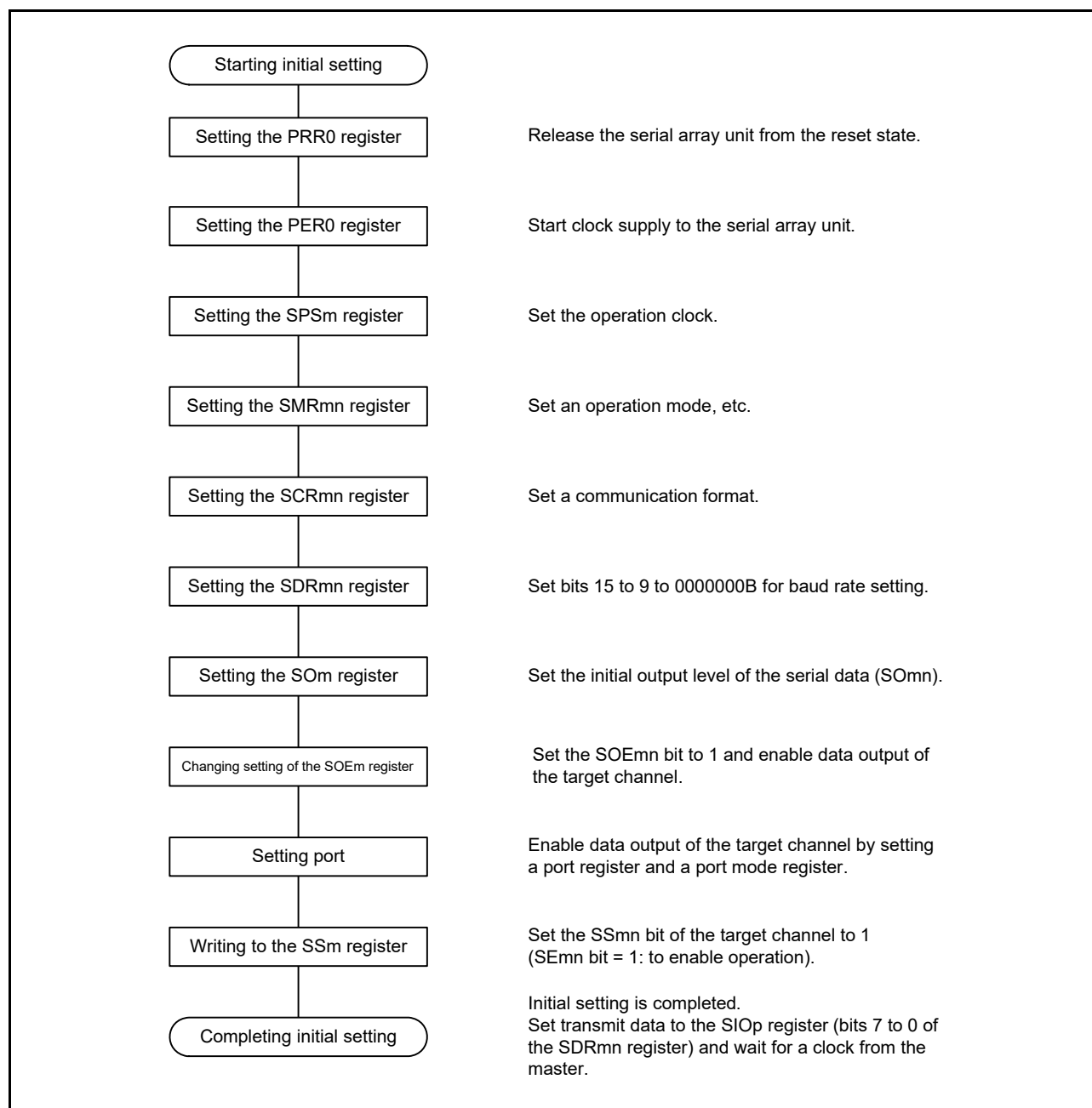
Remark 2. : Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode, : Setting disabled (set to the initial value)

×: 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 19 - 65 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 19 - 66 Procedure for Stopping Slave Transmission/Reception

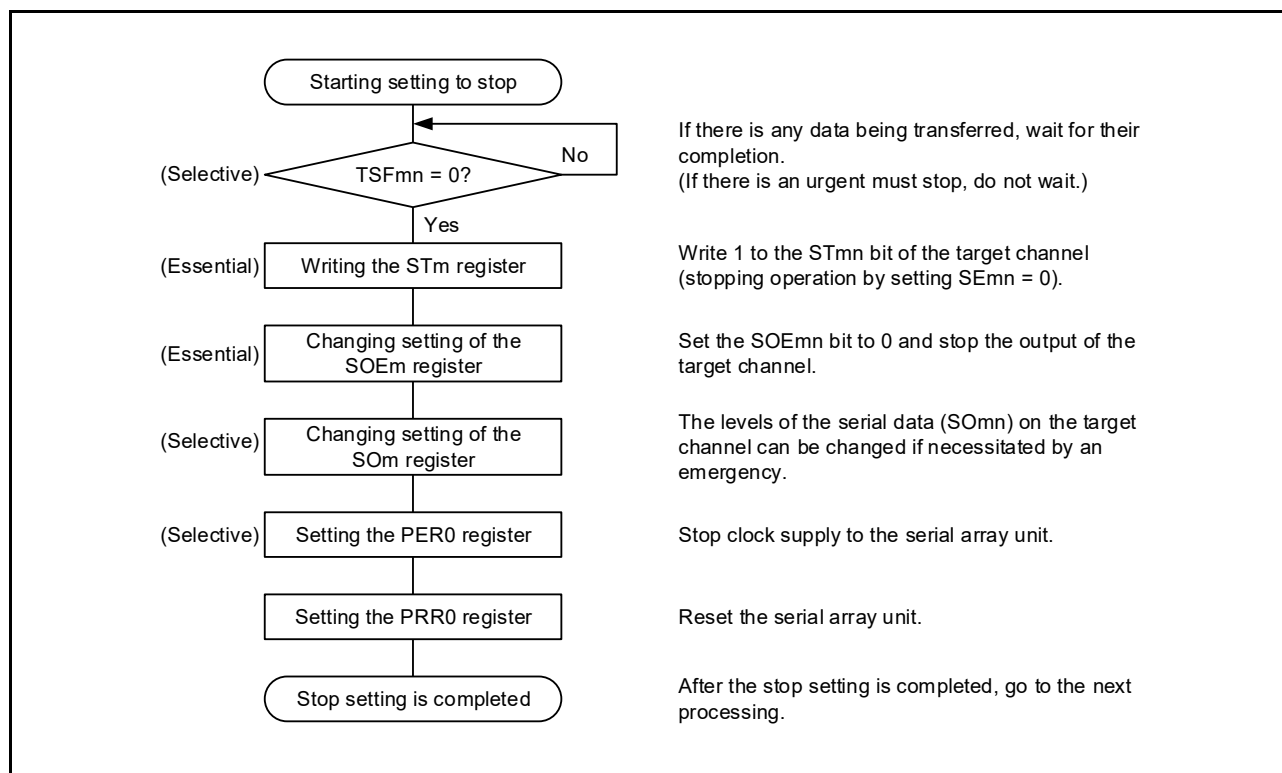
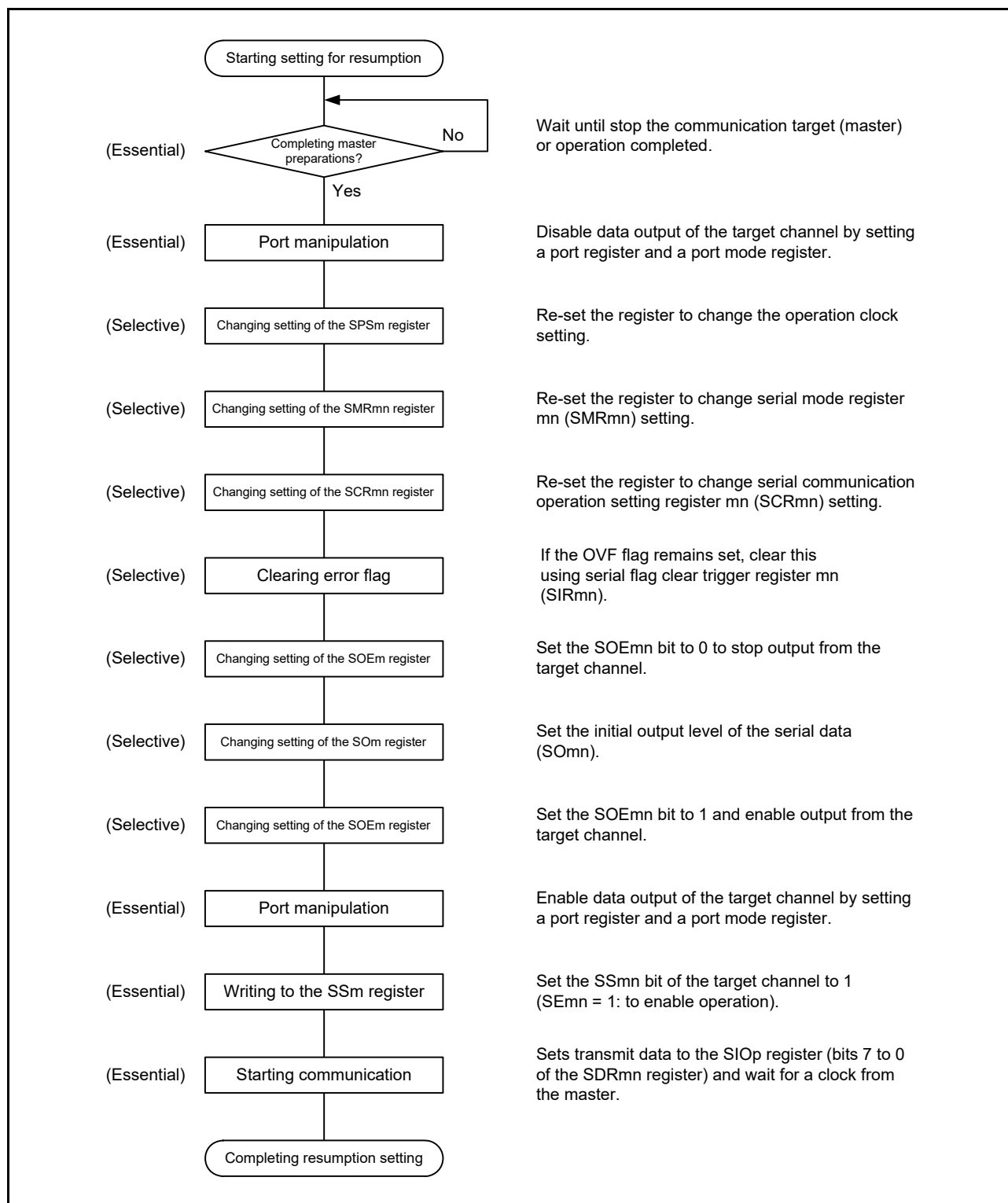


Figure 19 - 67 Procedure for Resuming Slave Transmission/Reception

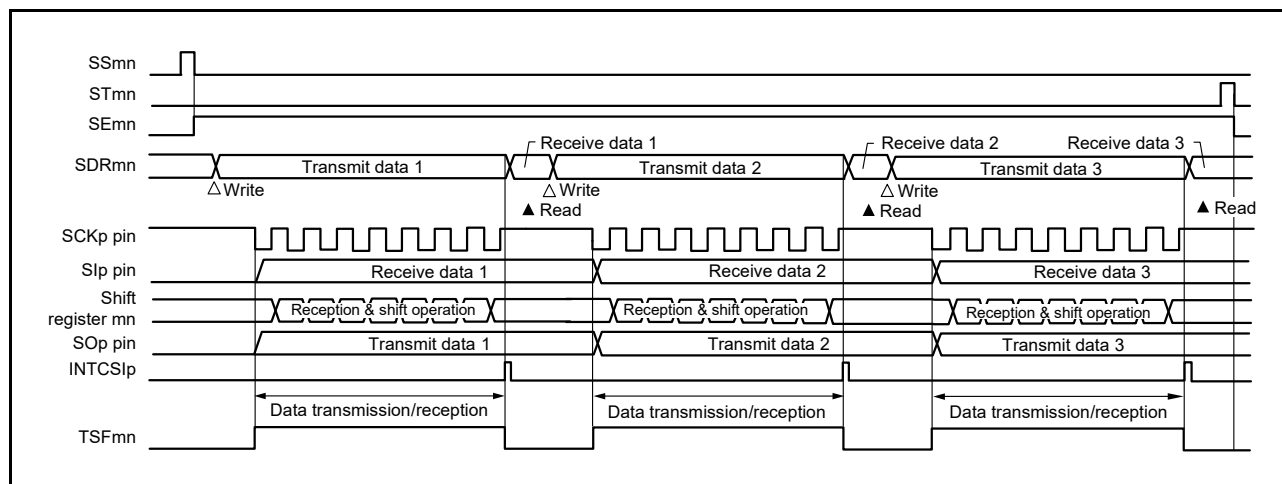


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

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 with initialization instead of restarting the communication.

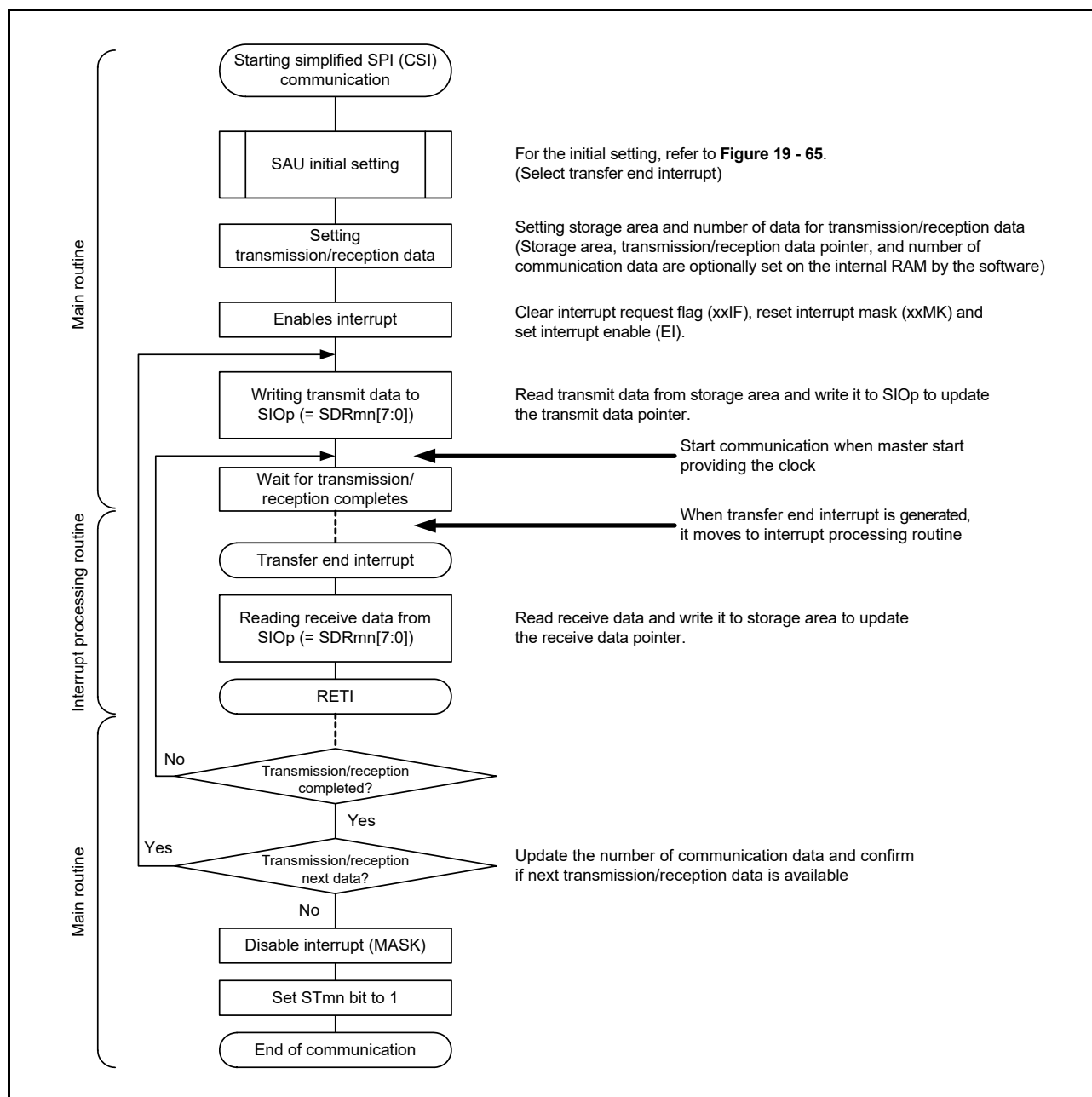
3. Processing flow (in single-transmission/reception mode)

Figure 19 - 68 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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

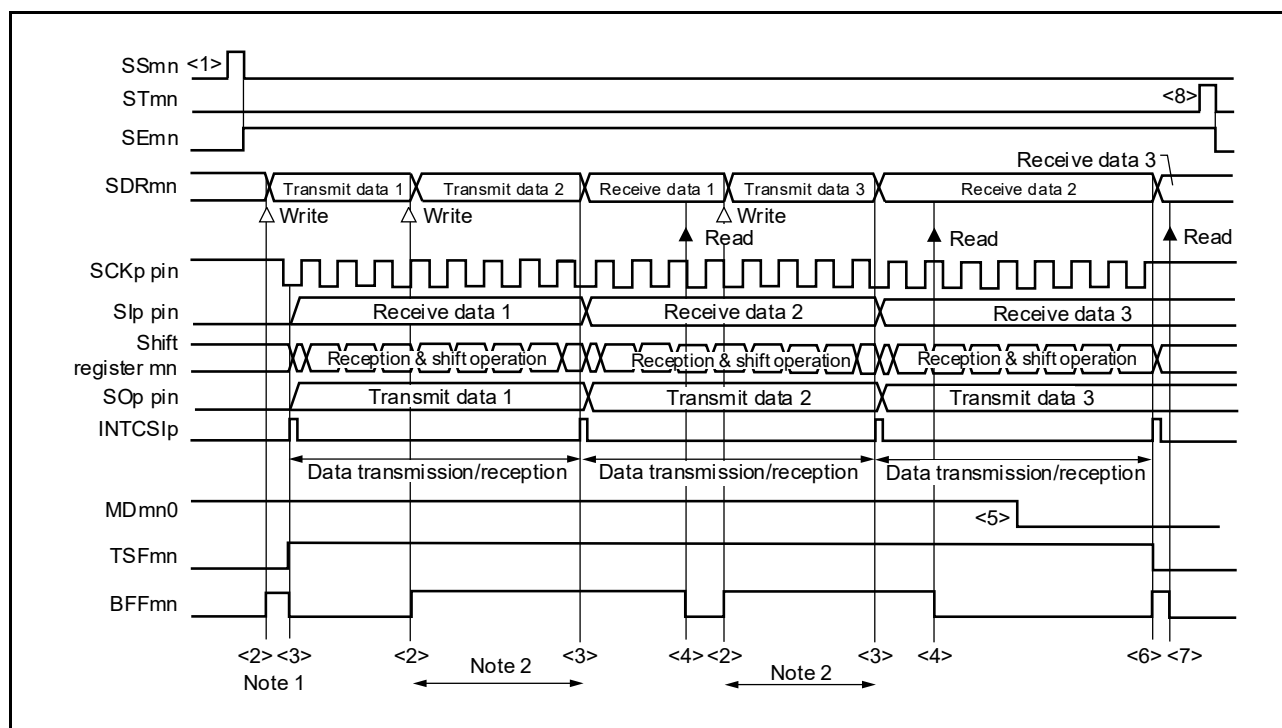
Figure 19 - 69 Flowchart of Slave Transmission/Reception (in Single-transmission/reception Mode)



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 19 - 70 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.

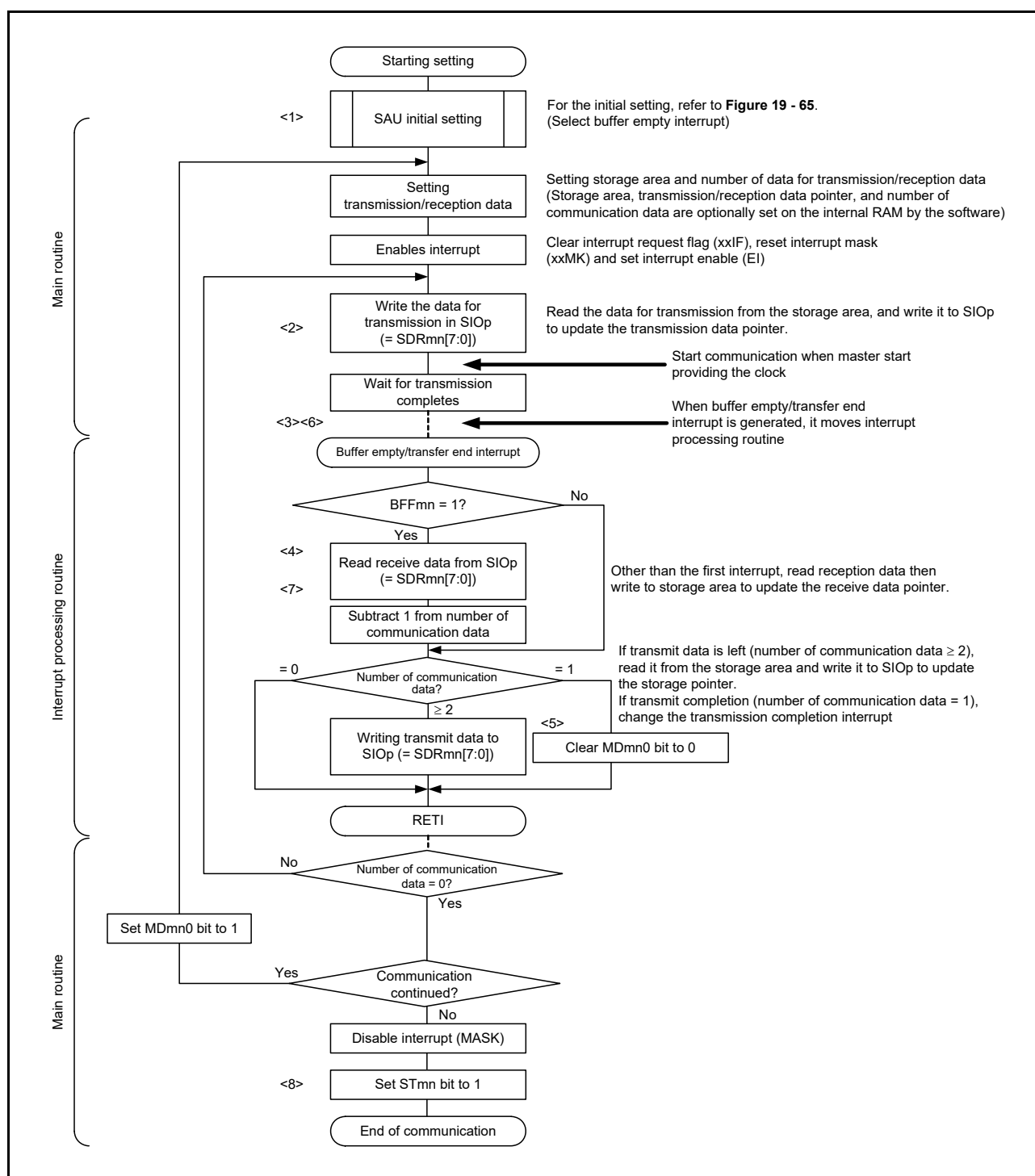
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 19 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 19 - 71 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 19 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

19.5.7 SNOOZE Mode Function

The SNOOZE mode allows the simplified SPI (CSl) reception on detection of the SCKp pin input while in the STOP mode. Normally the simplified SPI (CSl) stops communication in the STOP mode. However, using the SNOOZE mode enables the simplified SPI (CSl) to perform reception without CPU operation upon detection of the SCKp pin input. Only the following channels can be set to the SNOOZE mode.

- 44-pin products: CSI00
- 48- to 100-pin products: CSI00, CSI20

When using the simplified SPI (CSl) in SNOOZE mode, make the following setting before switching to the STOP mode (See **Figure 19 - 73 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 19 - 75 Flowchart of SNOOZE Mode Operation (continuous startup)**.)

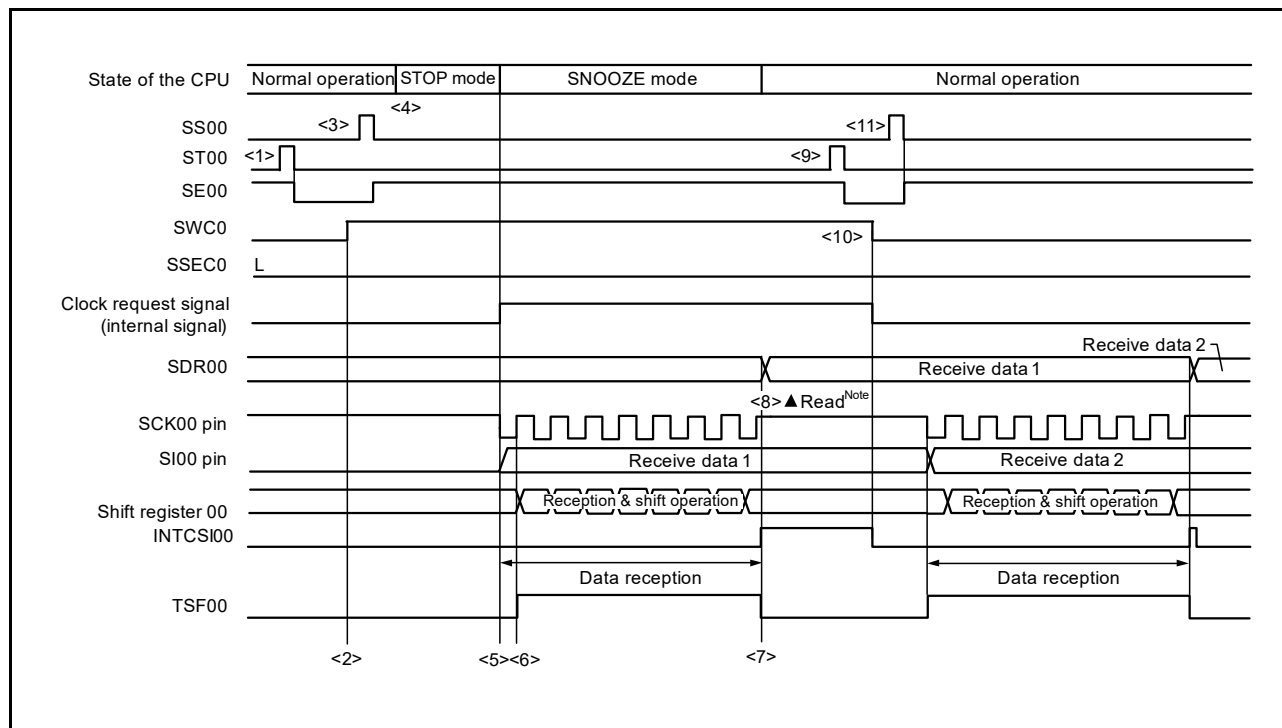
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSlp starts reception on detecting input of the serial clock on the SCKp 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 CSlp in the SNOOZE mode is 1 Mbps.

1. SNOOZE mode operation (once startup)

Figure 19 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

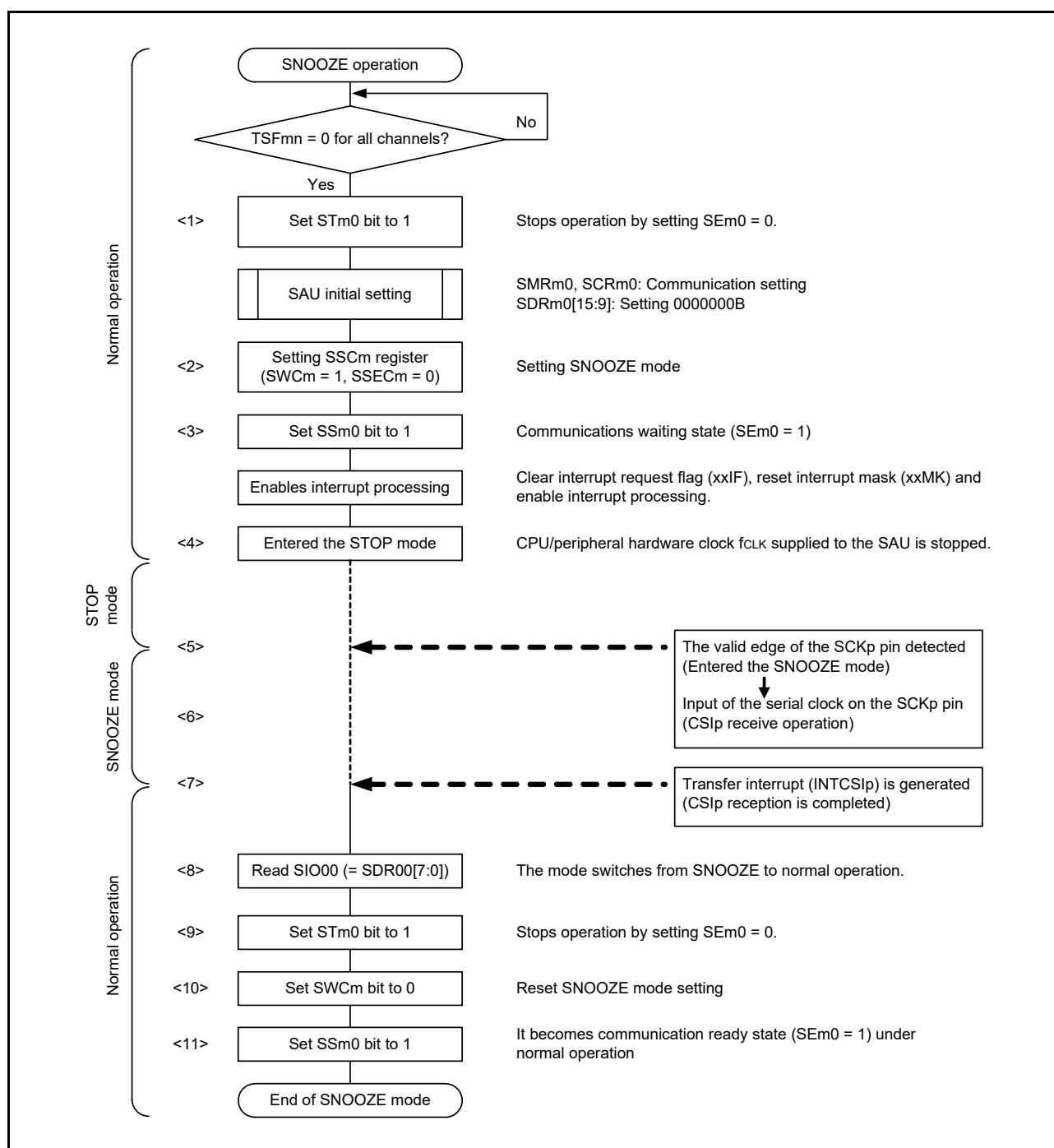
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVFM0 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 73 Flowchart of SNOOZE Mode Operation (once startup).

Remark 2. 44-pin products: m = 0; p = 00
48- to 100-pin products: m = 0, 1; p = 00, 20

Figure 19 - 73 Flowchart of SNOOZE Mode Operation (once startup)

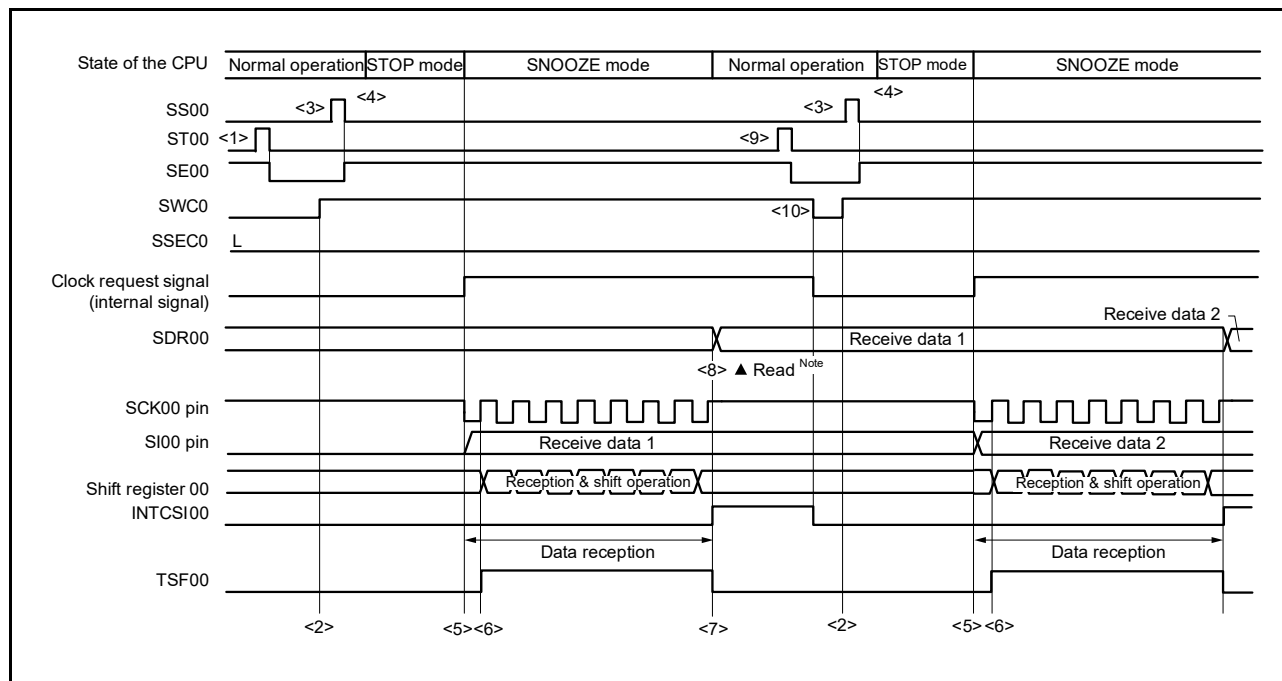


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 19 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0).**

Remark 2. 44-pin products: m = 0; p = 00
48- to 100-pin products: m = 0, 1; p = 00, 20

2. SNOOZE mode operation (continuous startup)

Figure 19 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

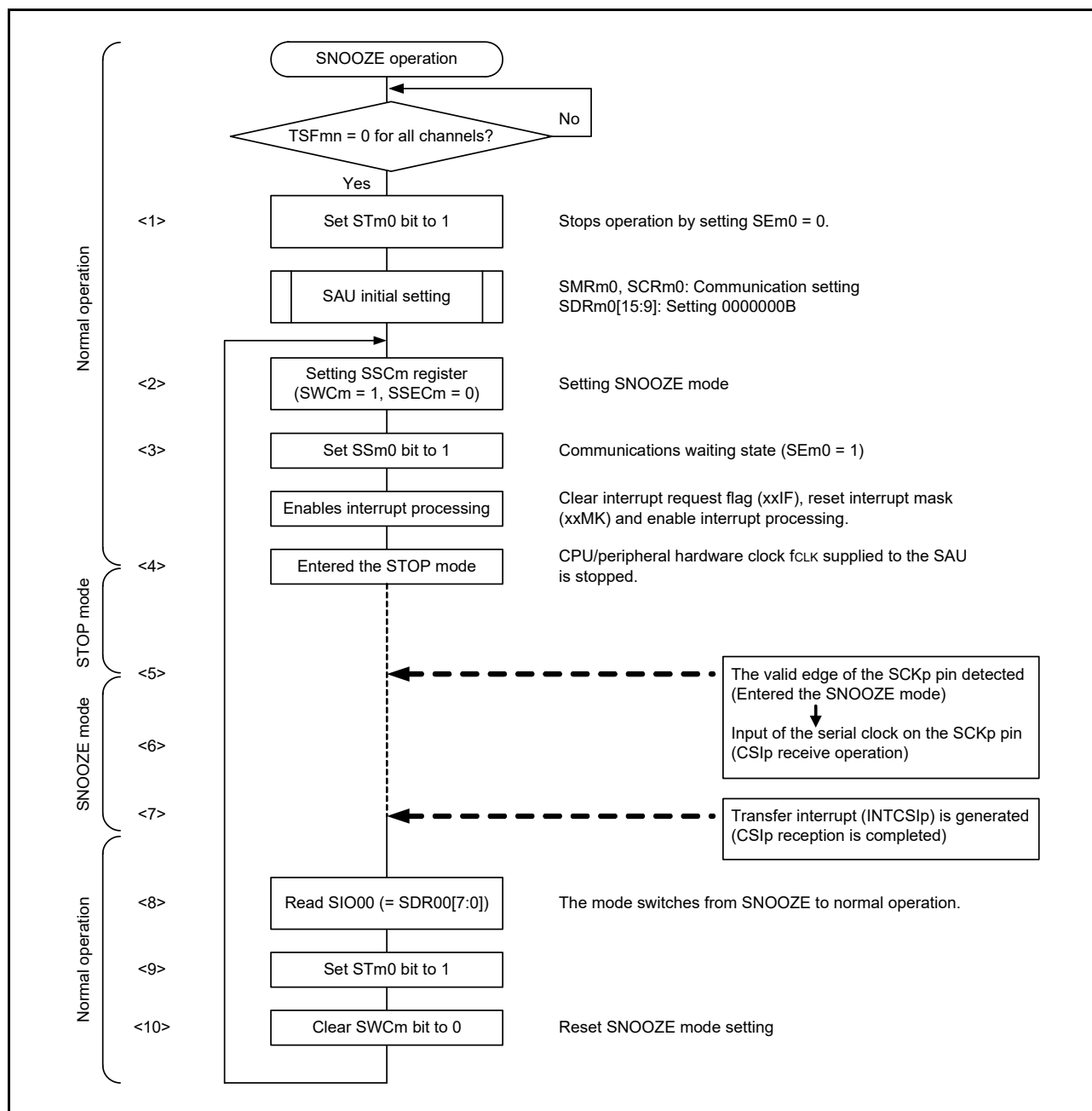
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEM0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVfm0 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 19 - 75 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. 44-pin products: m = 0; p = 00
48- to 100-pin products: m = 0, 1; p = 00, 20

Figure 19 - 75 Flowchart of SNOOZE Mode Operation (continuous startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 19 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0).**

Remark 2. 44-pin products: m = 0; p = 00
48- to 100-pin products: m = 0, 1; p = 00, 20

19.5.8 Calculating Transfer Clock Frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication can be calculated by the following expressions.

1. Master

$$(\text{Transfer clock frequency}) = \{\text{Operation clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [Hz]}$$

2. Slave

$$(\text{Transfer clock frequency}) = \{\text{Frequency of serial clock (SCK) supplied by master}\} \text{Note [Hz]}$$

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 operation clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 19 - 3 Selection of Operation Clock For Simplified SPI (1/2)

SMRmn Register	SPSm Register								Operation 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
	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 19 - 3 Selection of Operation Clock For Simplified SPI (2/2)

SMRmn Register	SPSm Register								Operation Clock (fCLK) ^{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 to 3), mn = 00 to 03, 10 to 13

19.5.9 Procedure for Processing Errors that Occurred During Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication is described in **Table 19 - 4**.

Table 19 - 4 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 to 3), mn = 00 to 03, 10 to 13

The serial data output (SO00, SO01) and serial clock output (SCK00, SCK01) of channels 0 and 1 of serial array unit 0 can be selected as input for the logic and event link controller (ELCL). When the serial data output and serial clock output signals are to be used for event input to the ELCL but not for output to the SO0m and SCK0m pins, set the corresponding bit in port function output enable register 0 (PFOE0) to 0. See **4.3.14 Port function output enable registers (PFOEx)** for details.

19.6 Operation of UART (UART0 to UART3) 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 and UART2 reception support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1).

[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 Only UART0 and UART2 support 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.

UART3 uses channels 2 and 3 of SAU1.

<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	(CSI01)		(IIC01)
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	—	UART2 (supporting LIN-bus)	—
	1	CSI21		IIC21
	2	—	—	(IIC30)
	3	—		—

<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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	—	—	IIC30
	3	—		—

<52-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	—		—

<64-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<80-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<100-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

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. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

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.

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

UART handles the following four types of communication.

- UART transmission (See **19.6.1.**)
- UART reception (See **19.6.2.**)
- LIN transmission (UART2 only) (See **19.7.1.**)
- LIN reception (UART2 only) (See **19.7.2.**)

19.6.1 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	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	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. Only UART0 and UART2 support 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 42 Electrical Characteristics**.

Remark 1. f_{MCK} : Operation clock frequency of target channel

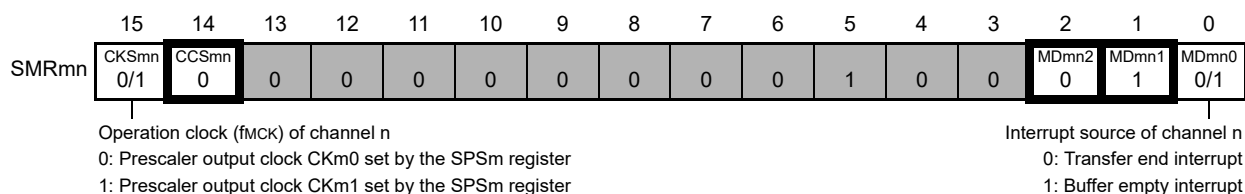
f_{CLK} : System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

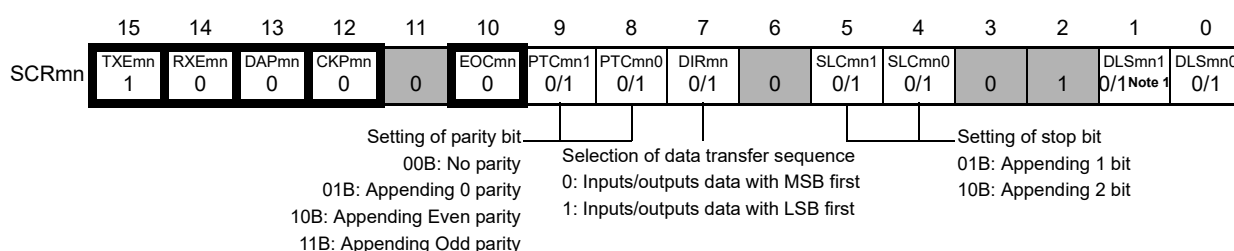
1. Register setting

Figure 19 - 76 Example of Contents of Registers for UART Transmission of UART (UART0 to UART3)

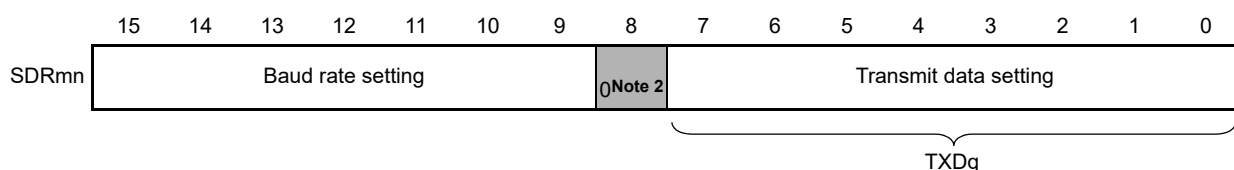
a) Serial mode register mn (SMRmn)



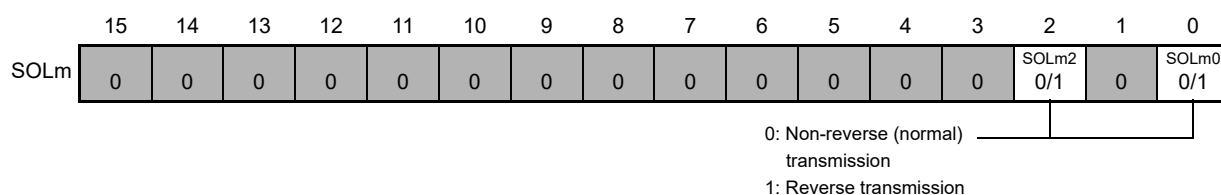
b) Serial communication operation setting register mn (SCRmn)



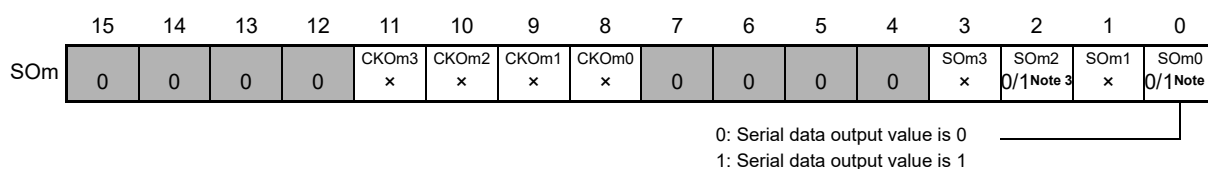
c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



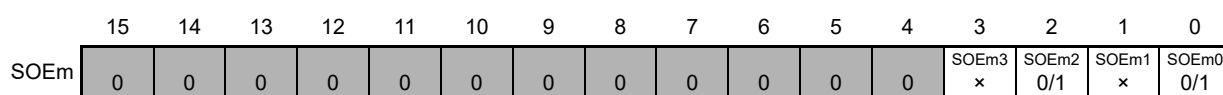
d) Serial output level register m (SOLm): Set only the bit of the target channel.



e) Serial output register m (SOM): Set only the bit of the target channel.



f) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



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

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 x	SSm2 0/1	SSm1 x	SSm0 0/1

Note 1. This bit is only present in the SCR00 and SCR10 registers. This bit is fixed to 1 for the other registers.

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area. Only UART0 and UART2 support the 9-bit data length.

Note 3. 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), q: UART number (q = 0 to 3),
mn = 00, 02, 10, 12

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 19 - 77 Initial Setting Procedure for UART Transmission

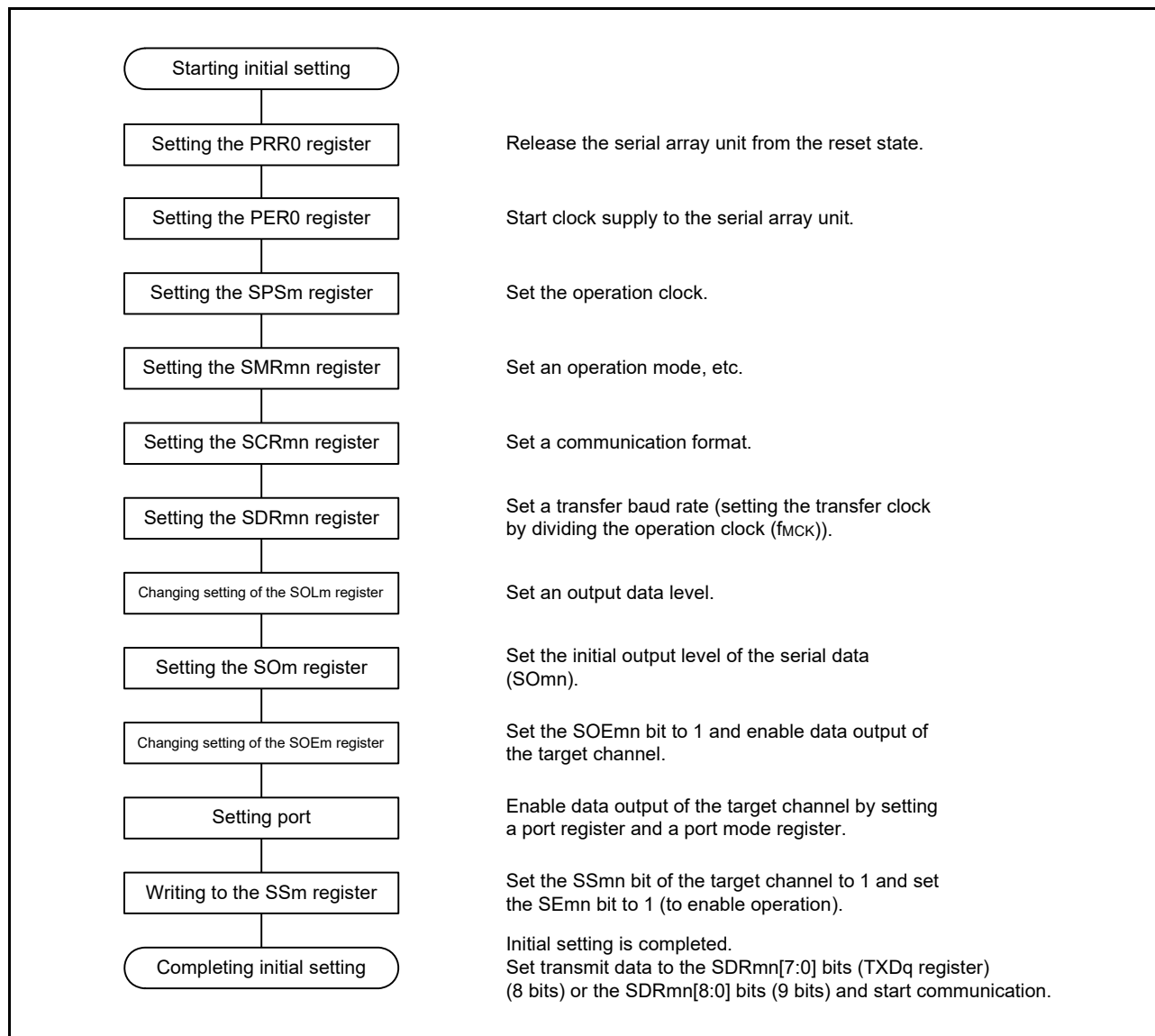


Figure 19 - 78 Procedure for Stopping UART Transmission

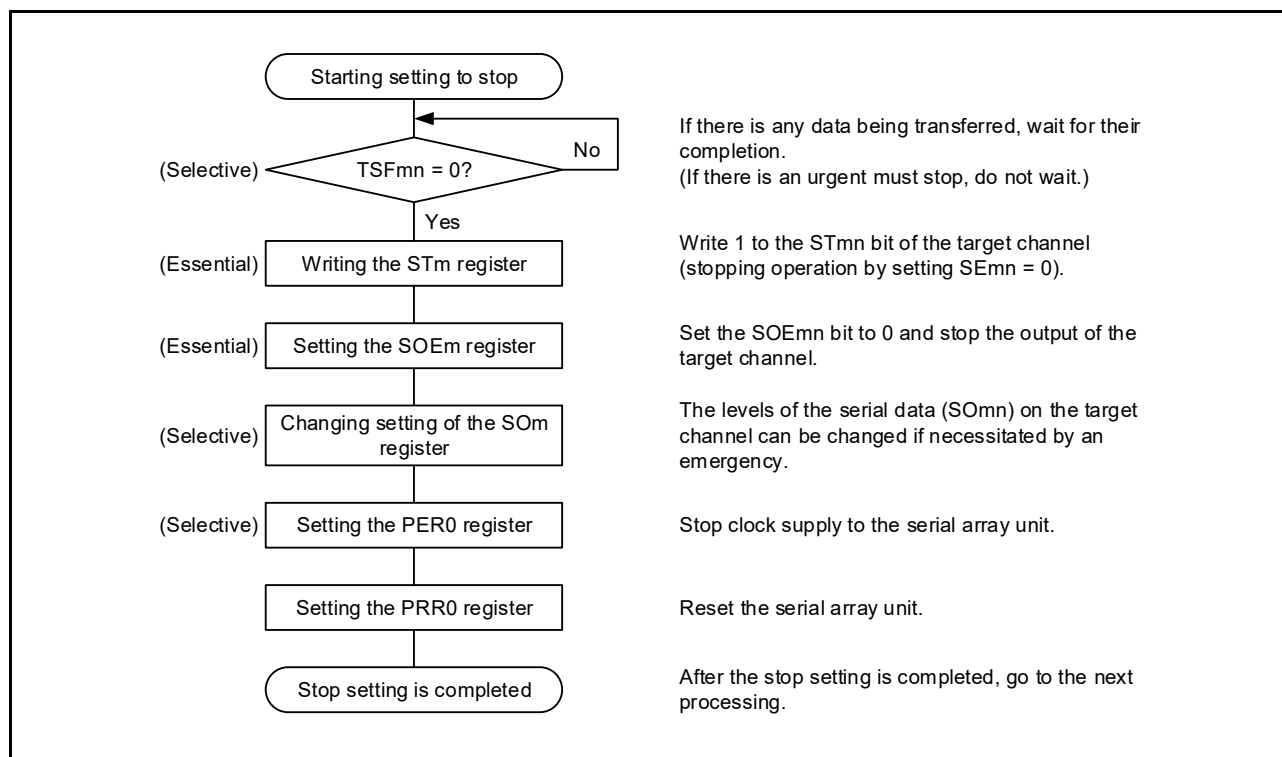
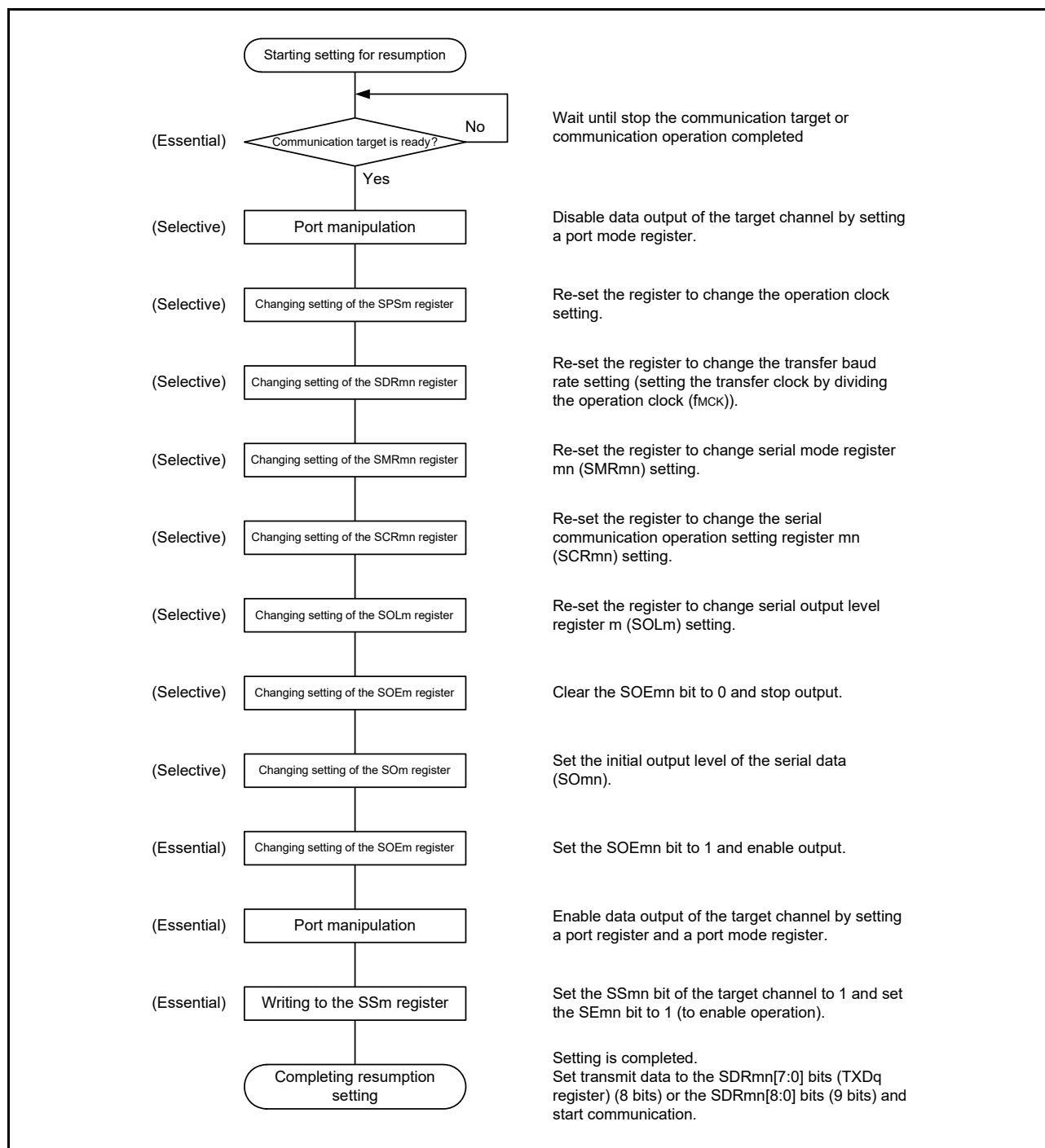


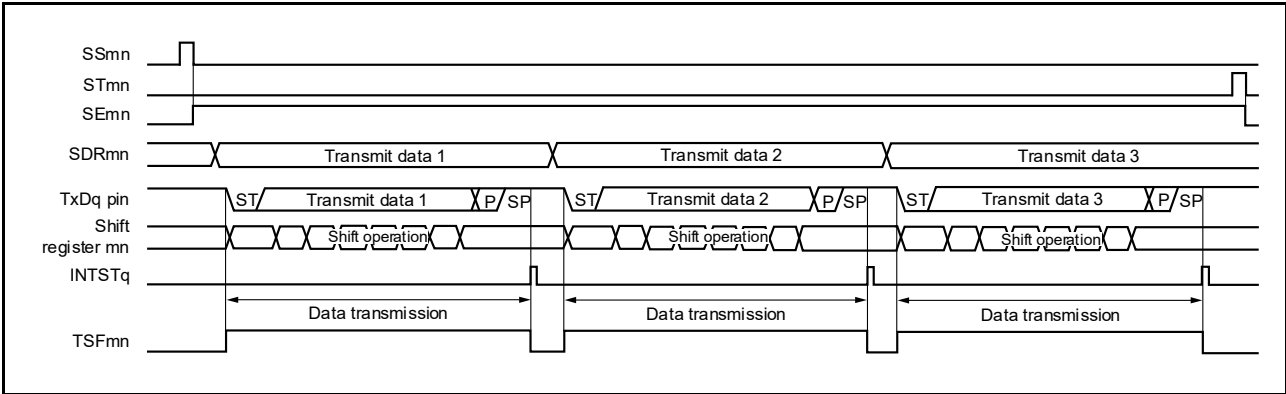
Figure 19 - 79 Procedure for Resuming UART Transmission



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 perform initialization instead of restarting the communication.

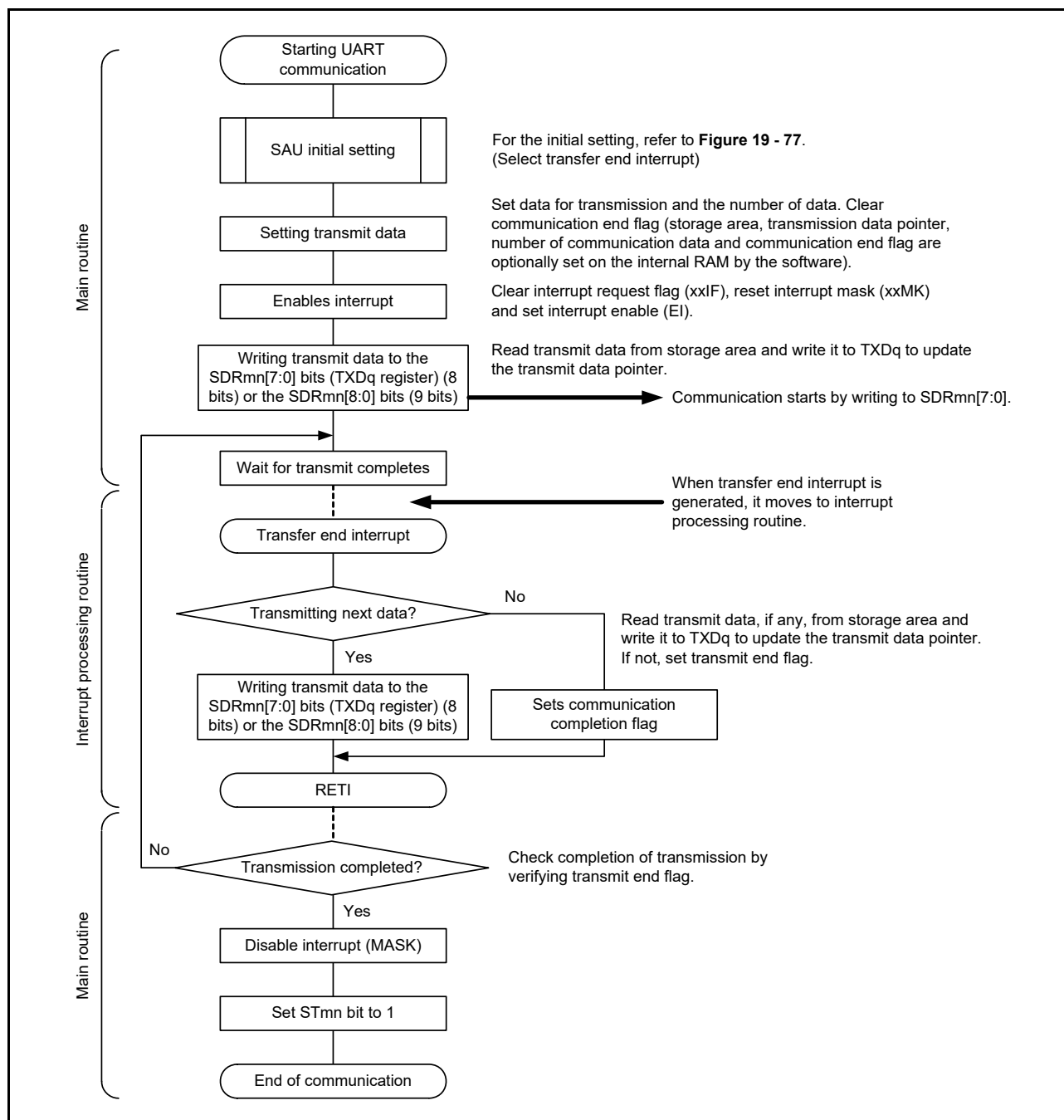
3. Processing flow (in single-transmission mode)

Figure 19 - 80 Timing Chart of UART Transmission (in Single-transmission Mode)



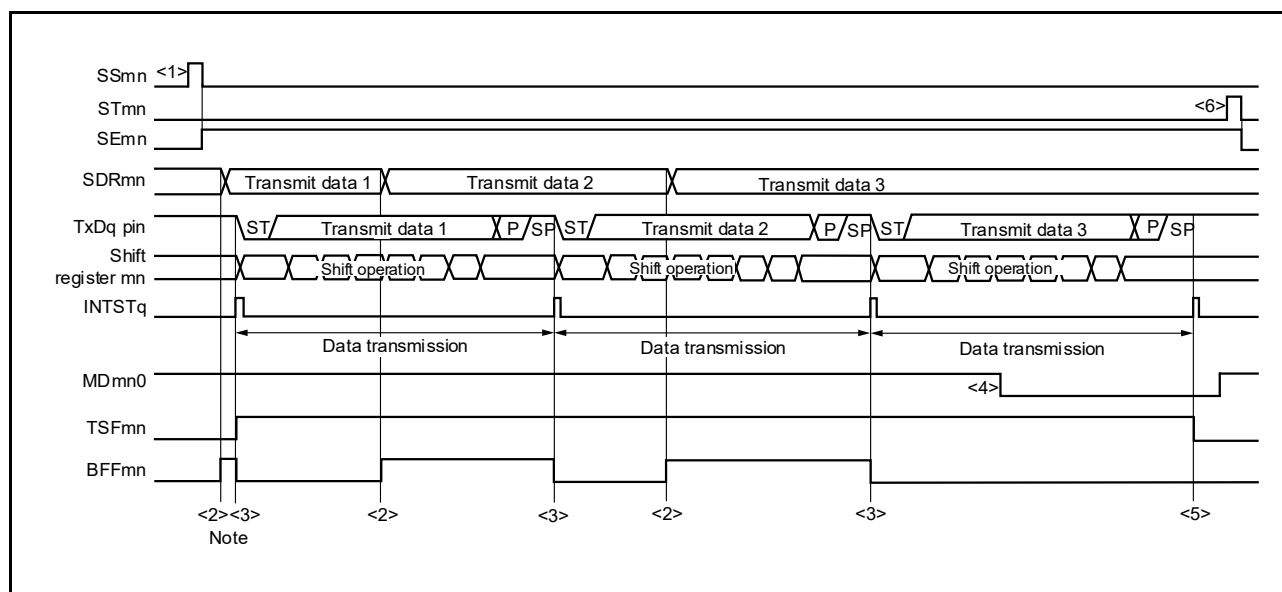
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3),
mn = 00, 02, 10, 12

Figure 19 - 81 Flowchart of UART Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 19 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)

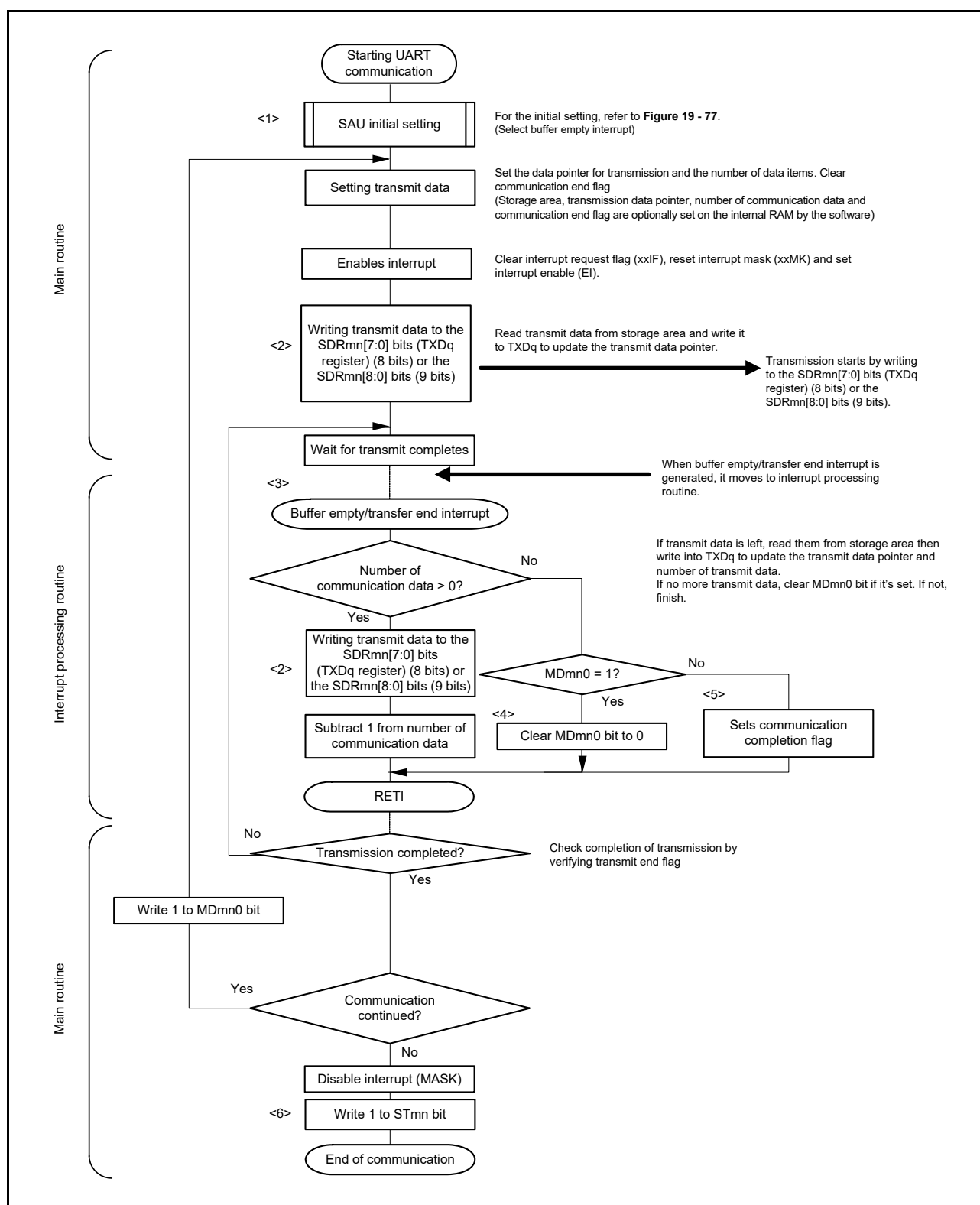


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 3),
mn = 00, 02, 10, 12

Figure 19 - 83 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 19 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

19.6.2 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	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
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. Only UART0 and UART2 support 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 42 Electrical Characteristics**.

Remark 1. f_{MCK} : Operation 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, 13

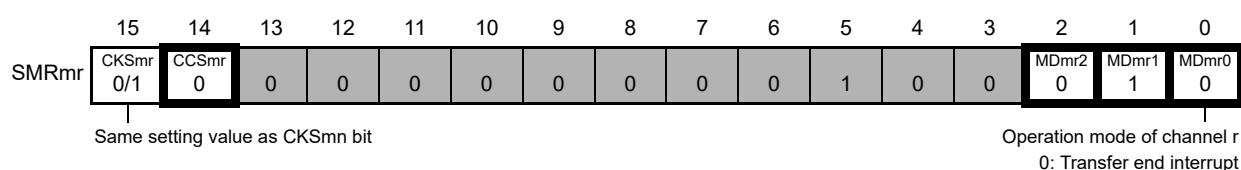
1. Register setting

Figure 19 - 84 Example of Contents of Registers for UART Reception of UART (UART0 to UART3)

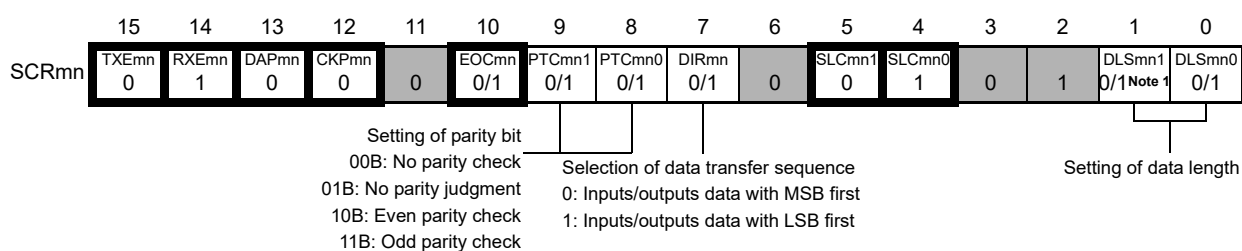
a) Serial mode register mn (SMRmn)



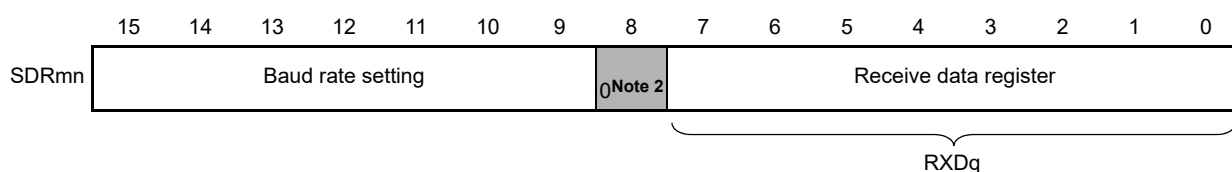
b) Serial mode register mr (SMRmr)



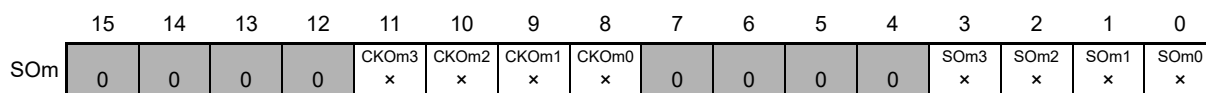
c) Serial communication operation setting register mn (SCRmn)



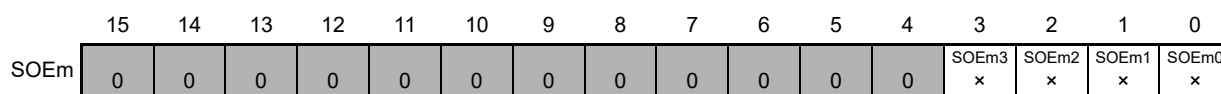
d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



e) Serial output register m (SOM): This register is not used in this mode.



f) Serial output enable register m (SOEm): This register is not used in this mode.



(Notes, Caution, and Remarks are listed on the next page.)

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 ×

Note 1. This bit is only present in the SCR01 and SCR11 registers. This bit is fixed to 1 for the other registers.

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the reception data specification area. Only UART0 and UART2 support the 9-bit data length.

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, 13

r: Channel number (r = n – 1), q: UART number (q = 0 to 3)

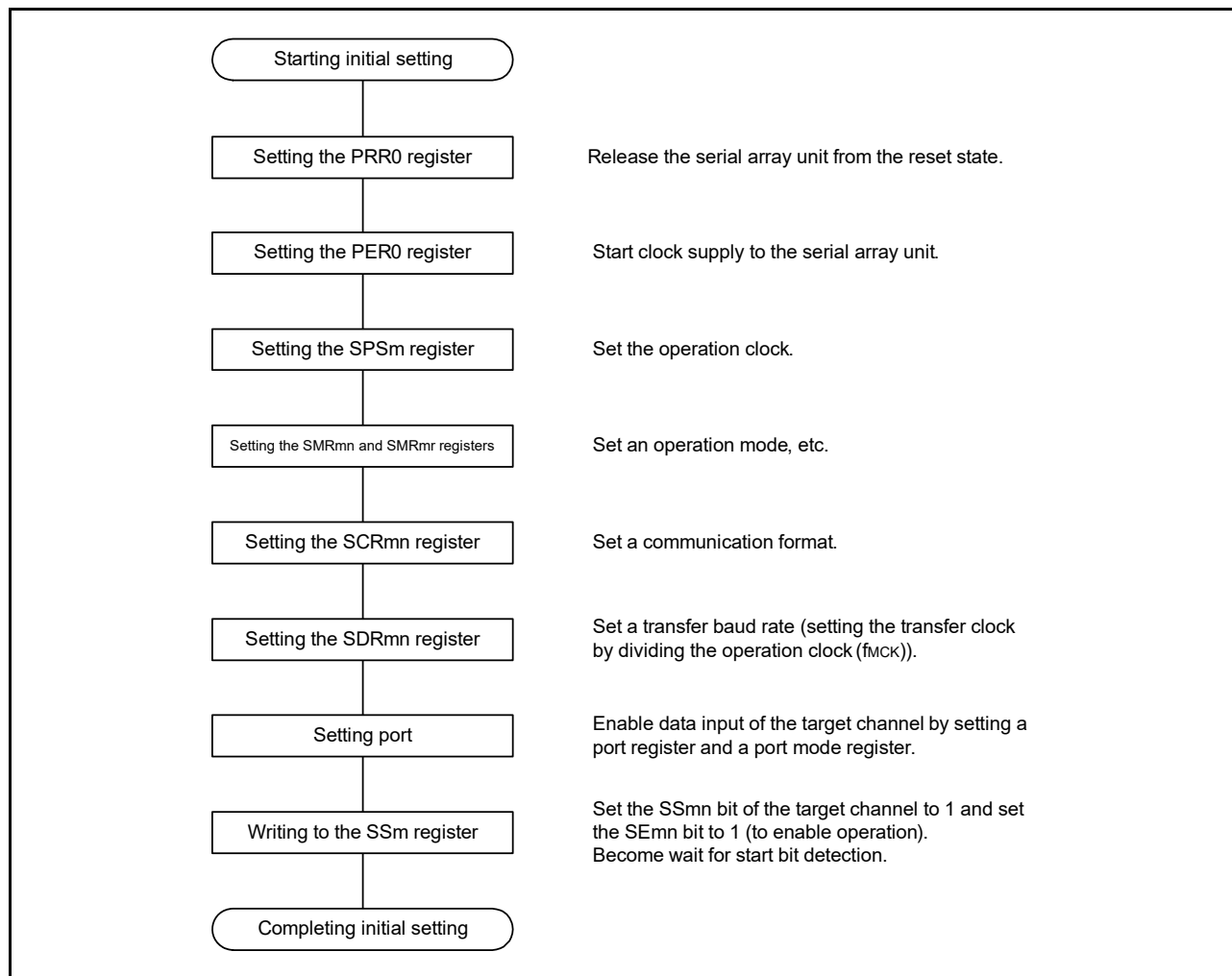
Remark 2. : Setting is fixed in the UART reception mode, : Setting disabled (set to the initial value)

×: 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 19 - 85 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 four cycles of fMCK have elapsed.

Figure 19 - 86 Procedure for Stopping UART Reception

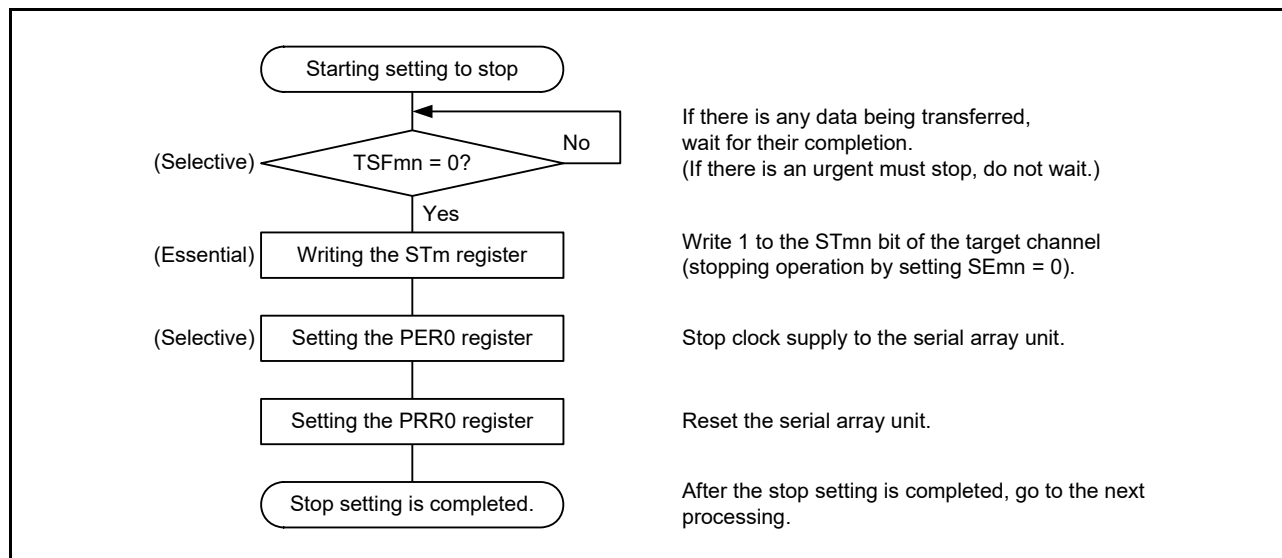
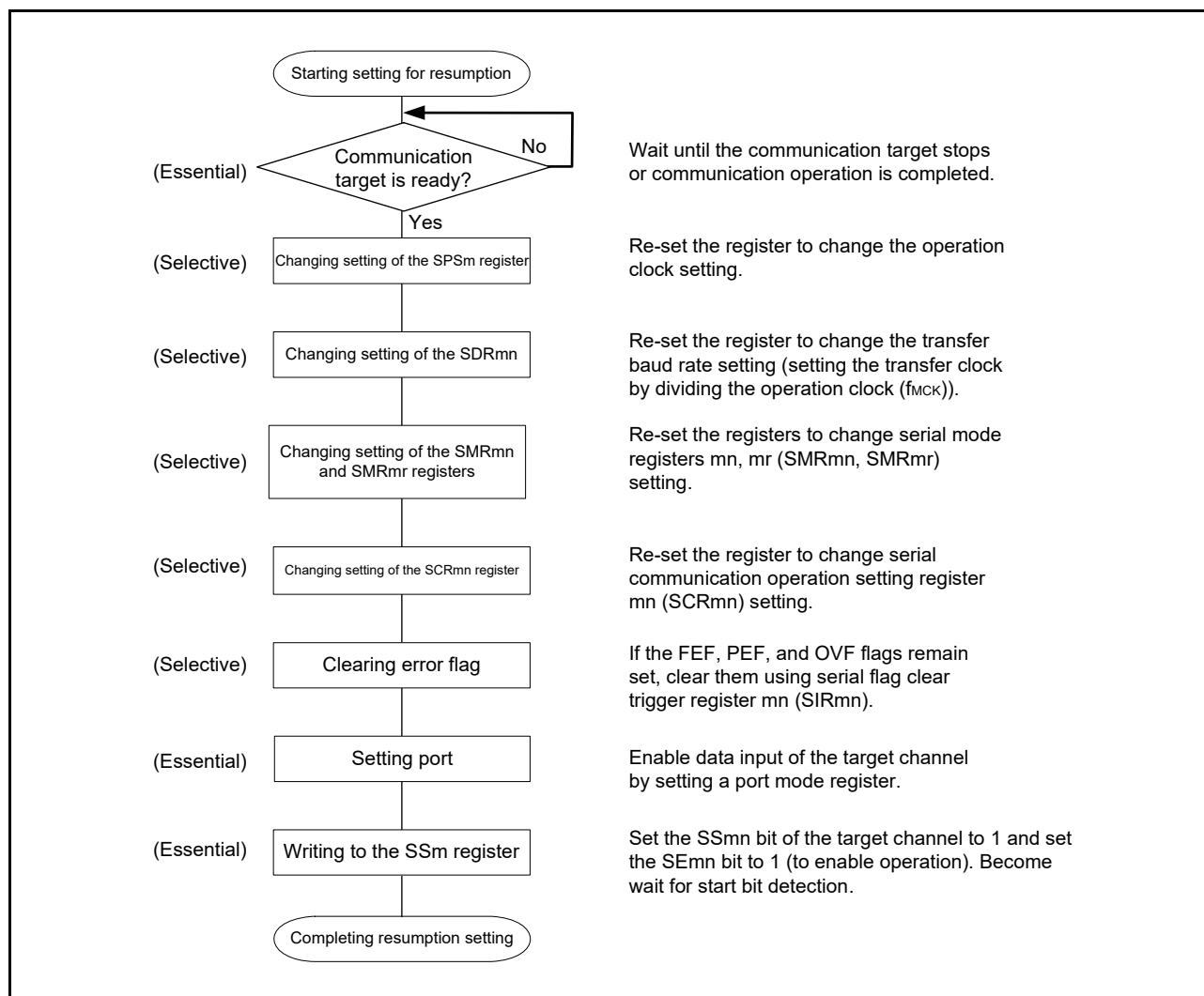


Figure 19 - 87 Procedure for Resuming UART Reception

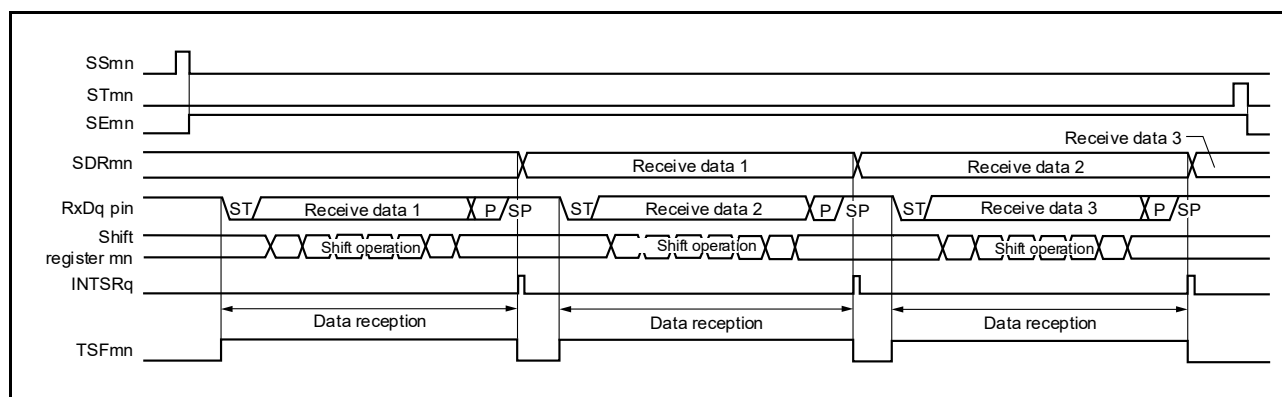


Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least four cycles of fmck have elapsed.

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 perform initialization instead of restarting the communication.

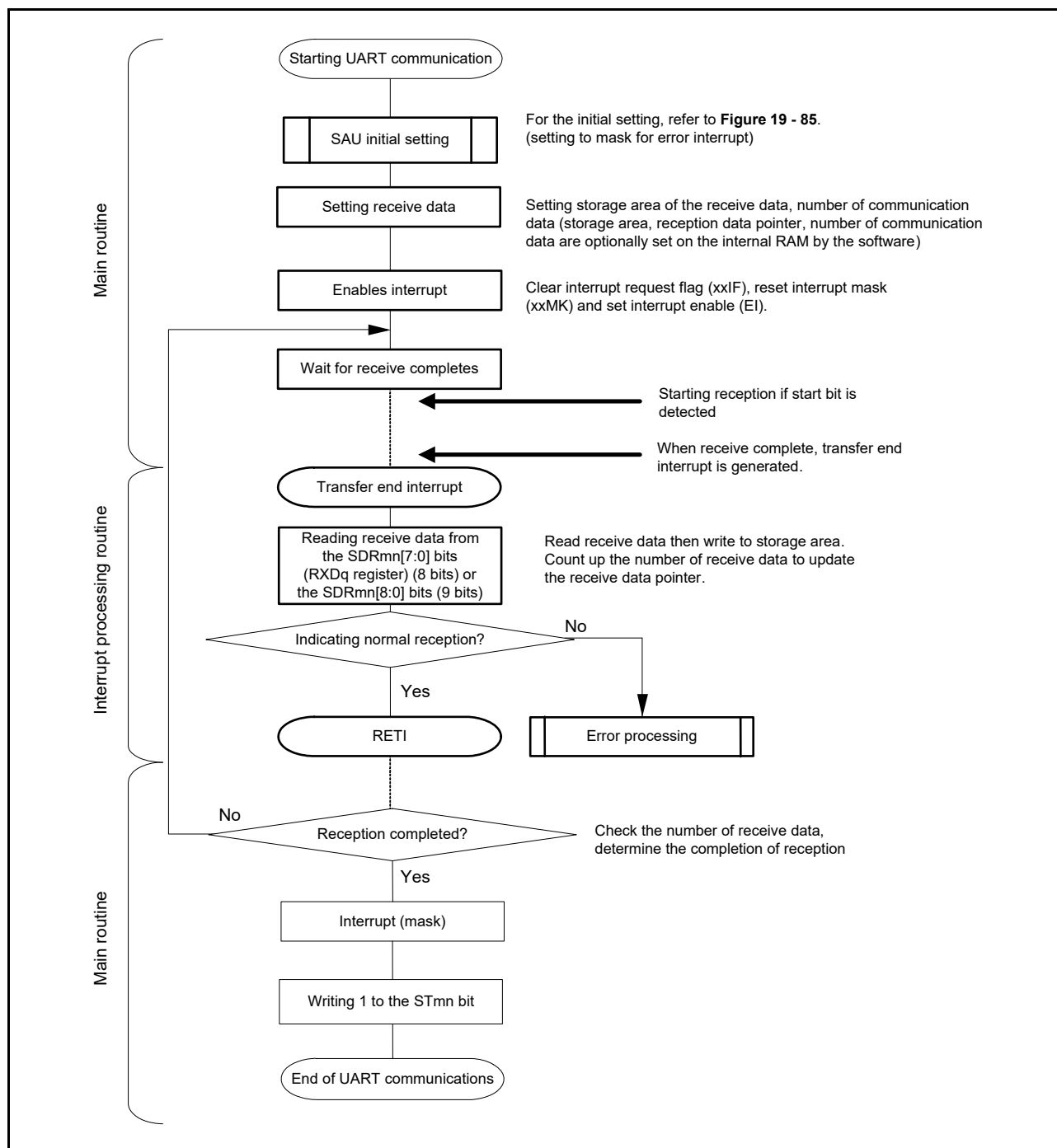
3. Processing flow

Figure 19 - 88 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13
 r: Channel number (r = n - 1), q: UART number (q = 0 to 3)

Figure 19 - 89 Flowchart of UART Reception



19.6.3 SNOOZE Mode

The SNOOZE mode allows the UART reception on detection of the RxDq pin input while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception without CPU operation. Only UART0 reception and UART2 reception can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 19 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1 or EOCm1 = 1, SSECM = 0)** and **Figure 19 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECM = 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 SPSm register and bits 15 to 9 of the SDRmn register with reference to **Table 19 - 5**.
- Set the EOCmn and SSECM bits. This is for enabling or stopping generation of an error interrupt (INTSREq) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq 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 setting 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 SWCm = 1, UARTq 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 SWCm 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 returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECM bit is set to 1, the PEFmn, FEFmn, or OVFMn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECM = 1 is made, clear the PEFmn, FEFmn, and OVFMn flags before setting the SWCm bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq 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 RxDq 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 19 - 5 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})	SDRmn[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%

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})	SDRmn[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.

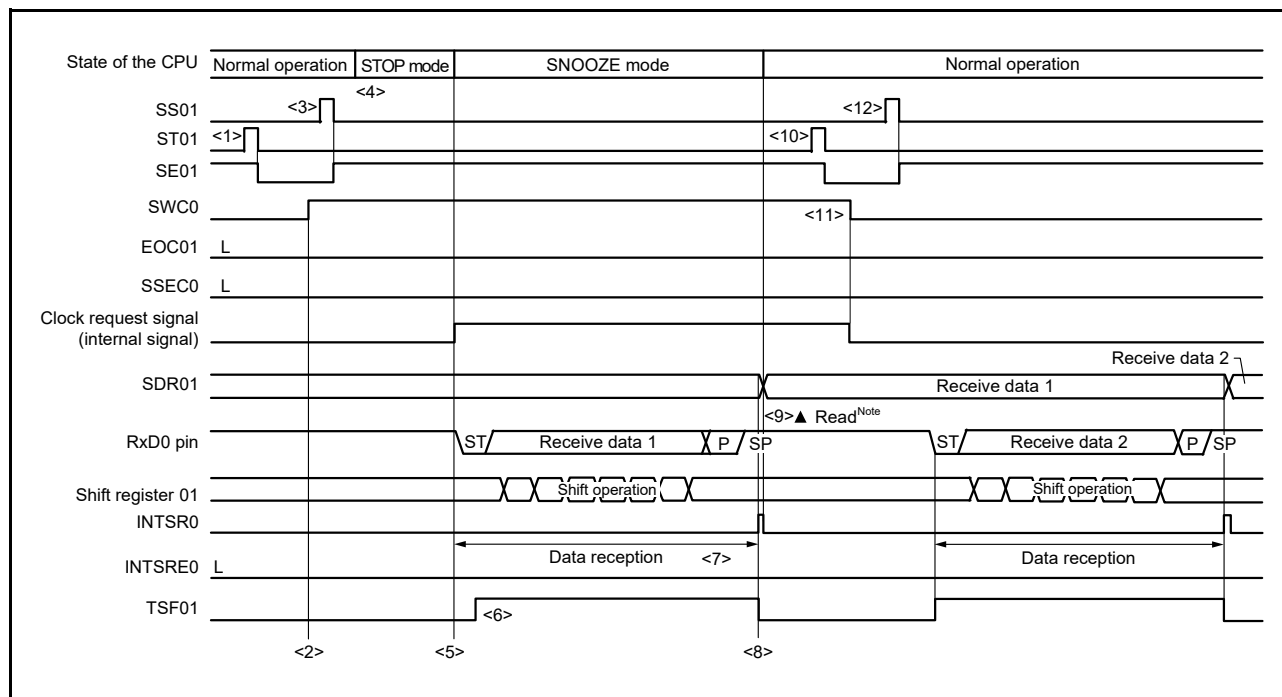
- In the case of f_{IH} ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ±2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception.

1. SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 19 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm = 1.

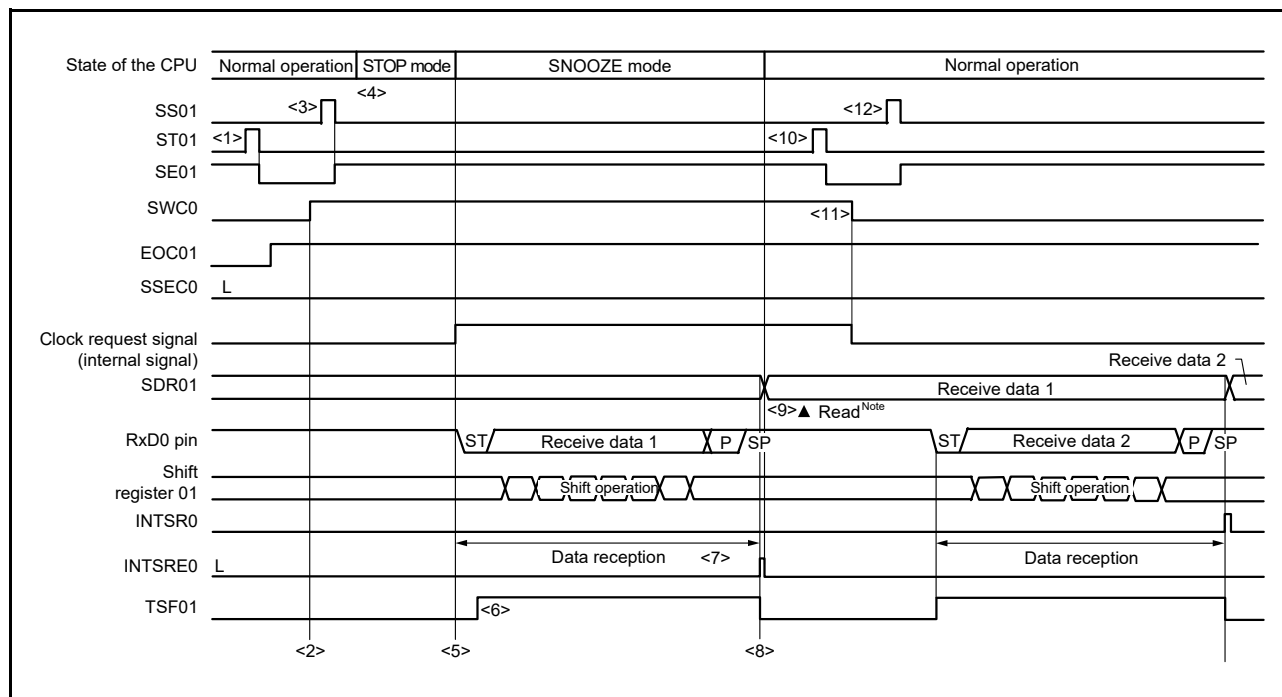
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEM1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 19 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0, 1; q = 0, 2

2. SNOOZE mode operation (EOCm1 = 1, SSECM = 0: Error interrupt (INTSREq) generation is enabled)
Because EOCm1 = 1 and SSECM = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 19 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECM = 0)



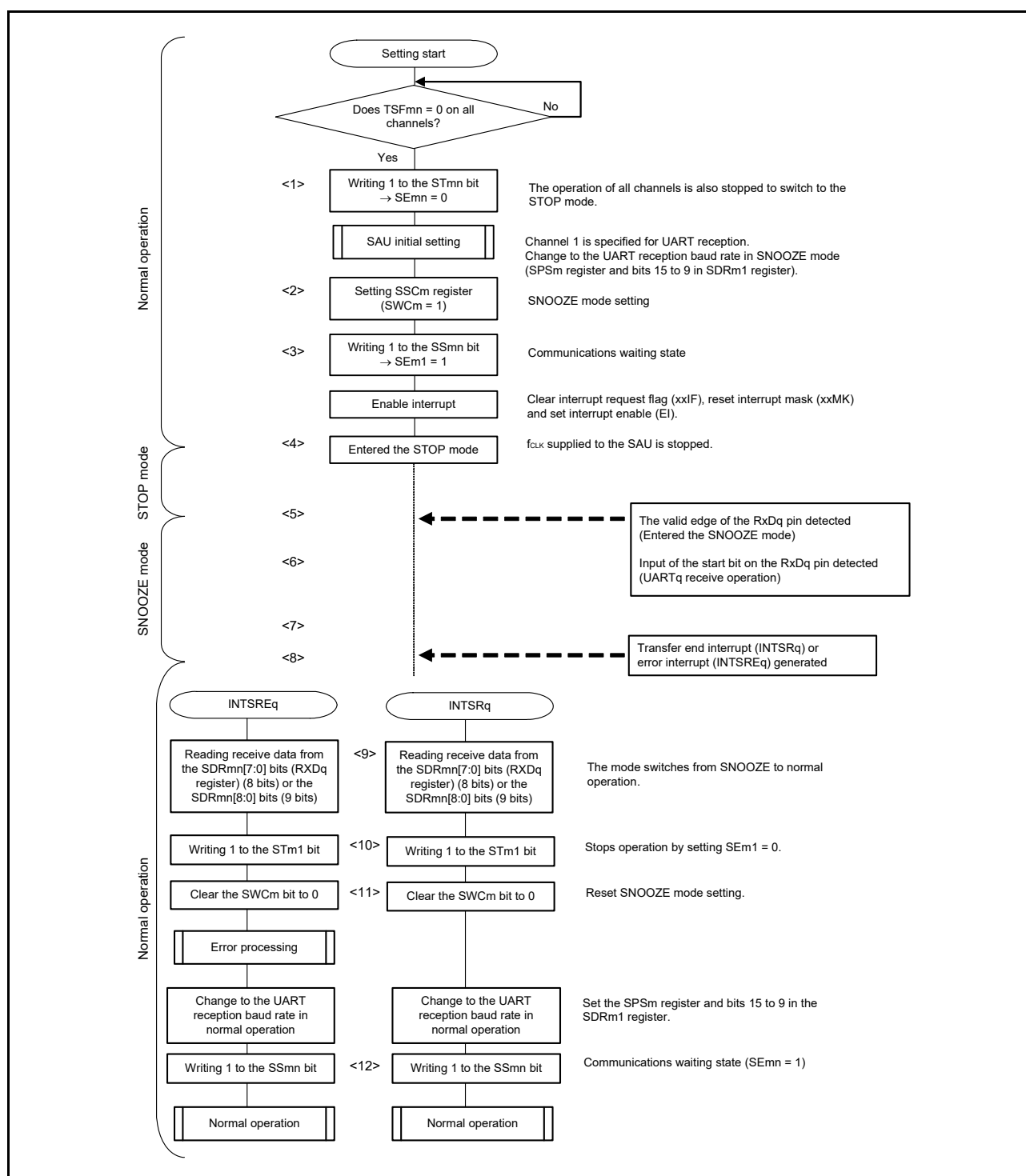
Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEM1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 19 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1 or EOCm1 = 1, SSECM = 0).

Remark 2. m = 0, 1; q = 0, 2

Figure 19 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1 or EOCm1 = 1, SSECM = 0)

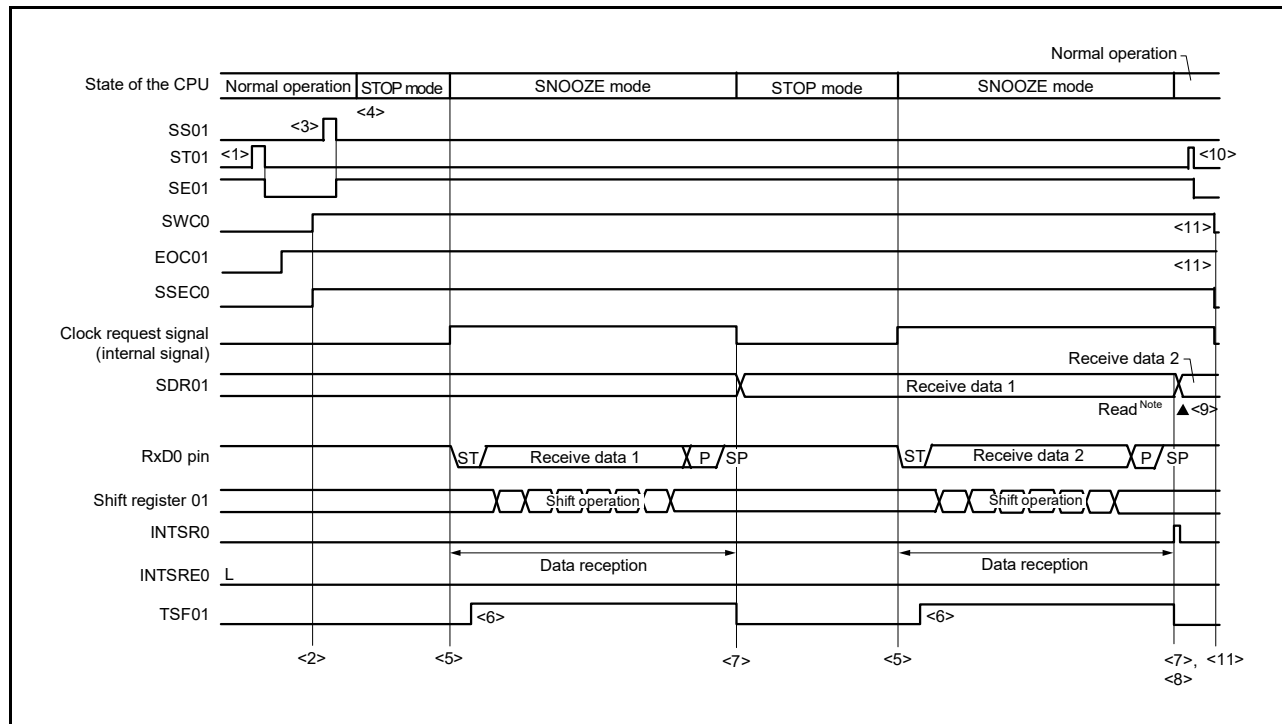


Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 19 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECM = 0/1)** and **Figure 19 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECM = 0)**.

Remark 2. m = 0, 1; q = 0, 2; n = 0 to 3

3. SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)
Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 19 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

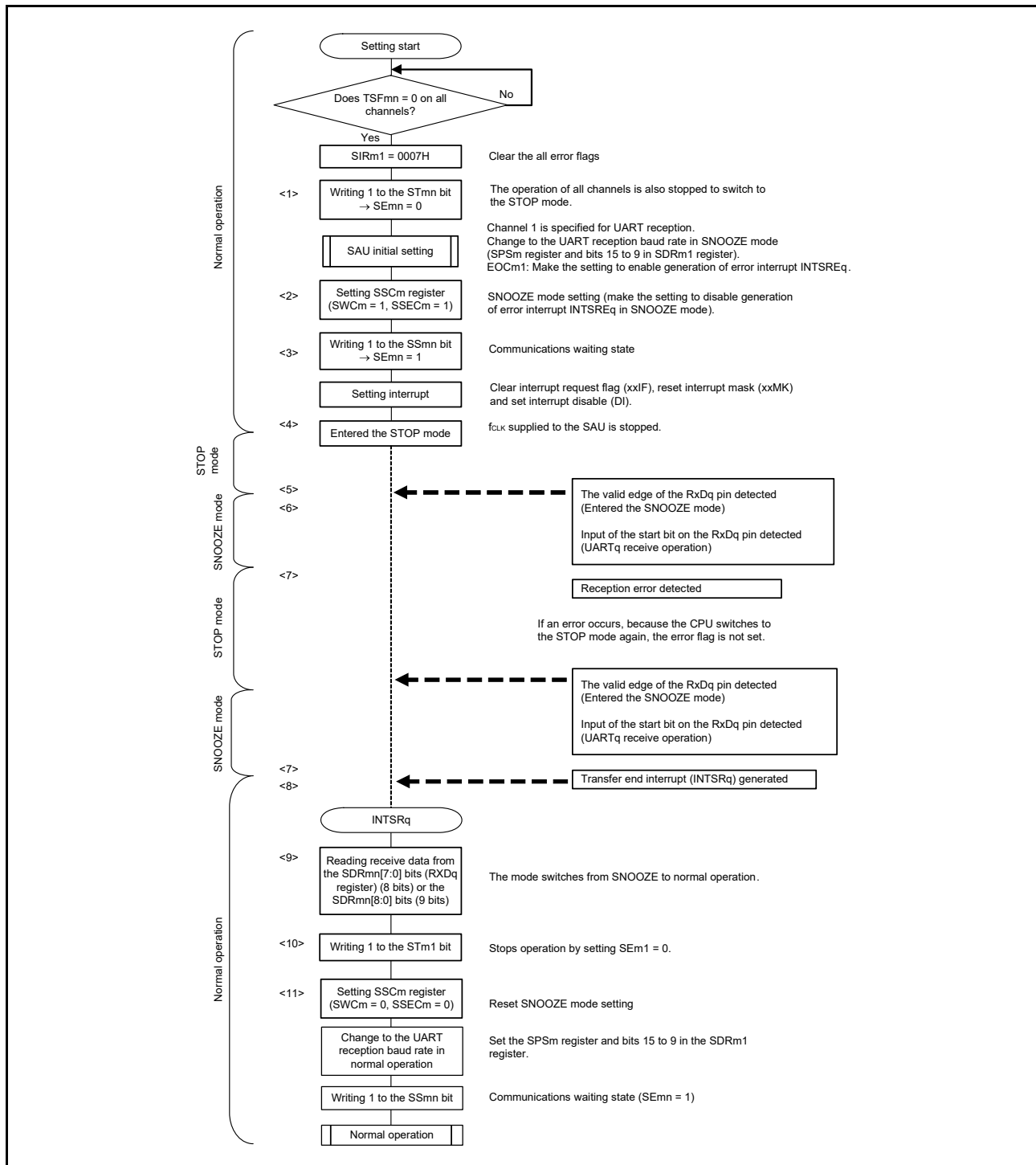
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFM1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, and OVFM1 flags before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0, 1; q = 0, 2

Figure 19 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Caution If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, and OVFm1 flags before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0, 1; q = 0, 2; n = 0 to 3

19.6.4 Calculating Baud Rate

1. Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

$$(\text{Baud rate}) = \{\text{Operation clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{bps}]$$

Caution 1. The operation clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Caution 2. 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 to 13

Table 19 - 6 Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation 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
	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
	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 to 3), mn = 00 to 03, 10 to 13

2. Baud rate error during transmission

The baud rate error of UART (UART0 to UART3) 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			
	Operation 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, 12

3. Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART3) 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 **19.6.4 Calculating Baud Rate, 1. Baud rate calculation expression.**)

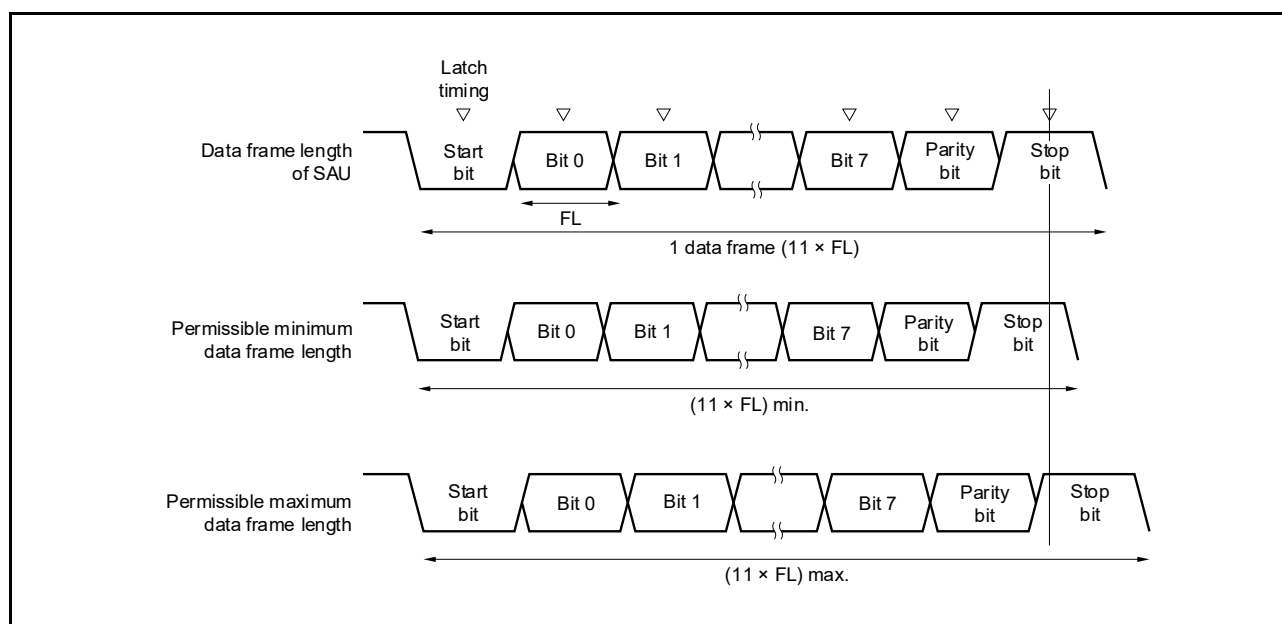
k: $\text{SDRmn}[15:9] + 1$

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, 13

Figure 19 - 95 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 19 - 95**, 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.

19.6.5 Procedure for Processing Errors that Occurred During UART (UART0 to UART3) Communication

The procedure for processing errors that occurred during UART (UART0 to UART3) communication is described in **Table 19 - 7** and **Table 19 - 8**.

Table 19 - 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 19 - 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 to 13

19.7 LIN Communication Operation

19.7.1 LIN Transmission

Of UART transmission, UART2 supports LIN communication. For LIN transmission, channel 0 of unit 1 is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported	Not 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			

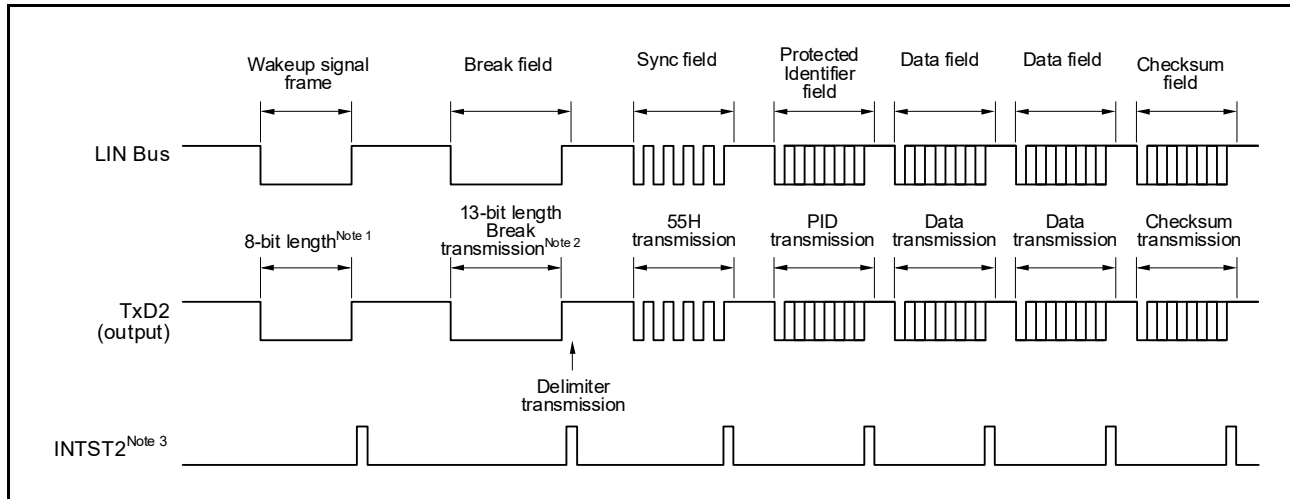
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 42 Electrical Characteristics**. In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

Remark f_{MCK} : Operation 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 19 - 96 outlines a transmission operation of LIN.

Figure 19 - 96 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.

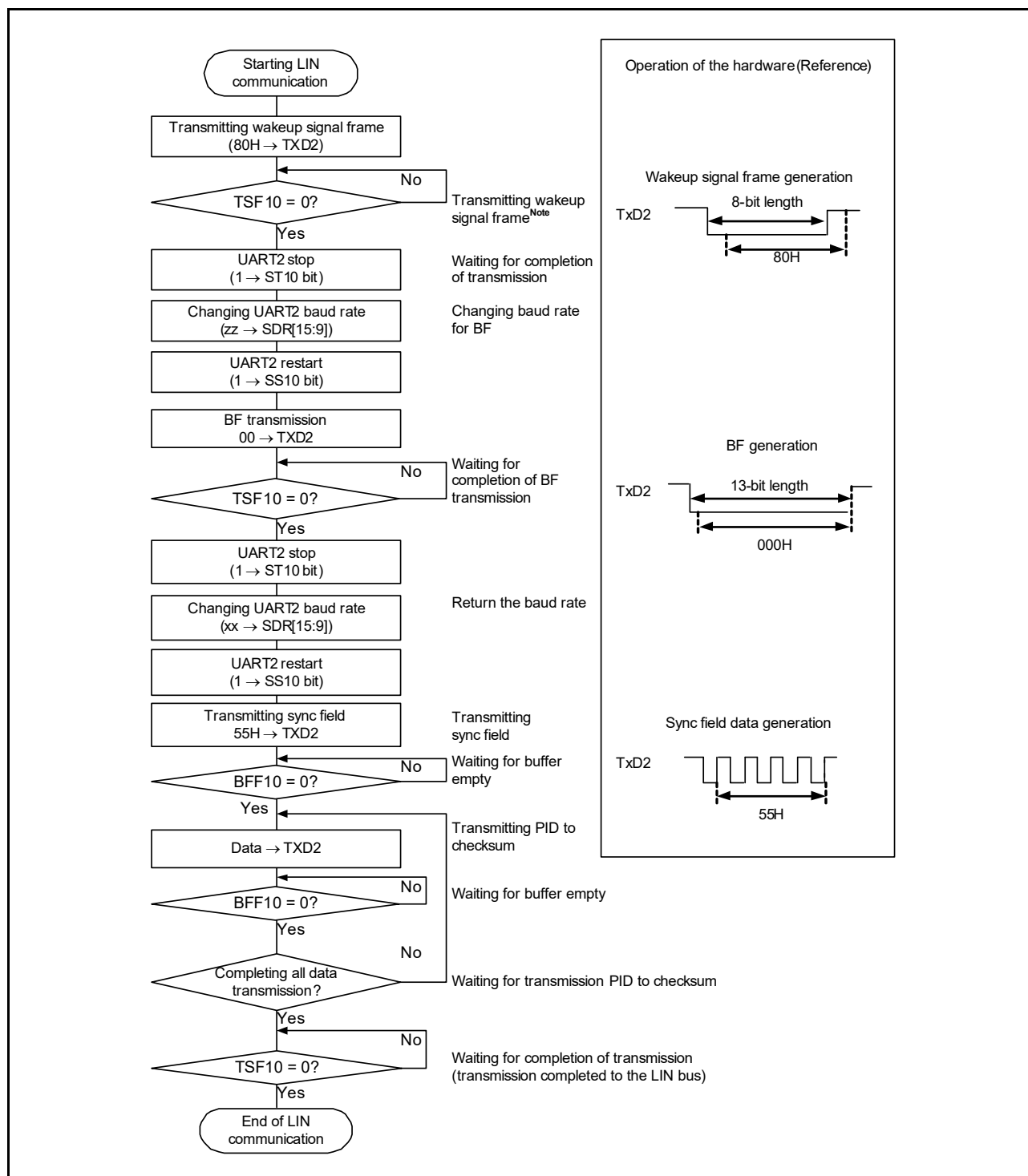
$$(\text{Baud rate of break field}) = 9/13 \times N$$

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 19 - 97 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.

19.7.2 LIN Reception

Of UART reception, UART2 supports LIN communication. For LIN reception, channel 1 of unit 1 is used.

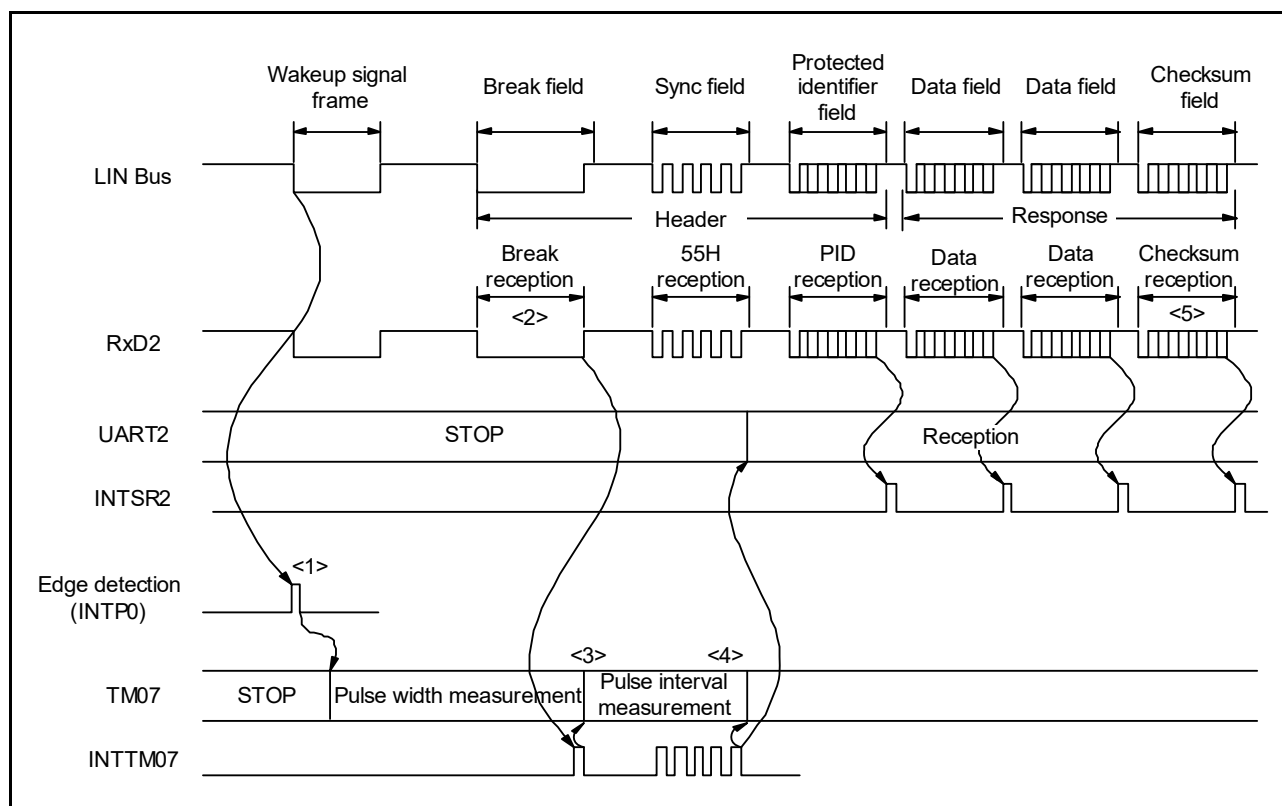
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported	Not supported
Target channel	—	—	Channel 1 of SAU1	—
Pins used	—	—	RxD2	—
Interrupt	—	—	INTSR2	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	—	—	INTSRE2	—
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 42 Electrical Characteristics**.

Remark f_{MCK} : Operation clock frequency of target channel
 f_{CLK} : System clock frequency

Figure 19 - 98 outlines a reception operation of LIN.

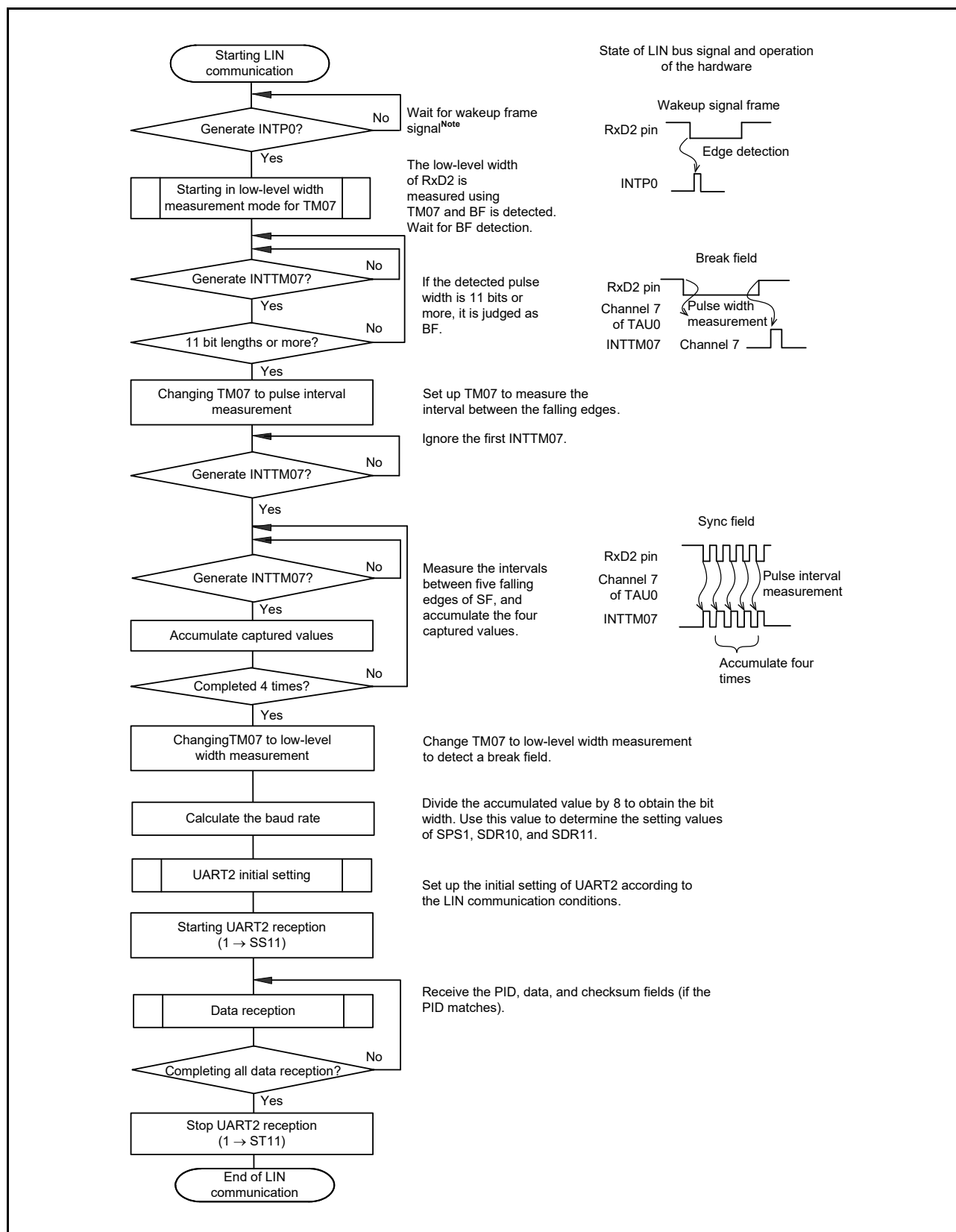
Figure 19 - 98 Reception Operation of LIN



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. (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.

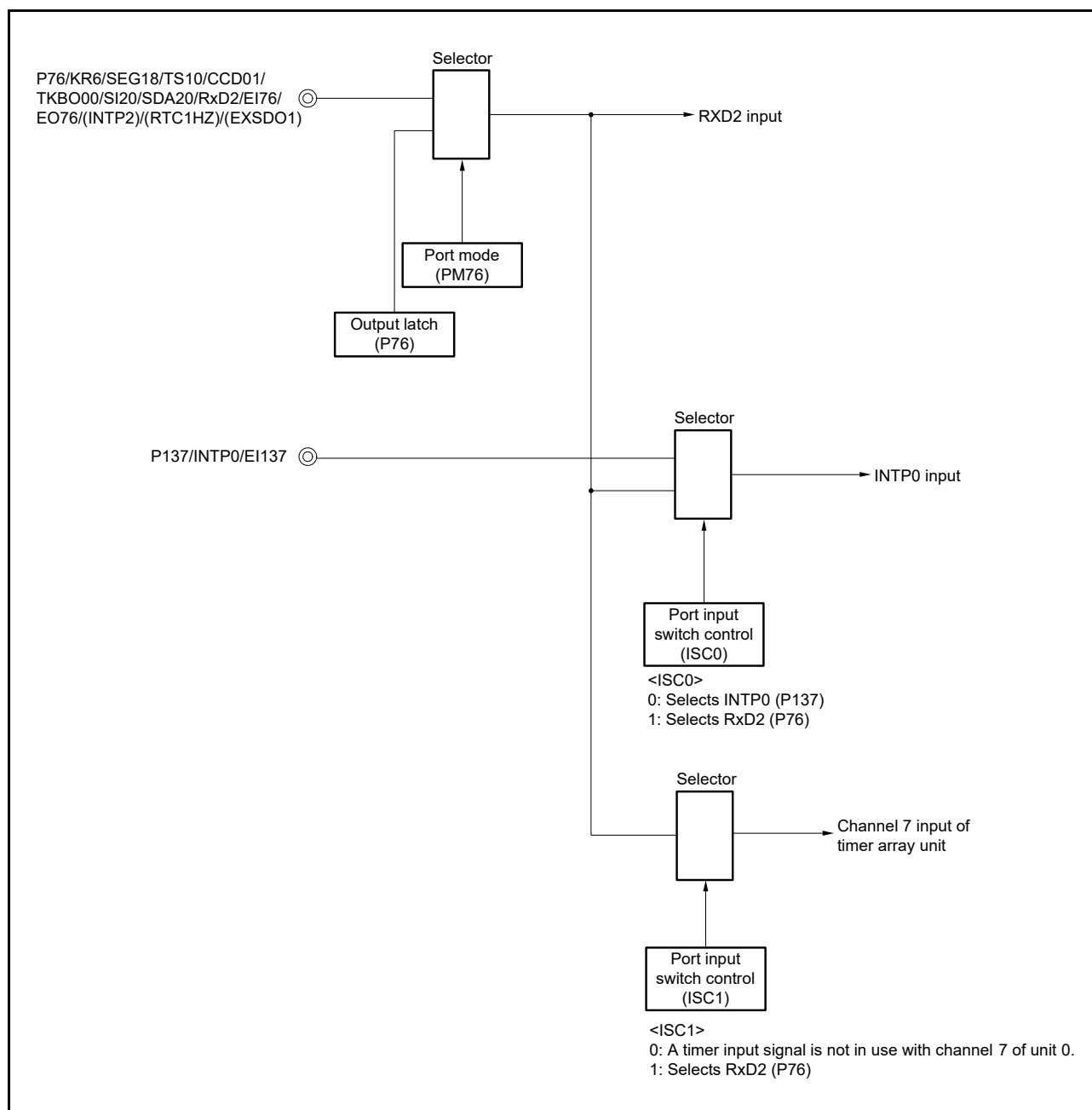
Figure 19 - 99 Flowchart of LIN Reception



Note This is only required if the LIN bus is in sleep mode.

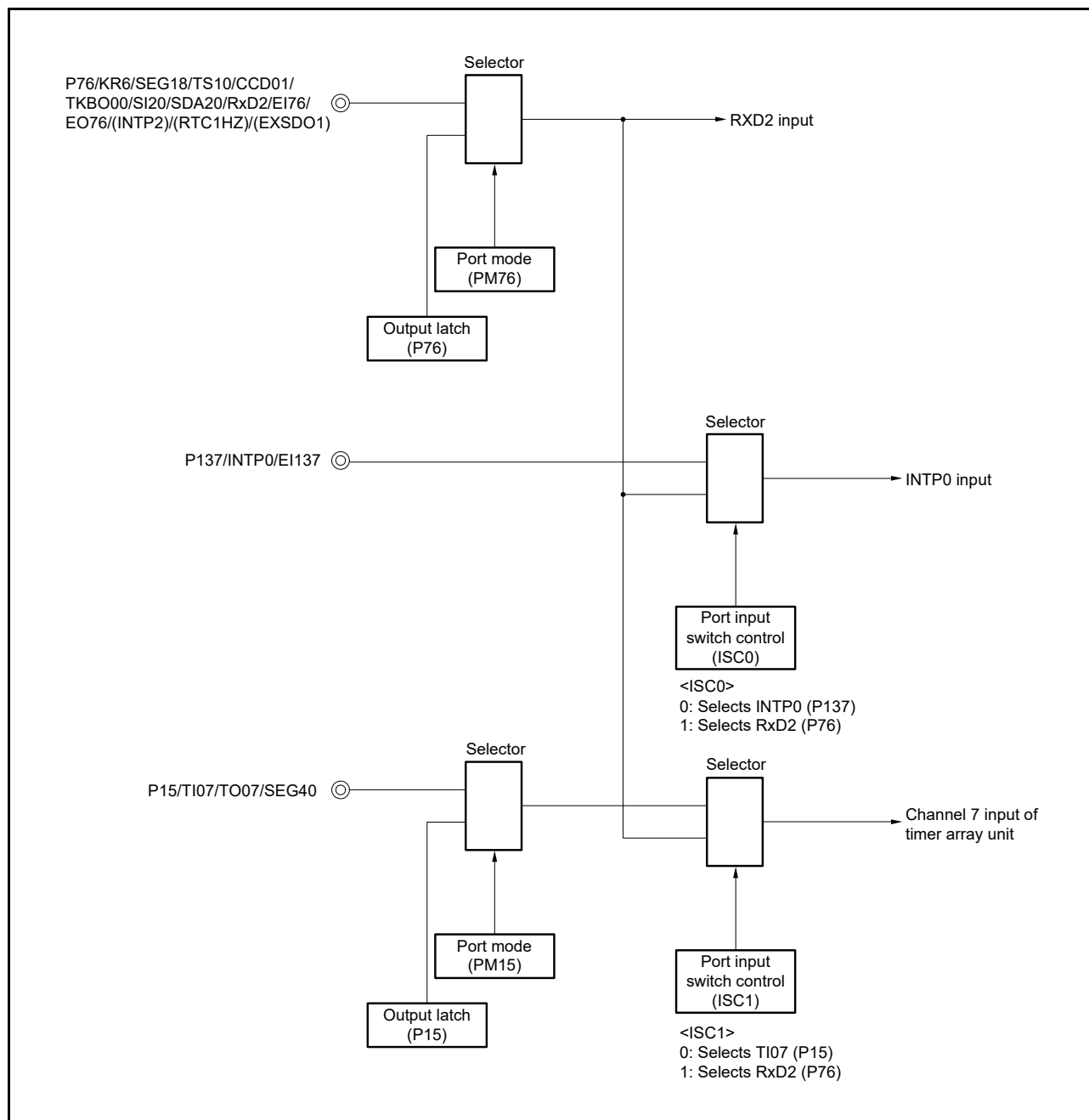
Figure 19 - 100 and **Figure 19 - 101** show the configuration of ports used for LIN reception. The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the 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 19 - 100 Port Configuration for LIN Reception for the 44- to 52-pin products with no function assigned to the TI07 pin



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 19 - 21**.)

Figure 19 - 101 Port Configuration for LIN Reception for the 64- to 100-pin products with a function assigned to the TI07 pin



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 19 - 21 Format of Input Switch Control Register (ISC)**.)

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)

19.8 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) 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

* The following functions are 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 **19.8.3 Data Reception, 2. Processing flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

The channel supporting simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) is channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1.

<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	(CSI01)		(IIC01)
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	—	UART2 (supporting LIN-bus)	—
	1	CSI21		IIC21
	2	—	—	(IIC30)
	3	—		—

<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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	—	—	IIC30
	3	—		—

<52-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	—		—

<64-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<80-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	(CSI31)		(IIC31)

<100-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	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting LIN-bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see **4.3.10 Peripheral I/O redirection registers (PIORx)**.

Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) performs the following four types of communication operations.

- Address field transmission (See **19.8.1.**)
- Data transmission (See **19.8.2.**)
- Data reception (See **19.8.3.**)
- Stop condition generation (See **19.8.4.**)

19.8.1 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	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL10, SDA10 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}	SCL30, SDA30 ^{Note 1}	SCL31, SDA31 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	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} : Operation 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 proceed communication via simplified I²C, set the port output mode register (POMxx) to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode. 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, IIC10, IIC20, IIC30, and IIC31, also set the POMxx register to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). 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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

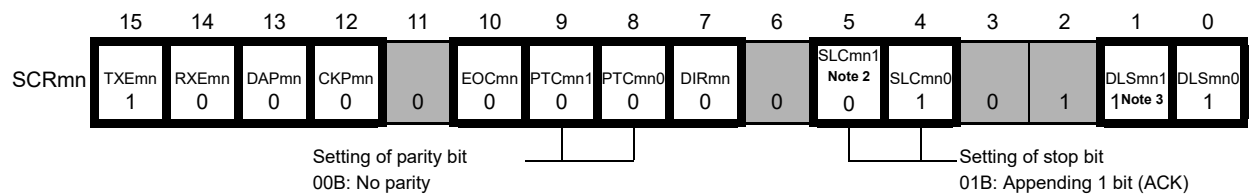
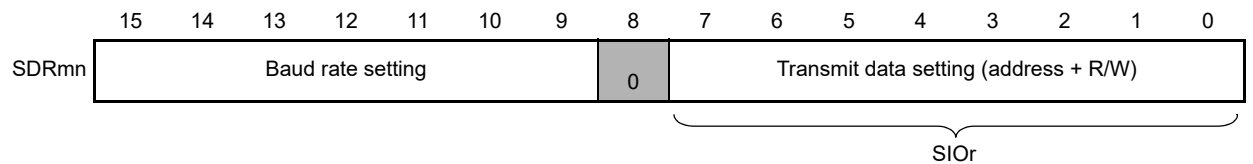
1. Register setting

Figure 19 - 102 Example of Contents of Registers for Address Field Transmission of Simplified I²C
(IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

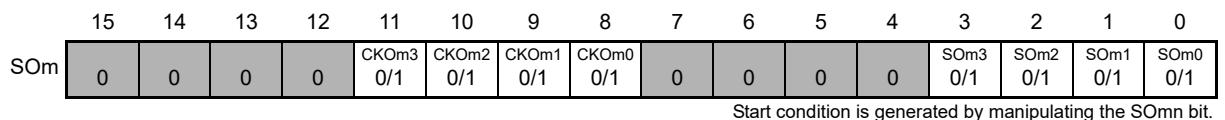
a) Serial mode register mn (SMRmn)



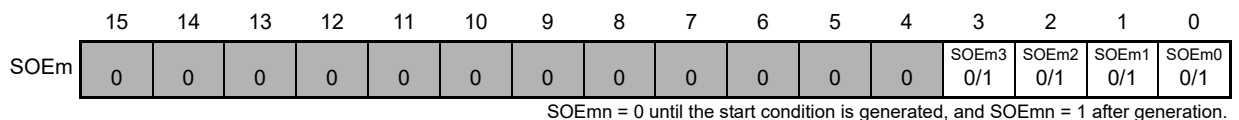
b) Serial communication operation setting register mn (SCRmn)

c) Serial data register mn (SDRmn) (lower 8 bits: SIO_r)

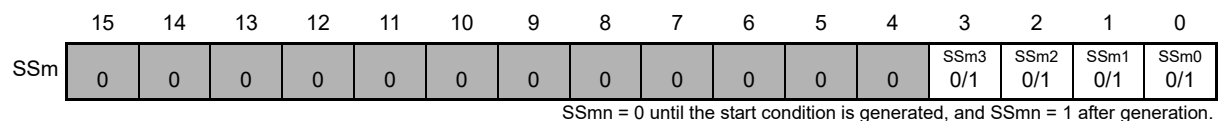
d) Serial output register m (SOM)



e) Serial output enable register m (SOEm)



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.

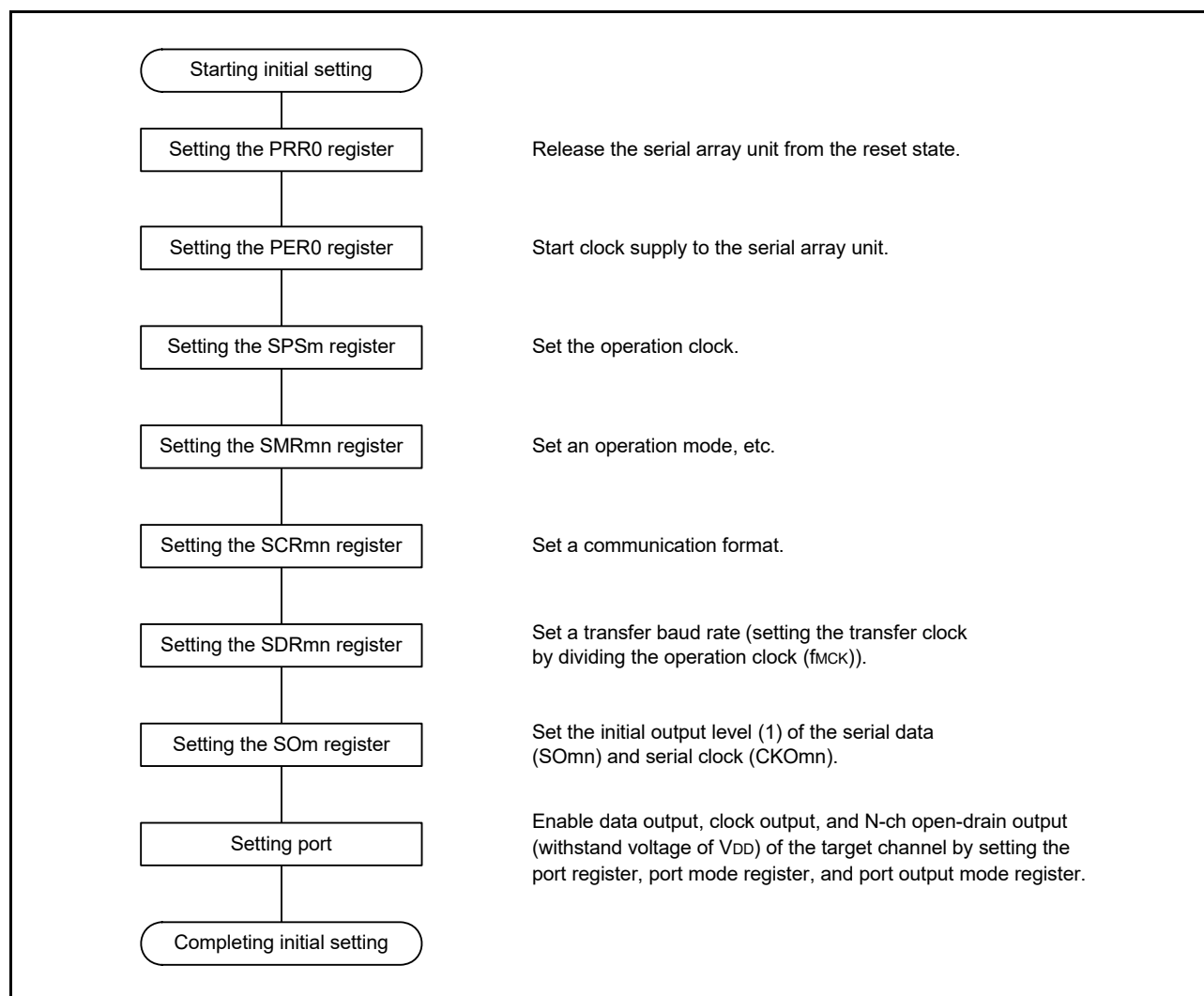
**Note 1.** This bit is only present in the SMR00, SMR03, SMR11, and SMR13 registers.**Note 2.** This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.**Note 3.** This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.

(Remarks continue on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

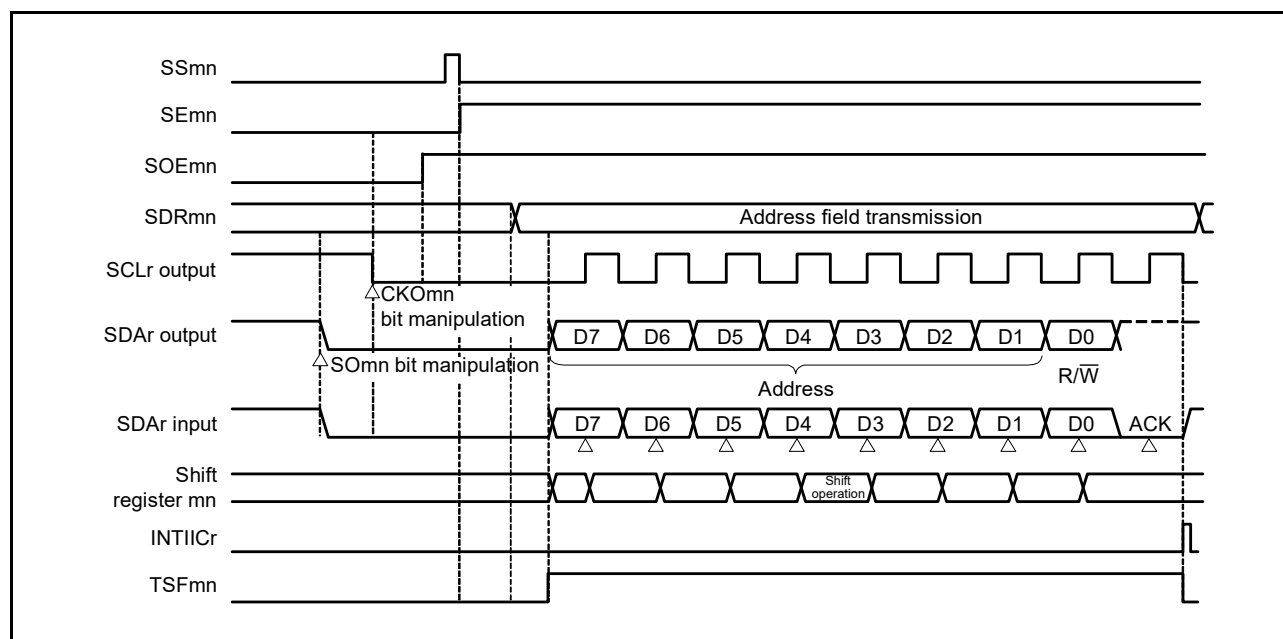
Remark 2. ☐: Setting is fixed in the IIC mode, ☐: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

2. Operation procedure

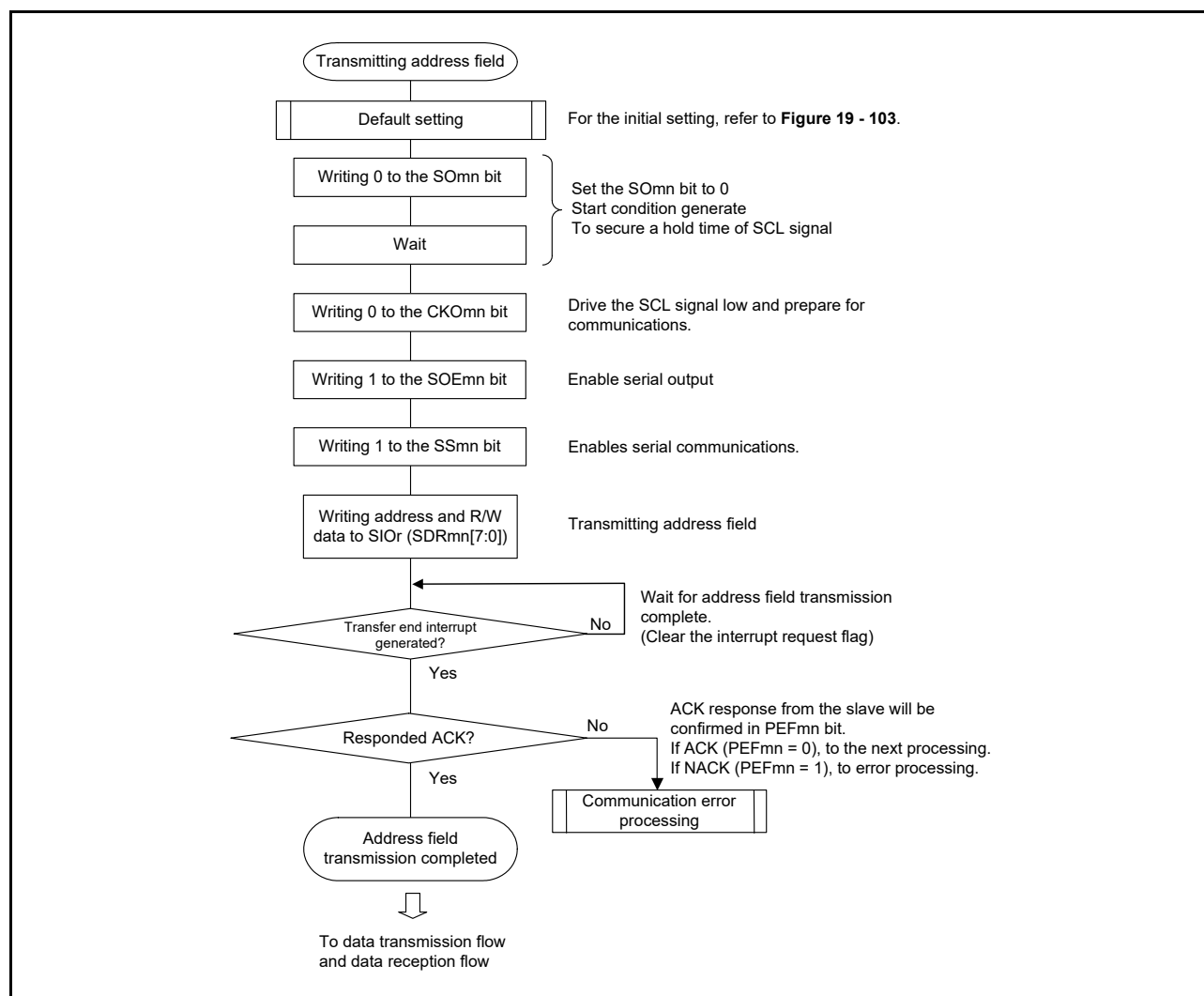
Figure 19 - 103 Initial Setting Procedure for Simplified I²C Address Field Transmission

3. Processing flow

Figure 19 - 104 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

Figure 19 - 105 Flowchart of Simplified I²C Address Field Transmission

19.8.2 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	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL10, SDA10 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}	SCL30, SDA30 ^{Note 1}	SCL31, SDA31 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	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] (SDR _{mn} [15:9] = 1 or more) f _{MCK} : Operation 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 perform communication via simplified I²C, set the port output mode register (POMxx) to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode. 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, IIC10, IIC20, IIC30, and IIC31, also set the POMxx register to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). 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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

1. Register setting

Figure 19 - 106 Example of Contents of Registers for Data Transmission 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	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 settingNote 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 0/1 Note 5	CKOm2 0/1 Note 5	CKOm1 0/1 Note 5	CKOm0 0/1 Note 5	0	0	0	0	SOM3 0/1 Note 5	SOM2 0/1 Note 5	SOM1 0/1 Note 5	SOM0 0/1 Note 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 1	SOEm1 1	SOEm0 1

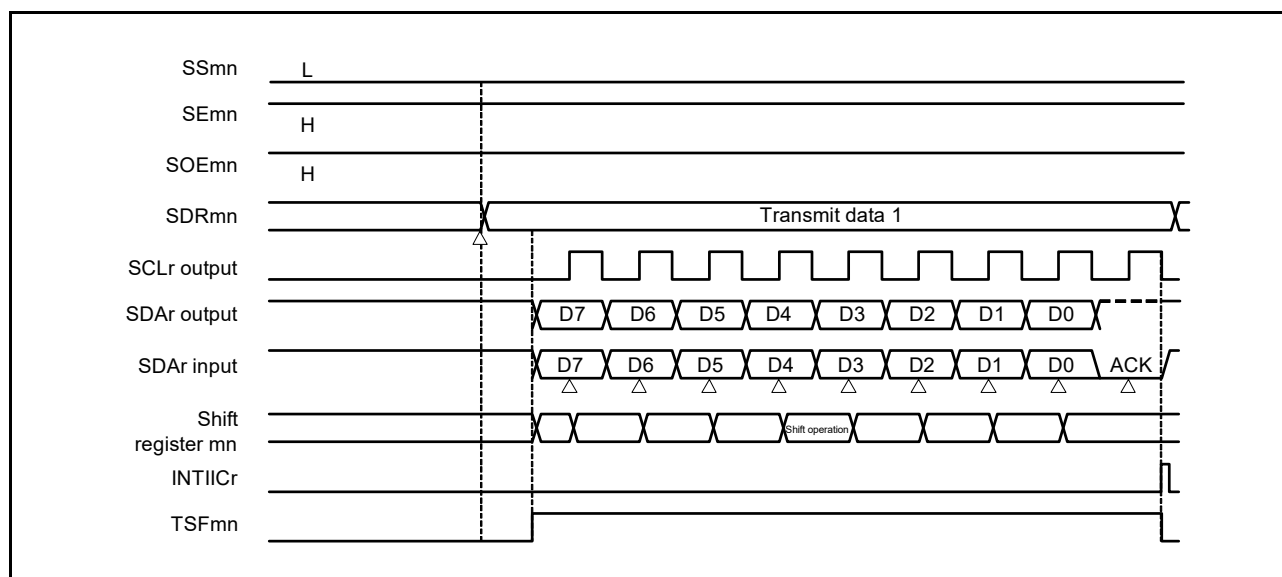
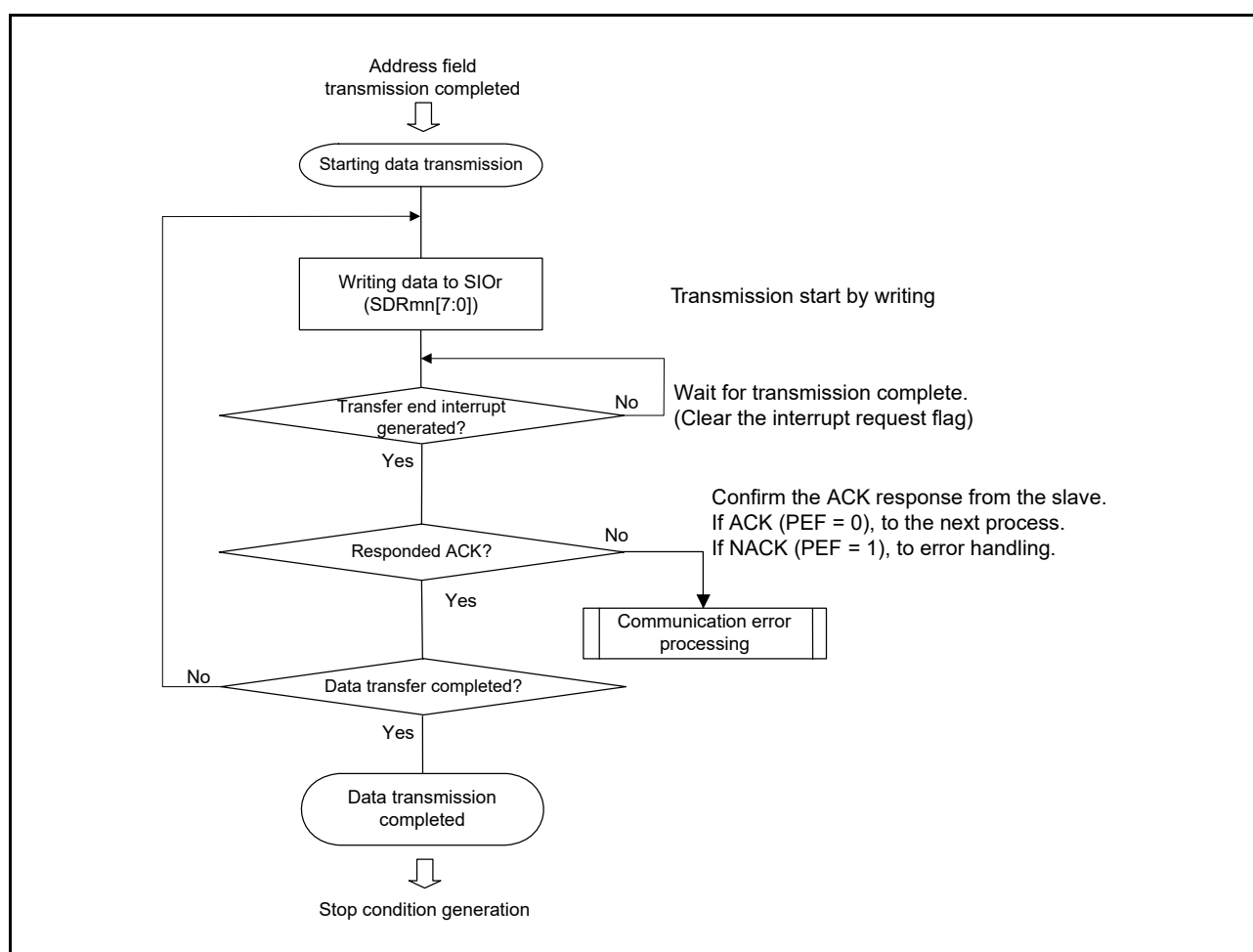
f) Serial channel start register m (SSm): Do not manipulate this register during data transmission/reception.

	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 0/1	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 registers.**Note 2.** This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.**Note 3.** This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.**Note 4.** This setting is not required because it is completed in address field transmission.**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 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13**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 19 - 107 Timing Chart of Data Transmission

Figure 19 - 108 Flowchart of Simplified I²C Data Transmission

19.8.3 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	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00 ^{Note 1}	SCL01, SDA01 ^{Note 1}	SCL10, SDA10 ^{Note 1}	SCL11, SDA11 ^{Note 1}	SCL20, SDA20 ^{Note 1}	SCL21, SDA21 ^{Note 1}	SCL30, SDA30 ^{Note 1}	SCL31, SDA31 ^{Note 1}
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	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} : Operation 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 perform communication via simplified I²C, set the port output mode register (POMxx) to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode. 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, IIC10, IIC20, IIC30, and IIC31, also set the POMxx register to 1 to specify N-ch open-drain output (withstand voltage of V_{DD}) mode for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). 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 42 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

1. Register setting

Figure 19 - 109 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	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 0	RXEmn 1	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): The lower 8 bits (SIO_r) are only valid during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate settingNote 4							0	Dummy transmit data setting (FFH)							
									SIO _r							

d) Serial output register m (SO_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
SO _m	0	0	0	0	CKOm3 0/1 Note 5	CKOm2 0/1 Note 5	CKOm1 0/1 Note 5	CKOm0 0/1 Note 5	0	0	0	0	SOm3 0/1 Note 5	SOm2 0/1 Note 5	SOm1 0/1 Note 5	SOm0 0/1 Note 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 0/1	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

f) Serial channel start register m (SSm): Do not manipulate this register during data transmission/reception.

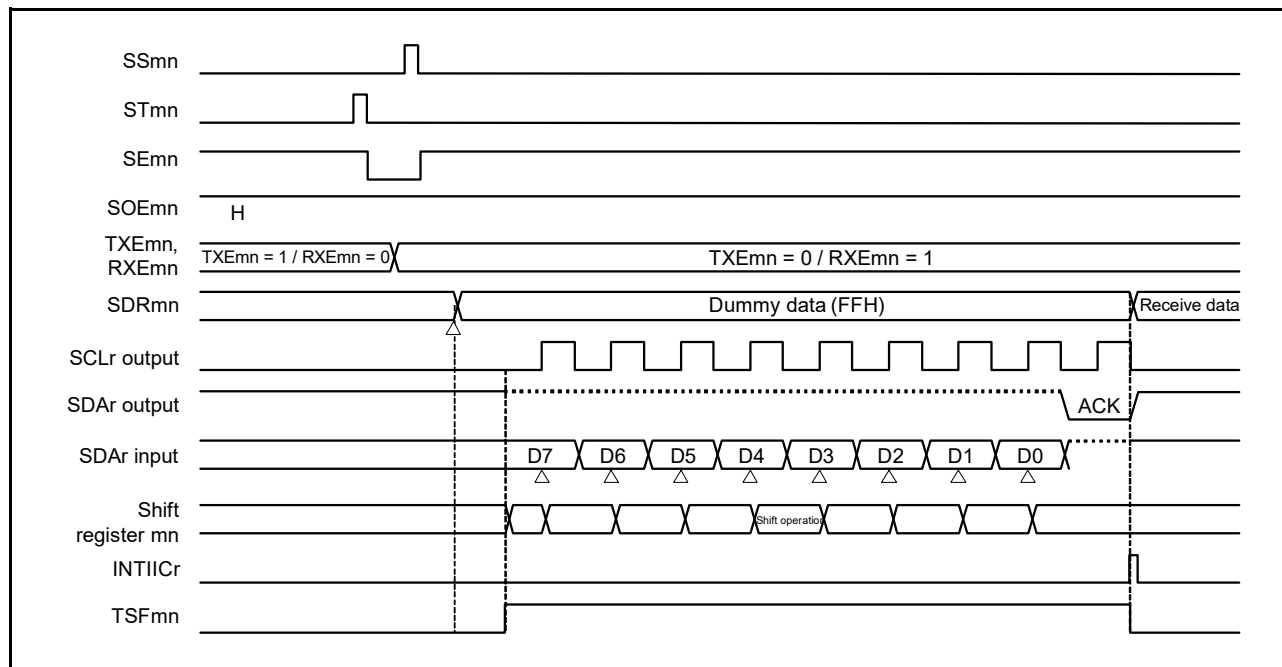
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 registers.**Note 2.** This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.**Note 3.** This bit is only present in the SCR00, SCR01, SCR10, and SCR11 registers. This bit is fixed to 1 for the other registers.**Note 4.** This setting is not required because it is completed in address field transmission.**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 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13**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

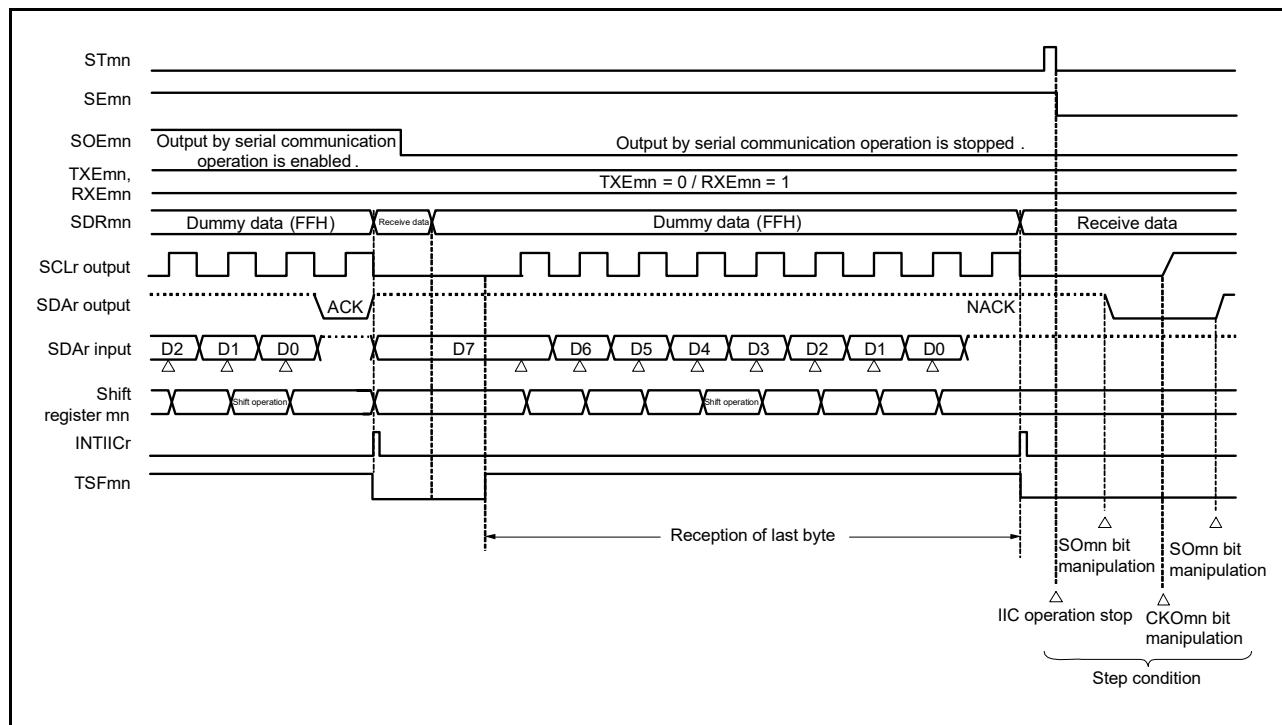
2. Processing flow

Figure 19 - 110 Timing Chart of Data Reception

a) When starting data reception

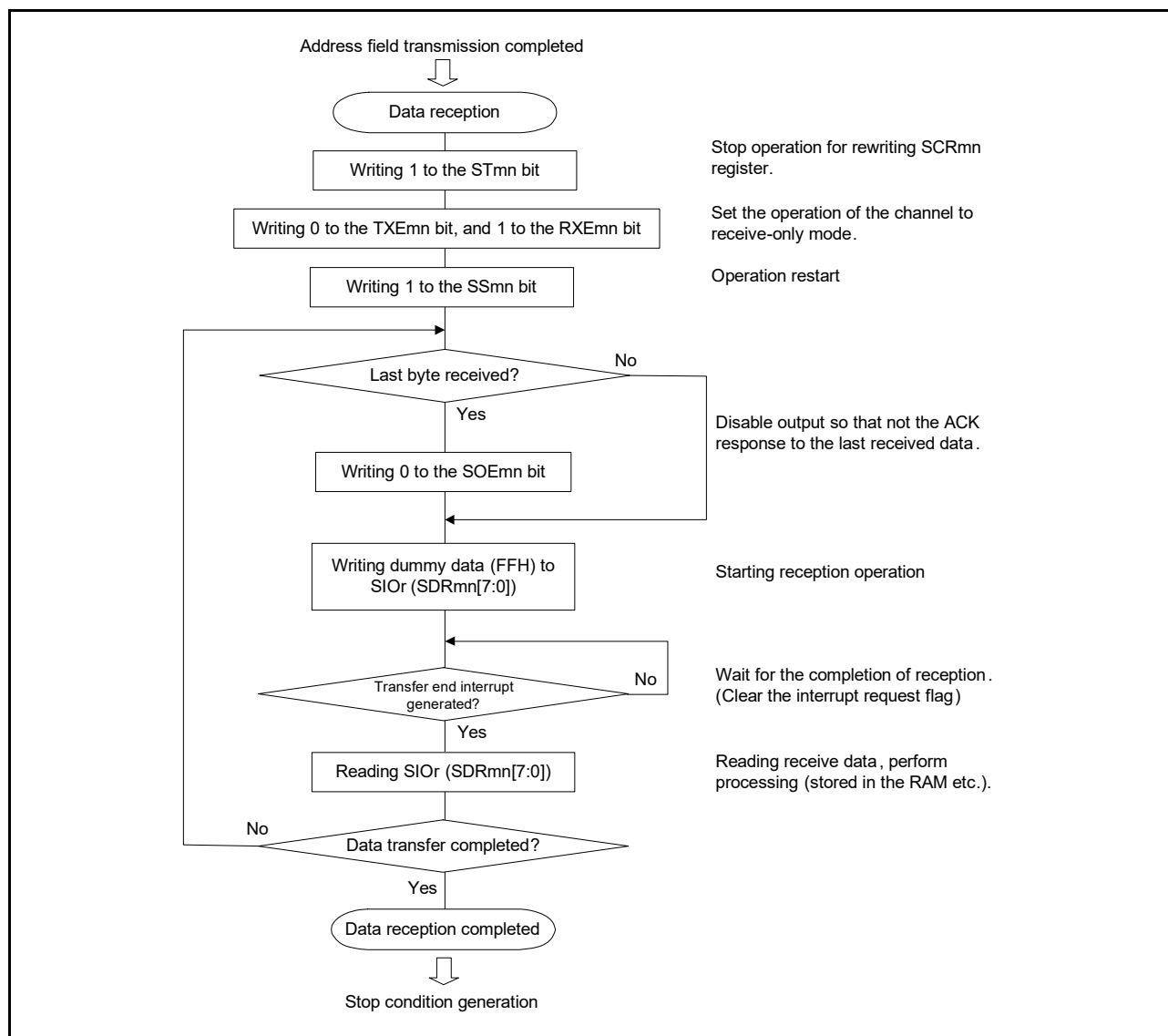


b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

Figure 19 - 111 Flowchart of Data Reception



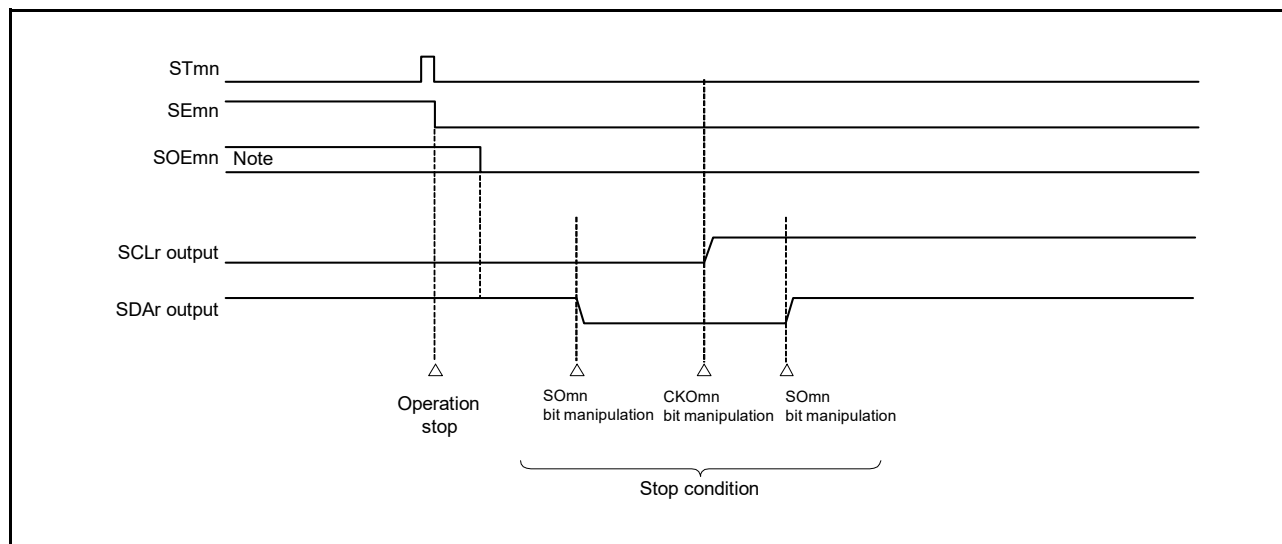
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

19.8.4 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.

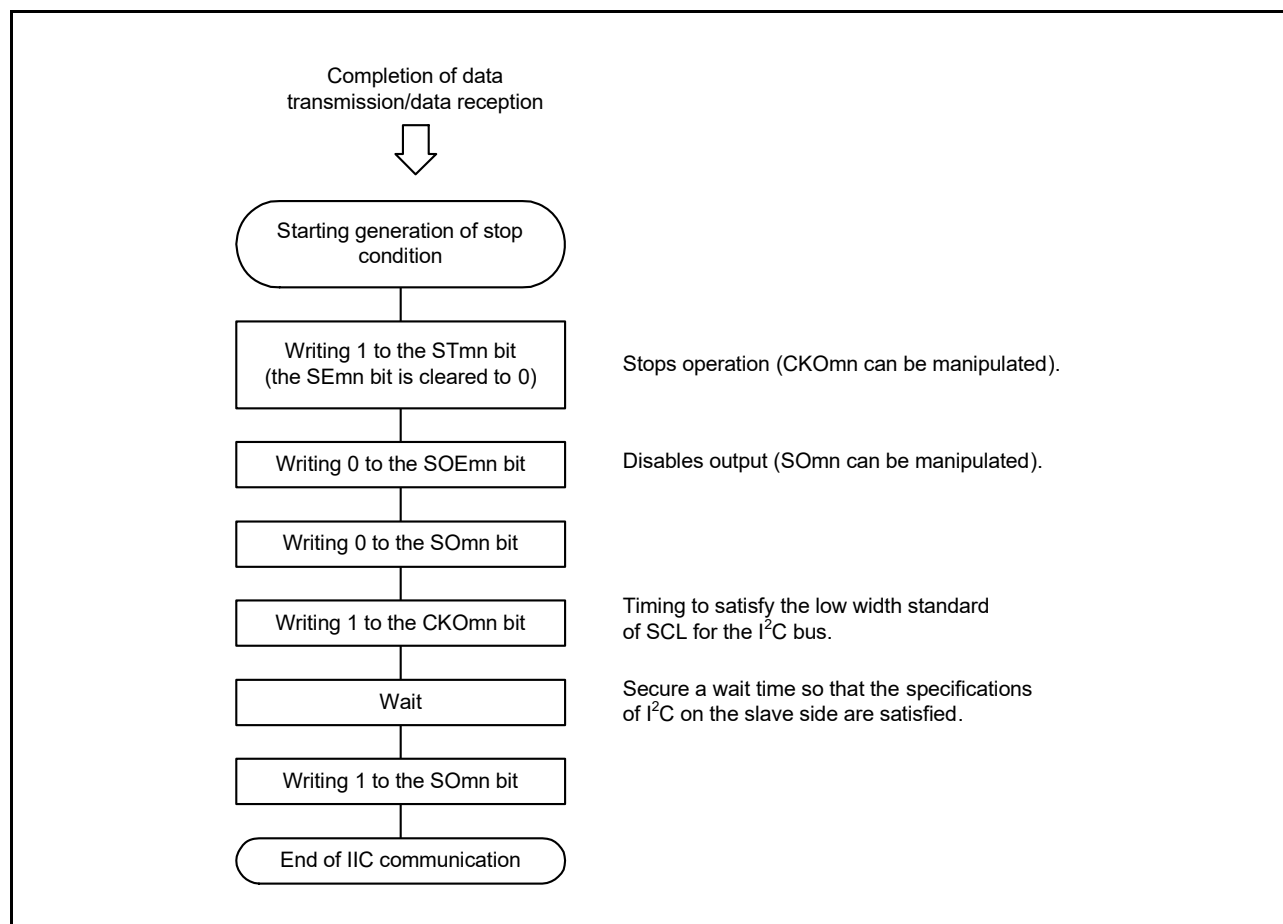
1. Processing flow

Figure 19 - 112 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 19 - 113 Flowchart of Stop Condition Generation



19.8.5 Calculating Transfer Rate

The transfer rate for simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication can be calculated by the following expressions.

$$(\text{Transfer rate}) = \{\text{Operation clock (fMCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

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 to 3), mn = 00 to 03, 10 to 13

The operation clock (fMCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 19 - 9 Selection of Operation Clock For Simplified I²C

SMRmn Register	SPSm Register								Operation 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 to 3), mn = 00 to 03, 10 to 13

Here is an example of setting an I²C transfer rate where $f_{MCK} = f_{CLK} = 32$ MHz.

I ² C Transfer Mode (Desired Transfer Rate)	fCLK = 32 MHz			
	Operation 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%.

19.8.6 Procedure for Processing Errors that Occurred during Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) Communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication is described in **Table 19 - 10** and **Table 19 - 11**.

Table 19 - 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 19 - 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 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31),
mn = 00 to 03, 10 to 13

Section 20 Serial Interface IICA (IICA)

This product has two IICA interfaces.

20.1 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 20 - 1** shows a block diagram of serial interface IICA.

Remark n = 0, 1

Figure 20 - 1 Block Diagram of Serial Interface IICA0

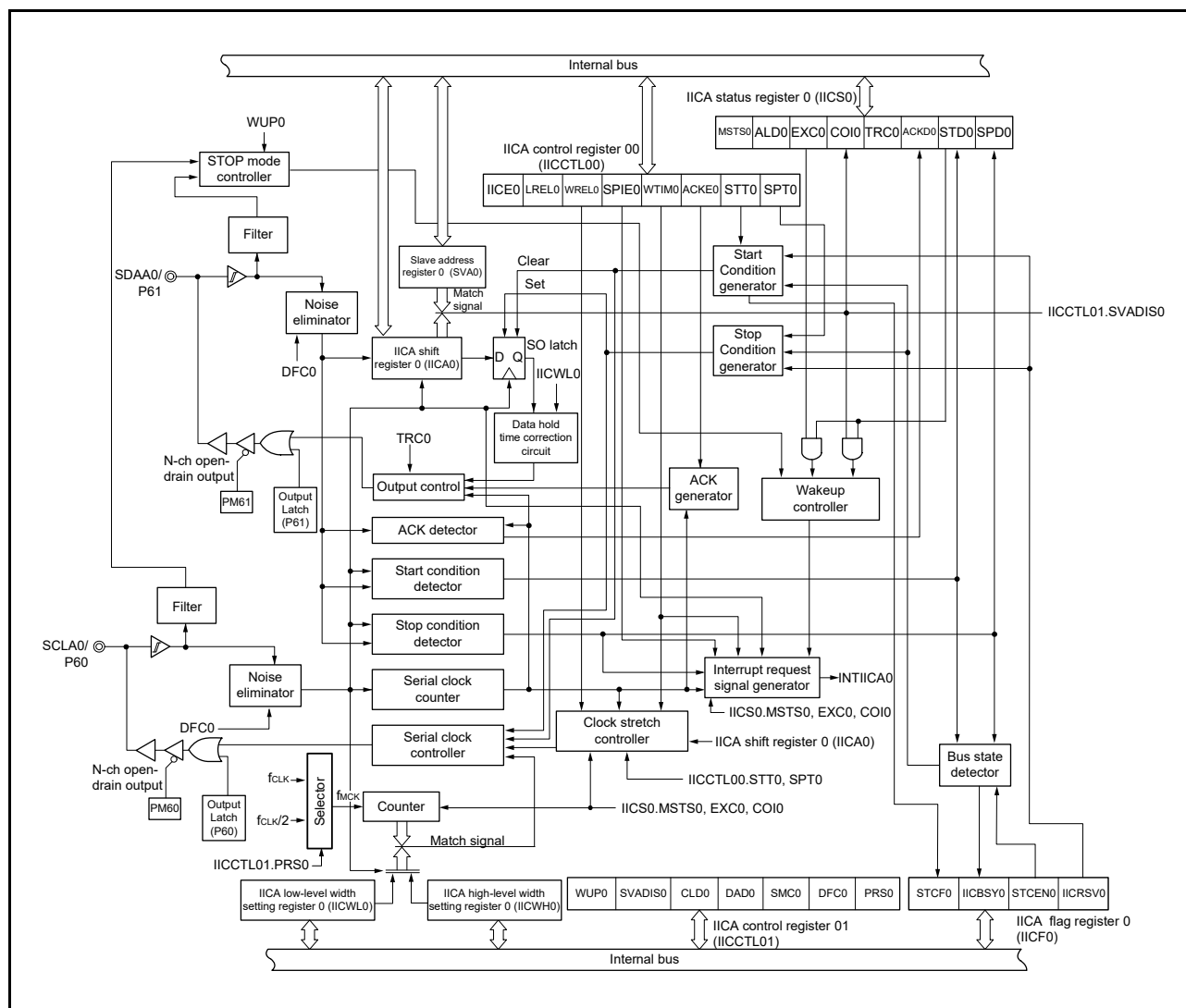
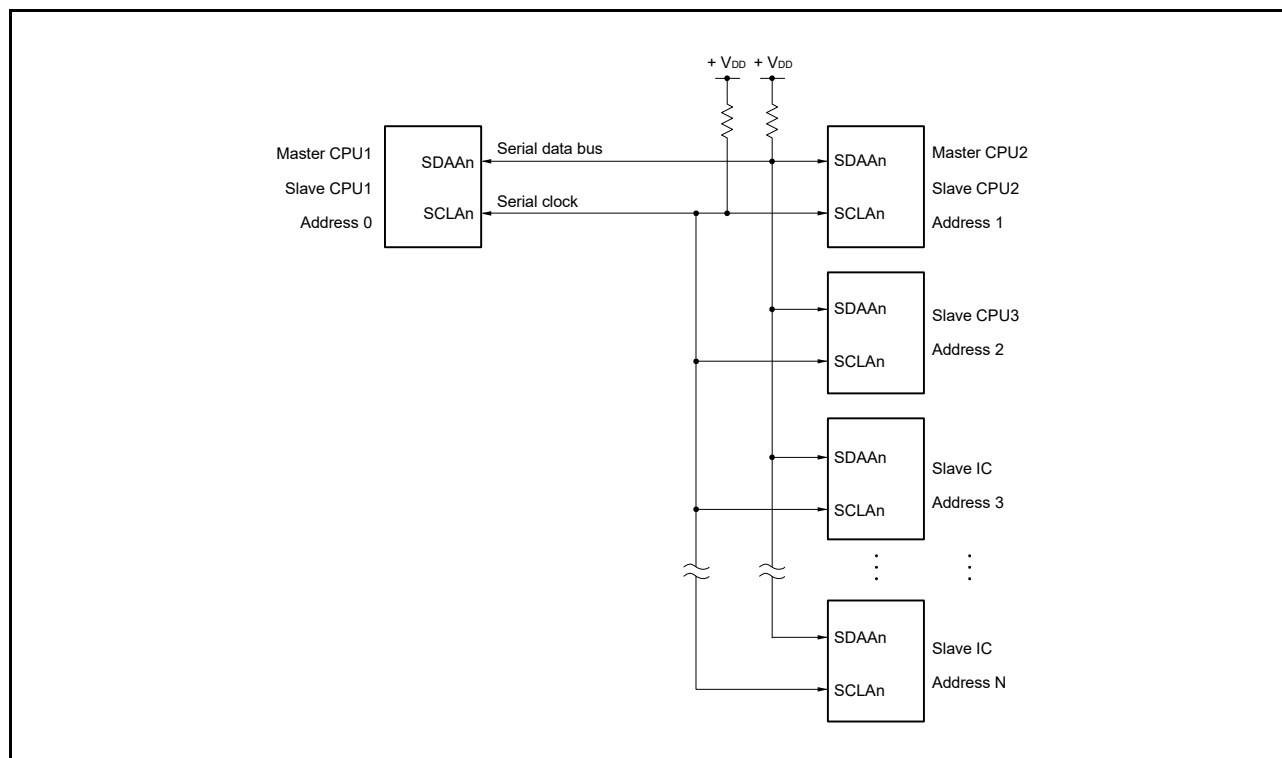


Figure 20 - 2 shows a serial bus configuration example.

Figure 20 - 2 Example of the Serial Bus Configuration Using the I²C Bus



Remark $n = 0, 1$

20.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 20 - 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 registers (PMxx) Port registers (Pxx) Port output mode registers (POMxx) Port mode control A registers (PMCAxx) Port mode control T registers (PMCTxx) Port mode control E registers (PMCExx) LCD port function registers (PFSEGx)

Remark n = 0, 1

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 20 - 3 Format of IICA Shift Register n (IICAn)

Address: FFF50H (IICA0), FFF54H (IICA1)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

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.

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 20 - 4 Format of Slave Address Register n (SVAn)

Address: F0234H (SVA0), F023CH (SVA1)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 Note

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)

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.

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. Note that the bus state cannot be detected immediately after the IICA operation has been enabled. Set the initial state of the bus state detector by the STCENn bit.

Remark 1. SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)

STTn bit: Bit 1 of IICA control register n0 (IICCTLn0)

SPTn bit: Bit 0 of IICA control register n0 (IICCTLn0)

STCFn bit: Bit 7 of IICA flag register n (IICFn)

IICBSYn bit: Bit 6 of IICA flag register n (IICFn)

STCENn bit: Bit 1 of IICA flag register n (IICFn)

IICRSVn bit: Bit 0 of IICA flag register n (IICFn)

Remark 2. n = 0, 1

20.3 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)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Remark n = 0, 1

20.3.1 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 IICA0 is to be used, be sure to set bit 4 (IICA0EN) of this register to 1. When serial interface IICA1 is to be used, be sure to set bit 6 (IICA1EN) 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 20 - 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	IICA1EN	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. Be sure to set the following registers while the setting of the IICAnEN bit is 1 when making the serial interface IICA settings. If IICAnEN = 0, the values of the registers for controlling the serial interface IICA are cleared to 00H, and writing to them is ignored.

- 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)
- IICA shift register n (IICAn)
- Slave address register n (SVAn)

Note, however, that writing to the following registers is valid.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Caution 2. Be sure to set bit 1 to 0.

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, 1

20.3.2 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 6 (IICA1RES) and 4 (IICA0RES) to 1 to reset serial interfaces IICA1 and IICA0, respectively. 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 20 - 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	IICA1RES	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 Be sure to set bits 7 and 1 to 0.

Remark n = 0, 1

20.3.3 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, 1

Figure 20 - 7 Format of IICA Control Register n0 (IICCTLn0) (1/5)

Address: F0230H (IICCTL00), F0238H (IICCTL10)

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 to 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 the LRELn bit to 1 by using a 1-bit memory manipulation instruction immediately after enabling operation of I²C (IICEn = 1).

Remark n = 0, 1

Figure 20 - 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 • MSTn • 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"> <tr> <th>Condition for clearing (LRELn = 0)</th><th>Condition for setting (LRELn = 1)</th></tr> <tr> <td> • Automatically cleared after execution • Reset </td><td> • Set by instruction </td></tr> </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"> <tr> <th>Condition for clearing (WRELn = 0)</th><th>Condition for setting (WRELn = 1)</th></tr> <tr> <td> • Automatically cleared after execution • Reset </td><td> • Set by instruction </td></tr> </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"> <tr> <th>Condition for clearing (SPIEn = 0)</th><th>Condition for setting (SPIEn = 1)</th></tr> <tr> <td> • Cleared by instruction • Reset </td><td> • Set by instruction </td></tr> </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 the LRELn bit to 1 by using a 1-bit memory manipulation instruction immediately after enabling operation of I²C (IICEn = 1).

Remark n = 0, 1

Figure 20 - 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, 1

Figure 20 - 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 to 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 to 1, the STTn bit is cleared and the STTn clear flag (STCFn) is set to 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 to 1, setting it to 1 again before the clear condition is met is not allowed.	
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, 1

Figure 20 - 7 Format of IICA Control Register n0 (IICCTLn0) (5/5)

SPTnNote	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 to 1, setting it to 1 again before the clear condition is met is not allowed.	
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, 1

20.3.4 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 (wakeup operation is stopped), 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 20 - 8 Format of IICA Status Register n (IICS_n) (1/4)

Address: FFF51H (IICS0), FFF55H (IICS1)

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 value indicates either that arbitration is not in progress or that the result of arbitration was a win.						
1		This value indicates that the result of arbitration was a loss. The MSTS _n bit is cleared when the ALD _n bit has this value.						
Condition for clearing (ALD _n = 0)					Condition for setting (ALD _n = 1)			
• Automatically cleared after the IICS_n register is readNote • 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, 1

Figure 20 - 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, 1

Figure 20 - 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 (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. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

Figure 20 - 8 Format of IICA Status Register n (IICS_n) (4/4)

ACKD _n	Detection of acknowledge (ACK)	
0	Acknowledge was not detected.	
1	Acknowledge was detected.	
Condition for clearing (ACKD _n = 0)		Condition for setting (ACKD _n = 1)
- When a stop condition is detected - At the rising edge of the next byte's first clock - Cleared by LREL_n = 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

STD _n	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 (STD _n = 0)		Condition for setting (STD _n = 1)
- When a stop condition is detected - At the rising edge of the next byte's first clock following address transfer - 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 detected

SPD _n	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 (SPD _n = 0)		Condition for setting (SPD _n = 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 WUP_n 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. LREL_n: Bit 6 of IICA control register n0 (IICCTL_n0)

IICEn: Bit 7 of IICA control register n0 (IICCTL_n0)

Remark 2. n = 0, 1

20.3.5 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 20 - 9 Format of IICA Flag Register n (IICFn) (1/2)

Address: FFF52H (IICF0), FFF56H (IICF1)

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	

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)	
- Cleared by instruction - When a start condition is detected - Reset	
Condition for setting (STCENn = 1)	
Set by instruction	

Figure 20 - 9 Format of IICA Flag Register n (IICFn) (2/2)

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

Note Bits 7 and 6 are read-only.

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)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

20.3.6 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 20 - 10 Format of IICA Control Register n1 (IICCTLn1) (1/2)

Address: F0231H (IICCTL01), F0239H (IICCTL11)

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 fMCK after setting the WUPn bit to 1 (see Figure 20 - 22 Flow When Setting WUPn = 1). Clear the WUPn bit to 0 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 clearing the WUPn bit to 0. (The interface must be released from the clock stretch state and transmit data must be written after the WUPn bit has been cleared to 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.					
<table border="1"> <tr> <th>Condition for clearing (WUPn = 0)</th><th>Condition for setting (WUPn = 1)</th></tr> <tr> <td>• Cleared by instruction (after address match, address reception with the all address match function enabled, or extension code reception)</td><td>• Set by instruction (when the MSTSn, EXCn, and COIn bits are 0, and the STDn bit also 0 (communication not entered))^{Note 2}</td></tr> </table>		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}
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}				

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 20.5.11 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.				
<table border="1"> <tr> <th>Condition for clearing (CLDn = 0)</th><th>Condition for setting (CLDn = 1)</th></tr> <tr> <td> - When the SCLAn pin is at low level - When IICEn = 0 (operation stop) - Reset </td><td> - When the SCLAn pin is at high level </td></tr> </table>		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
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				

Figure 20 - 10 Format of IICA Control Register n1 (IICCTLn1) (2/2)

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

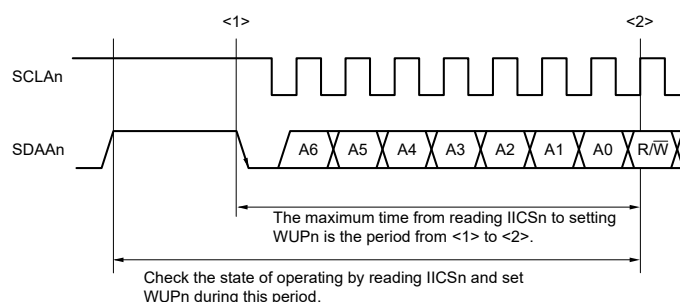
SMCn	Operation mode switching	
0	Operates in standard mode (fastest transfer rate: 100 kbps).	
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).	

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 to 1 or cleared to 0.		

PRSn	Control of IICA operation clock (fMCK)	
0	Selects fCLK (1 MHz ≤ fCLK ≤ 20 MHz).	
1	Selects fCLK/2 (20 MHz < fCLK).	

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.



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, 1

20.3.7 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 **20.4.2 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 20 - 11 Format of IICA Low-level Width Setting Register n (IICWLn)

Address: F0232H (IICWL0), F023AH (IICWL1)
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

20.3.8 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 20 - 12 Format of IICA High-level Width Setting Register n (IICWHn)

Address: F0233H (IICWH0), F023BH (IICWH1)
 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 **20.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

Remark 2. n = 0, 1

20.3.9 Registers for controlling the port functions multiplexed with the IICA inputs and outputs

Set the following registers for controlling the port functions multiplexed with the serial interface IICA inputs and outputs.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

20.4 I²C Bus Mode Functions

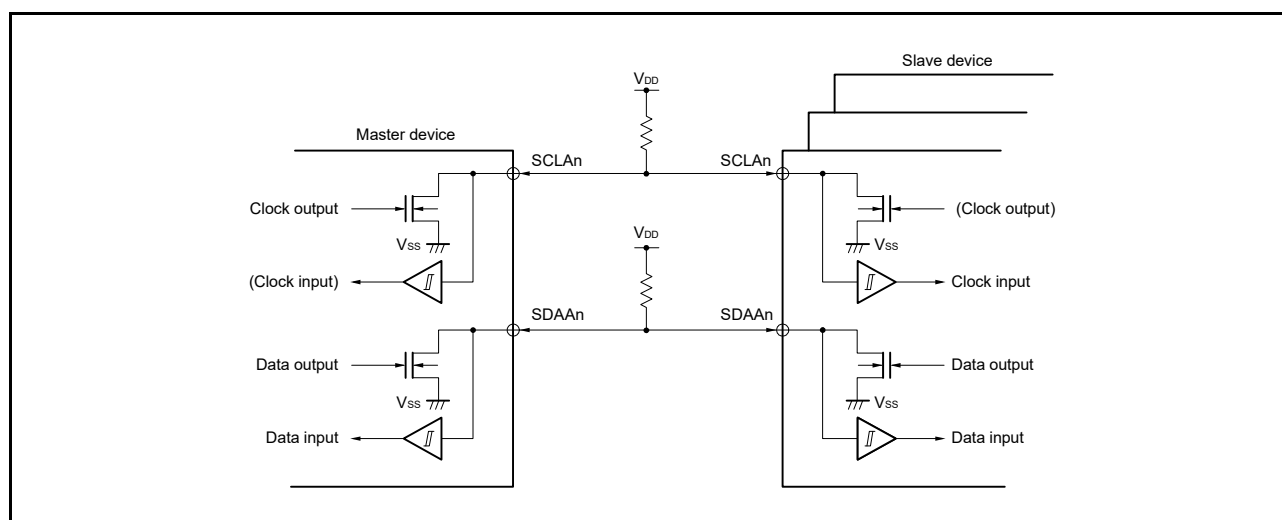
20.4.1 Pin Configuration

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 20 - 13 Pin Configuration Diagram



Remark n = 0, 1

20.4.2 Setting transfer clock by using IICWLn and IICWHn registers

1. Setting transfer clock on master side

$$\text{Transfer clock} = \frac{\text{fMCK}}{(\text{IICWLn} + 1) + (\text{IICWHn} + 1) + \text{fMCK}(\text{tR} + \text{tF})}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.

(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWLn} + 1 = \frac{0.52}{\text{Transfer clock}} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = \left(\frac{0.48}{\text{Transfer clock}} - \text{tR} - \text{tF} \right) \times \text{fMCK}$$

- When the normal mode

$$\text{IICWLn} + 1 = \frac{0.47}{\text{Transfer clock}} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = \left(\frac{0.53}{\text{Transfer clock}} - \text{tR} - \text{tF} \right) \times \text{fMCK}$$

- When the fast mode plus

$$\text{IICWLn} + 1 = \frac{0.50}{\text{Transfer clock}} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = \left(\frac{0.50}{\text{Transfer clock}} - \text{tR} - \text{tF} \right) \times \text{fMCK}$$

2. Setting IICWLn and IICWHn registers on slave side

(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWLn} + 1 = 1.3 \mu\text{s} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = (1.2 \mu\text{s} - \text{tR} - \text{tF}) \times \text{fMCK}$$

- When the normal mode

$$\text{IICWLn} + 1 = 4.7 \mu\text{s} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = (5.3 \mu\text{s} - \text{tR} - \text{tF}) \times \text{fMCK}$$

- When the fast mode plus

$$\text{IICWLn} + 1 = 0.50 \mu\text{s} \times \text{fMCK}$$

$$\text{IICWHn} + 1 = (0.50 \mu\text{s} - \text{tR} - \text{tF}) \times \text{fMCK}$$

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.)

(Remarks continue 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

fMCK: IICA operation clock frequency

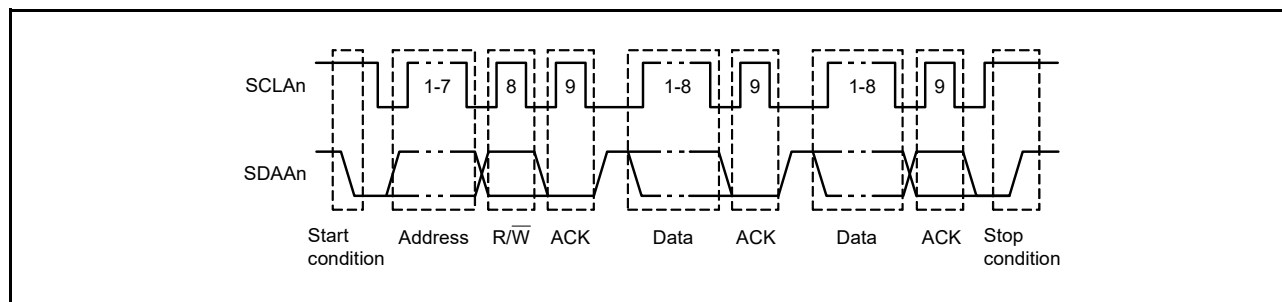
Remark 3. n = 0, 1

20.5 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 20 - 14 shows the transfer timing for the “start condition”, “address”, “data”, and “stop condition” output via the I²C bus's serial data bus.

Figure 20 - 14 I²C Bus Serial Data Transfer Timing

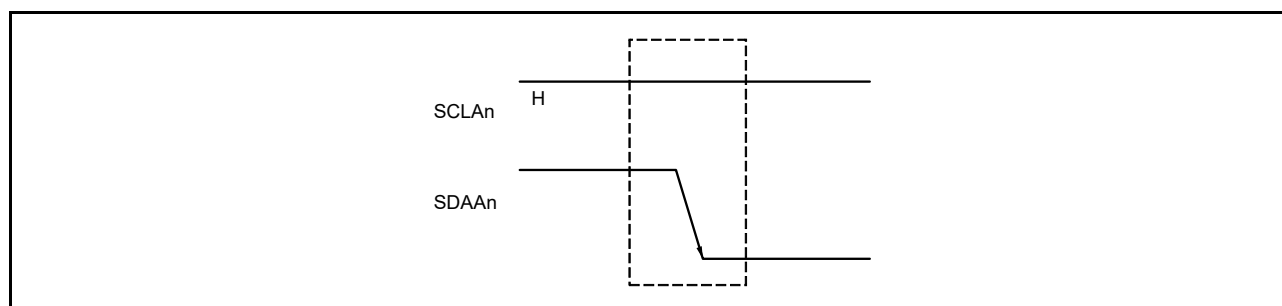


The master device generates the start condition, slave address, and stop condition. The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data). The serial clock (SCLAn) is continuously output by the master device. However, 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.

20.5.1 Start Conditions

When the SCLAn pin is at high level, changing the SDAAn pin from the high level to the low level generates a start 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 20 - 15 Start Conditions



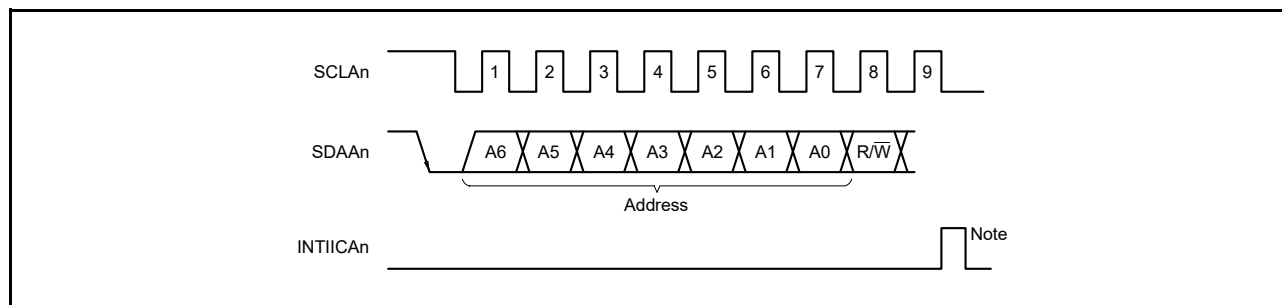
A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set to 1.

Remark n = 0, 1

20.5.2 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 20 - 16 Addresses



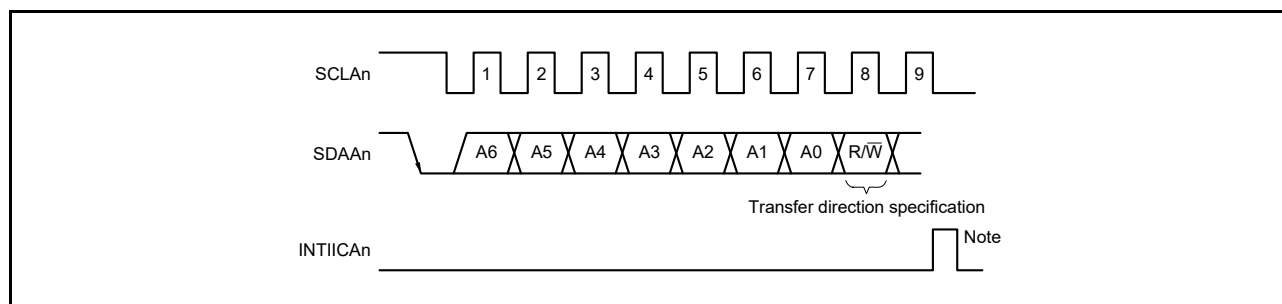
Note INTIICAn is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **20.5.3 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.

20.5.3 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 20 - 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 $n = 0, 1$

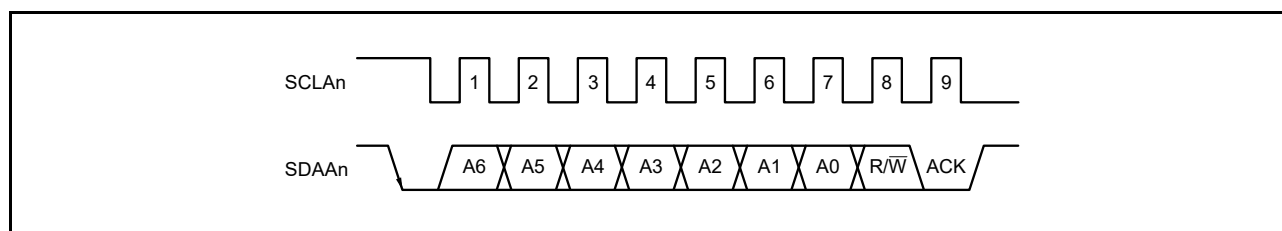
20.5.4 Acknowledge (ACK)

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 20 - 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.

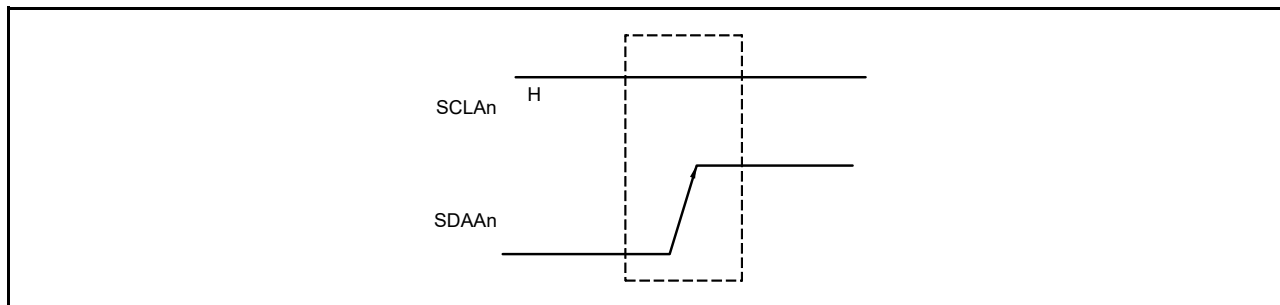
- When 8th cycle clock stretching is selected (bit 3 (WTIMn) of the IICCTLn0 register = 0):
By setting the ACKEn bit to 1 before release from the clock stretch state, ACK is generated at the falling edge of the eighth clock cycle of the SCLAn pin.
- When 9th cycle clock stretching is selected (bit 3 (WTIMn) of the IICCTLn0 register = 1):
ACK is generated if the ACKEn bit is set to 1 in advance.

Remark n = 0, 1

20.5.5 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 20 - 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.

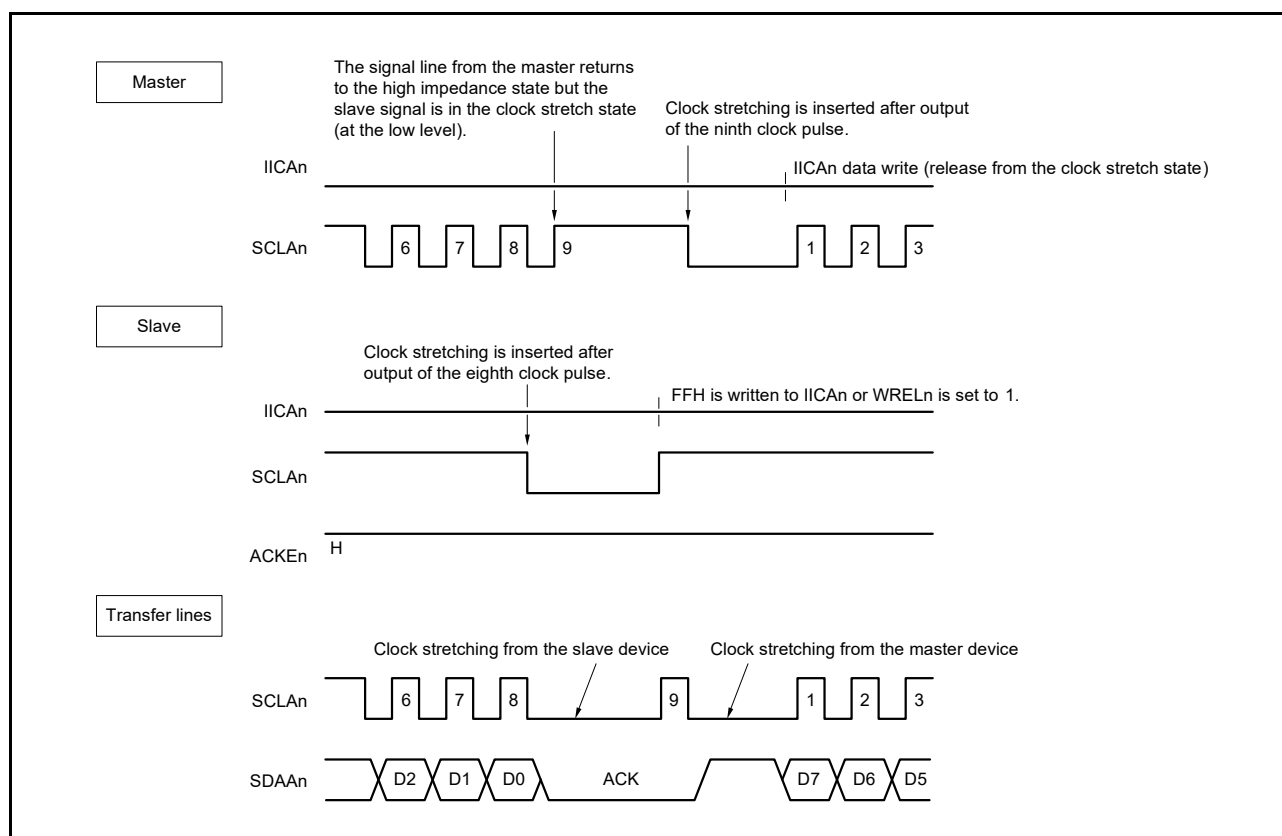
Remark n = 0, 1

20.5.6 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 20 - 20 Clock Stretching (1/2)

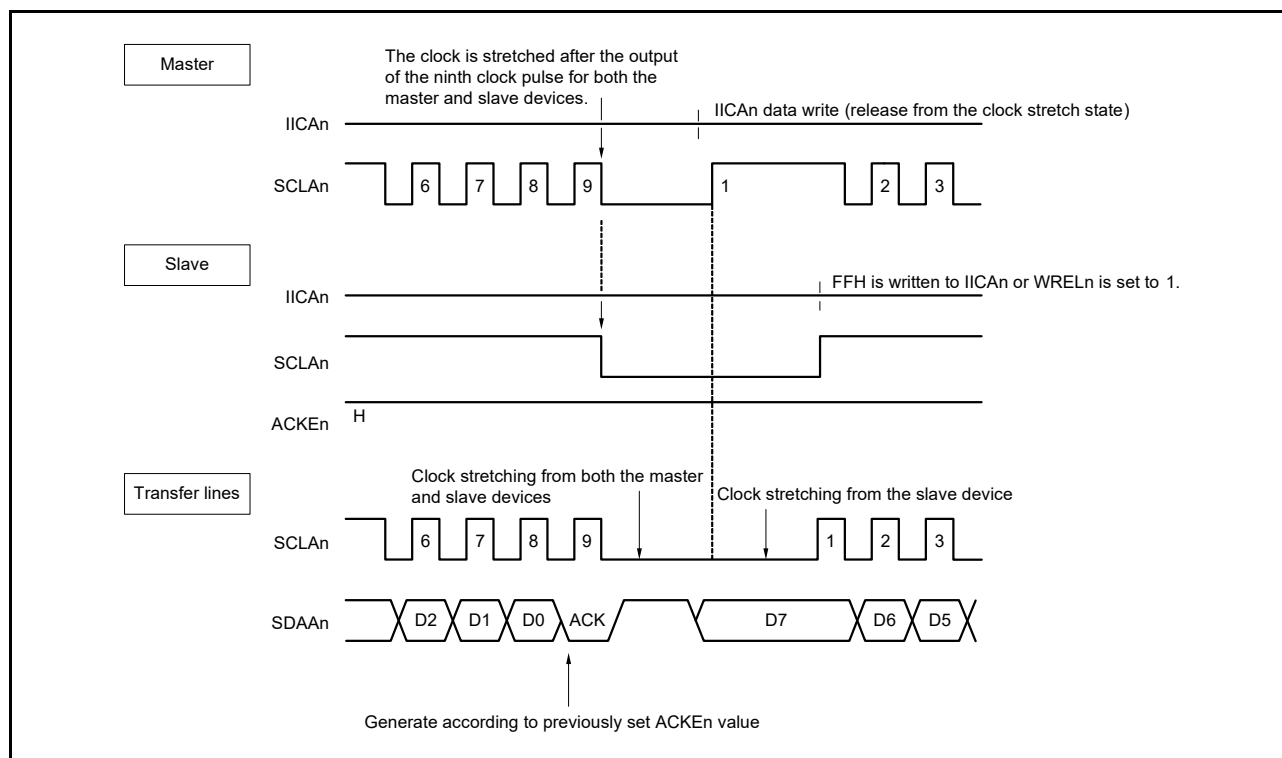
1. When clock stretching is set for the ninth and eighth clock cycles for the master and slave devices, respectively (master: transmission, slave: reception, and ACKEn = 1)



Remark n = 0, 1

Figure 20 - 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, 1

20.5.7 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}

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.

Note Master only

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, 1

20.5.8 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 20 - 2**.

Table 20 - 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, 3	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.
- Note 3.** When the WTIMn bit is 1, a slave device, which has received an extension code or has received an address with the all address match function enabled, generates INTIICAn and clock stretching at the falling edge of the ninth clock as well, in addition to the falling edge of the eighth clock.

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.

- During address transmission/reception
 - Slave device operation: The timing of the interrupt and clock stretching depends on the conditions described in **Note 1, Note 2, and Note 3** above.
 - 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.
- During data reception
 - Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.
- During data transmission
 - Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.

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**

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).

Note Master only

Remark n = 0, 1

20.5.9 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.

20.5.10 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, 1

20.5.11 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 20 - 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.

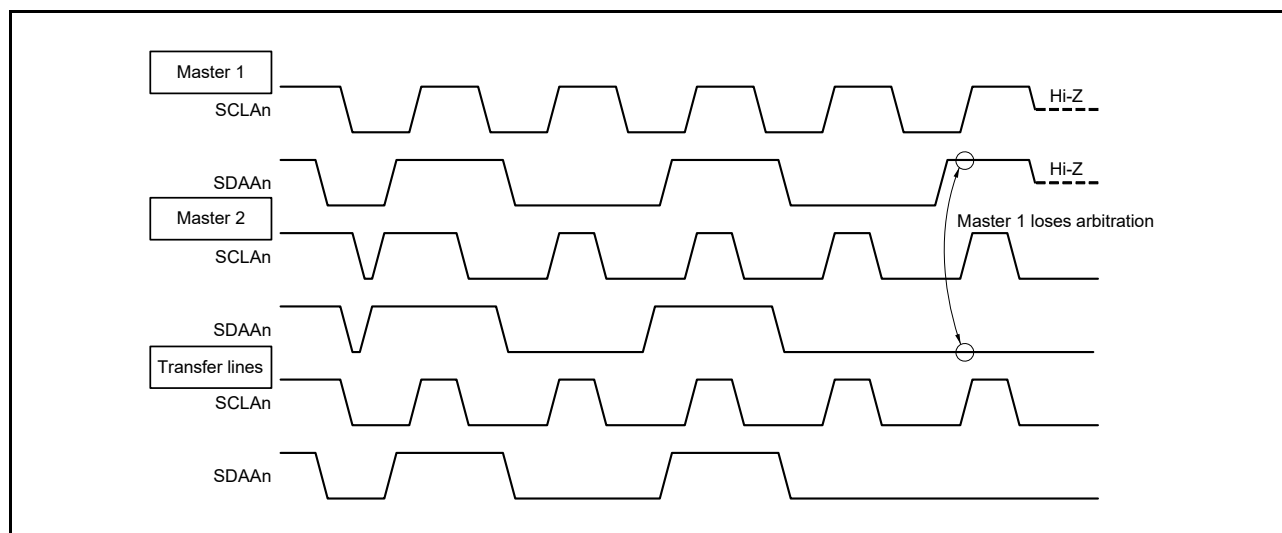
Remark 2. n = 0, 1

20.5.12 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 to 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 **20.5.8 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 20 - 21 Arbitration Timing Example



Remark n = 0, 1

Table 20 - 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.

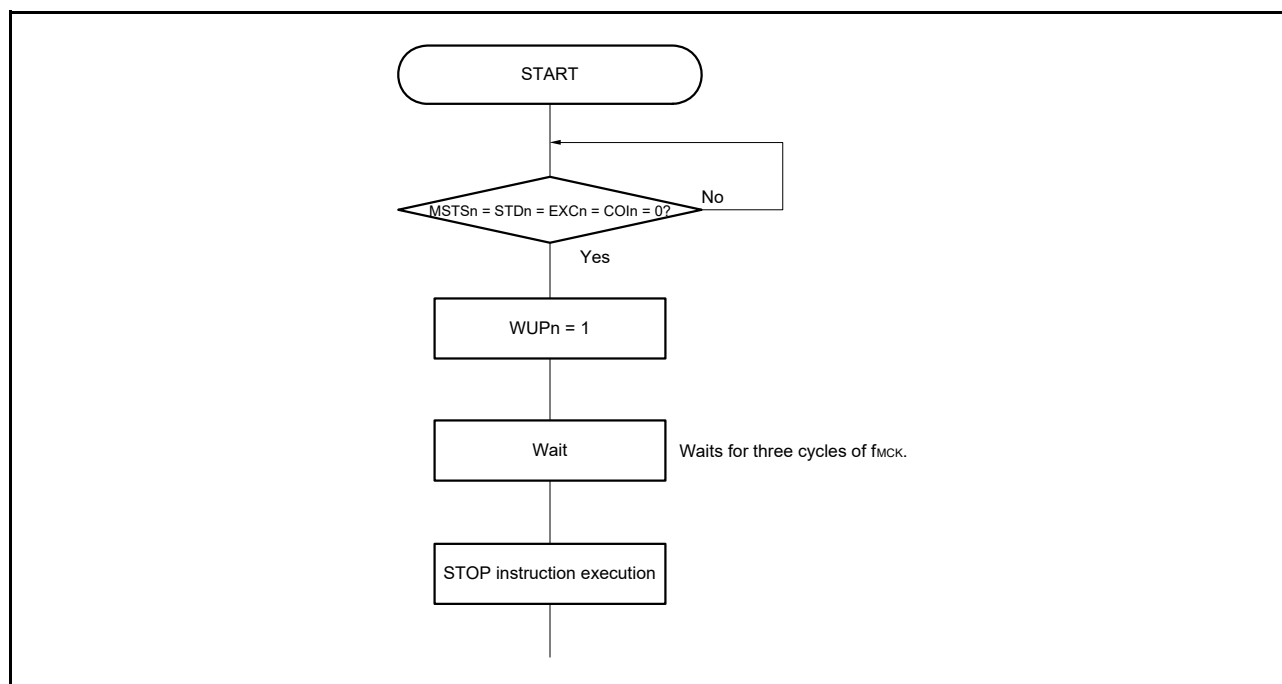
Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

20.5.13 Wakeup Function

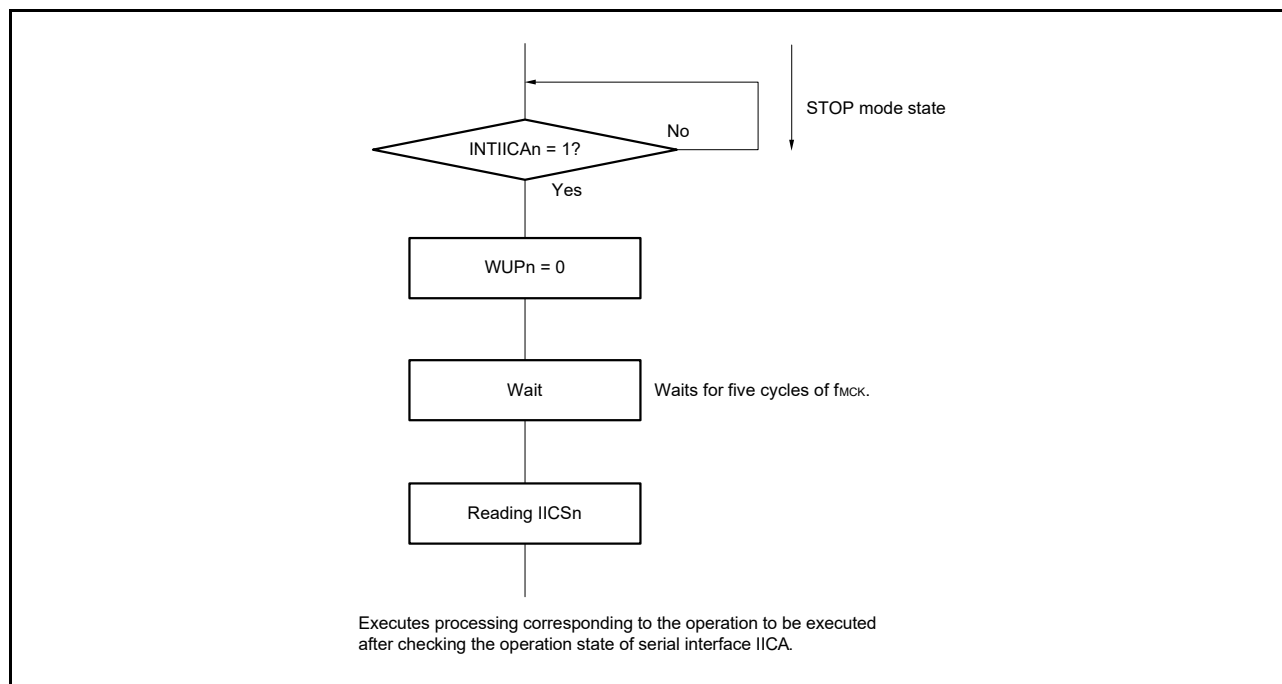
The I²C bus slave function is a function that generates an interrupt request signal (INTIICAn) when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. This function makes processing more efficient by preventing unnecessary 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 the WUPn bit to 0 after this interrupt has been generated. **Figure 20 - 22** shows the flow for setting WUPn = 1 and **Figure 20 - 23** shows the flow for setting WUPn = 0 upon an address match (or when the all address match function is enabled).

Figure 20 - 22 Flow When Setting WUPn = 1



Remark n = 0, 1

Figure 20 - 23 Flow When Setting WUPn = 0 upon Address Match (or When the All Address Match Function is Enabled) (Including Extension Code Reception)

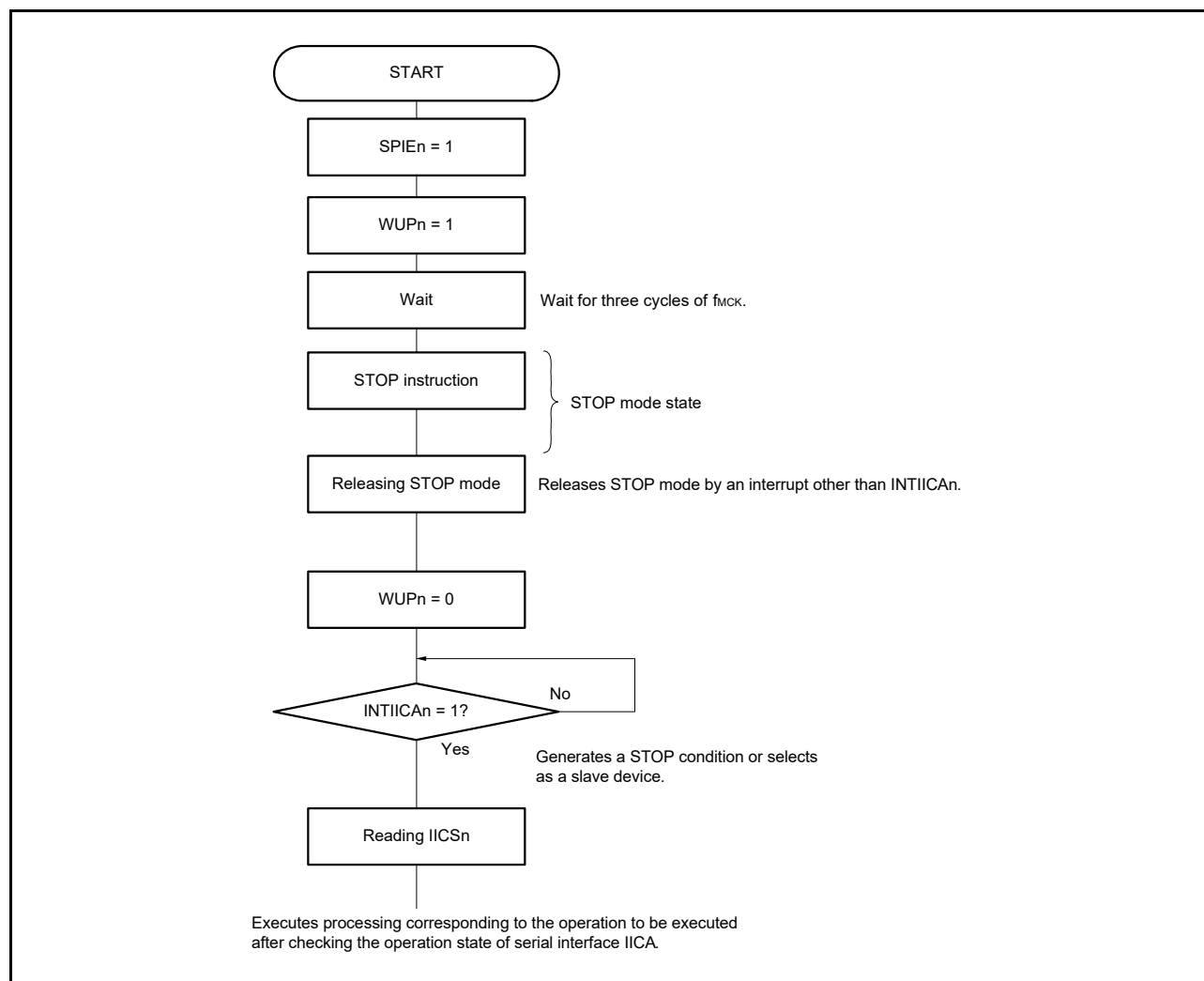


Use the following flows to perform the processing to release the STOP mode other than by an interrupt request signal (INTIICAn) generated from serial interface IICA.

- When operating next IIC communication as master: Flow shown in **Figure 20 - 24**.
- When operating next IIC communication as slave:
 - When released by INTIICAn interrupt: Same as the flow in **Figure 20 - 23**.
 - When released by other than INTIICAn interrupt: Wait for INTIICAn interrupt with WUPn left set to 1.

Remark n = 0, 1

Figure 20 - 24 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn



Remark $n = 0, 1$

20.5.14 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.

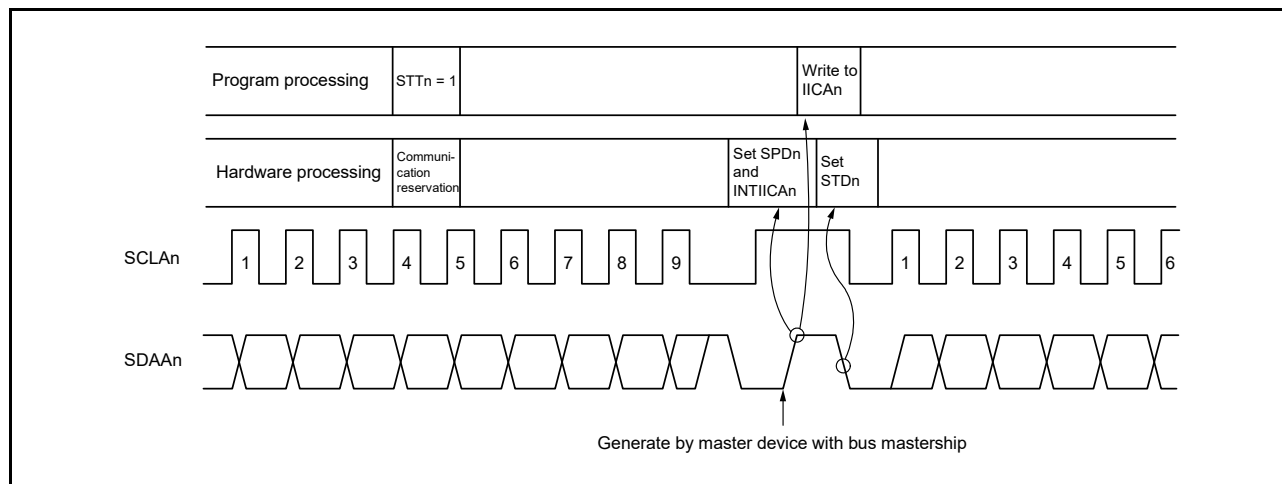
Wait time from setting STTn = 1 to checking the MSTSn flag:
 $(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4) / f_{MCK} + t_F \times 2$

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, 1

Figure 20 - 25 shows the communication reservation timing.

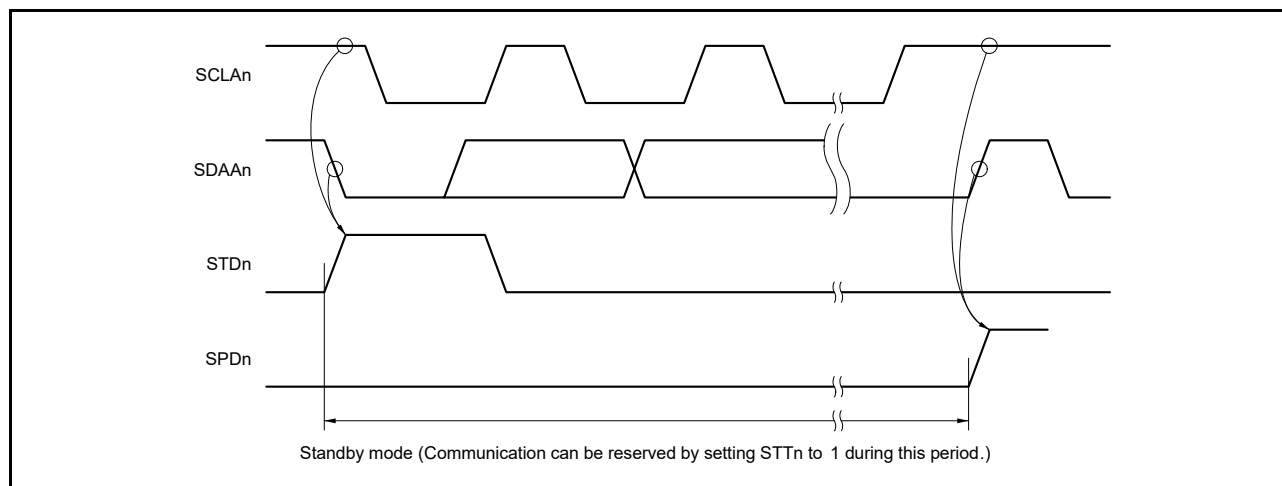
Figure 20 - 25 Communication Reservation Timing



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 20 - 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.

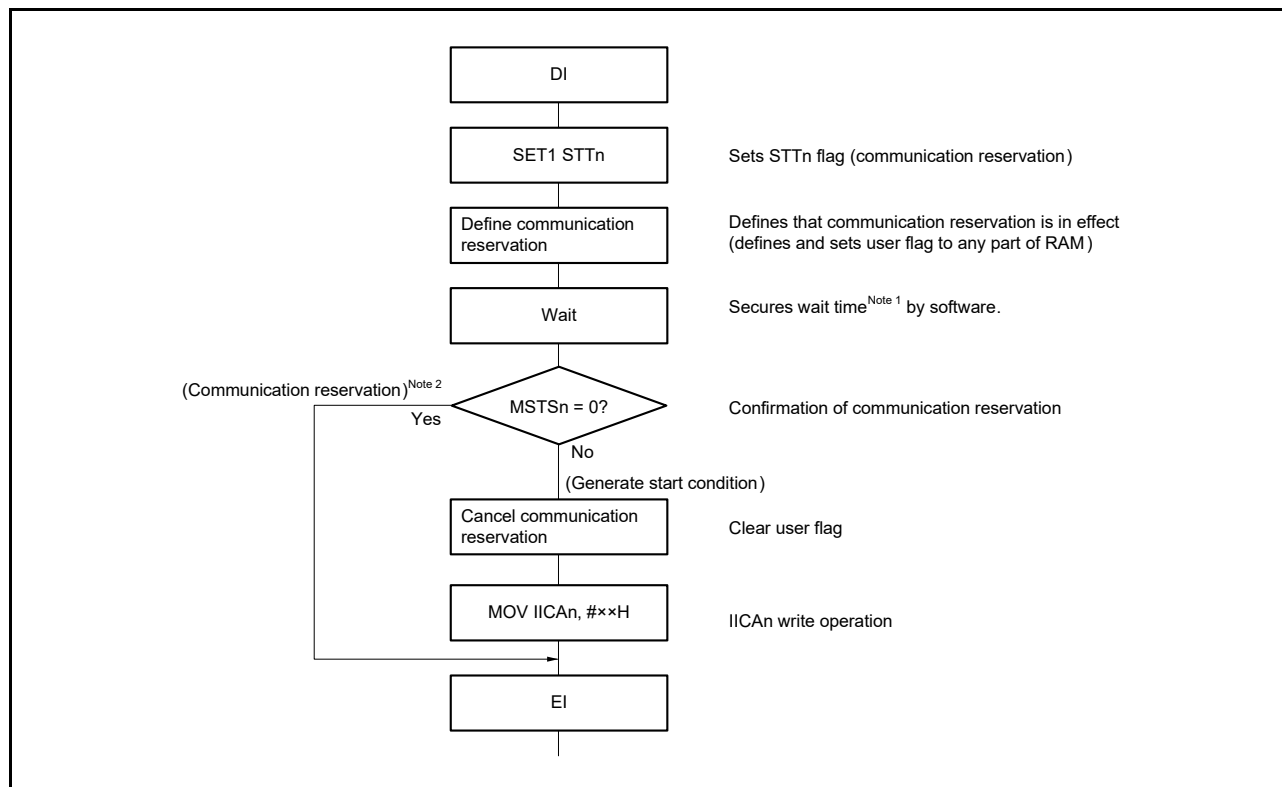
Figure 20 - 26 Timing for Accepting Communication Reservations



Remark n = 0, 1

Figure 20 - 27 shows the communication reservation protocol.

Figure 20 - 27 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.

$$(\text{IICWLn setting value} + \text{IICWHn setting value} + 4) / \text{fMCK} + \text{tF} \times 2$$

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

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, 1

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, 1

20.5.15 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 f_{MCK} 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, 1

20.5.16 Communication Operations

The following shows three operation procedures with the flowchart.

1. Master operation in single-master system

The flowchart when using the RL78/L23 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/L23 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/L23 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

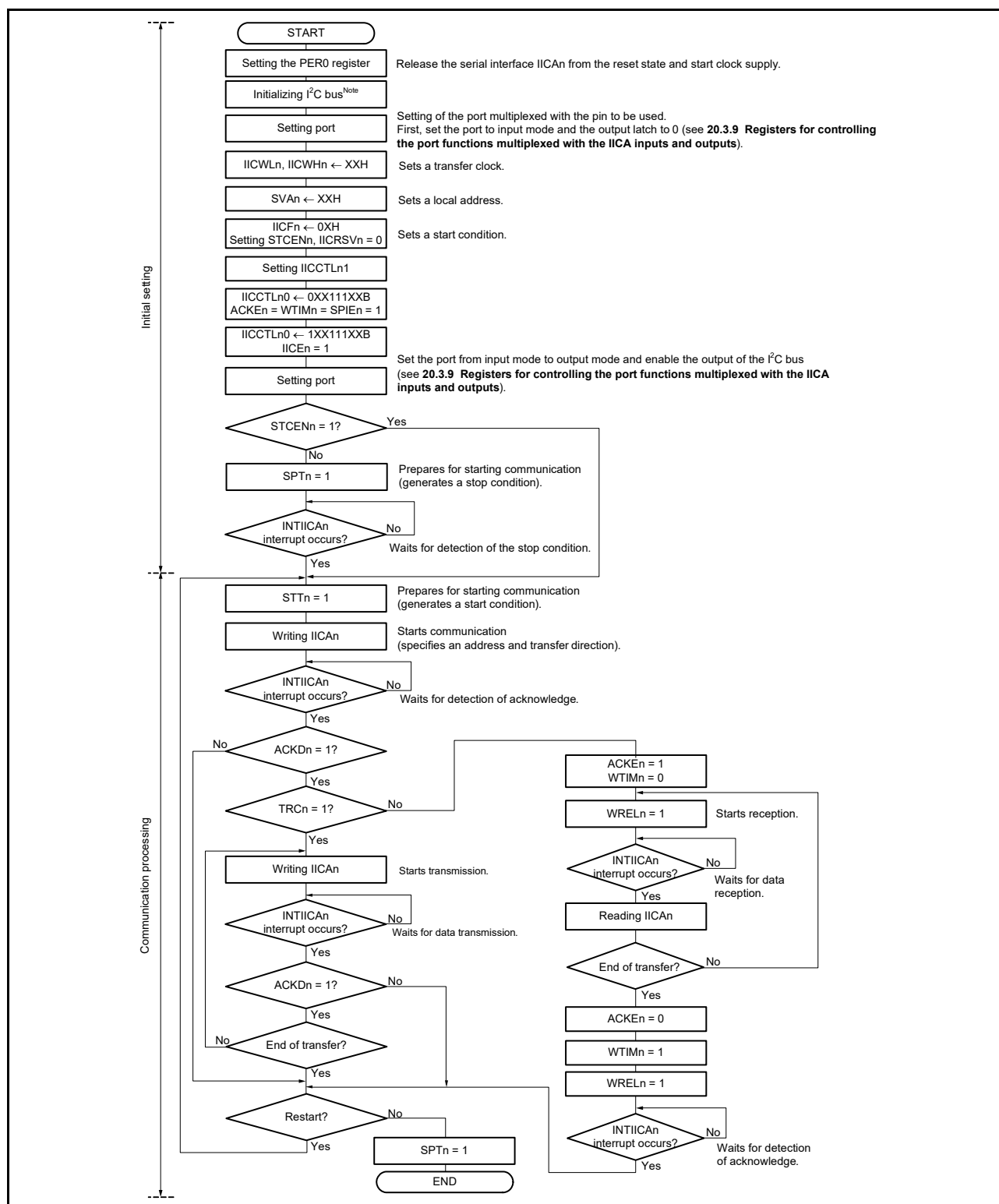
An example of when the RL78/L23 is used as the I²C bus slave is shown below.

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

1. Master operation in single-master system

Figure 20 - 28 Master Operation in Single-master System



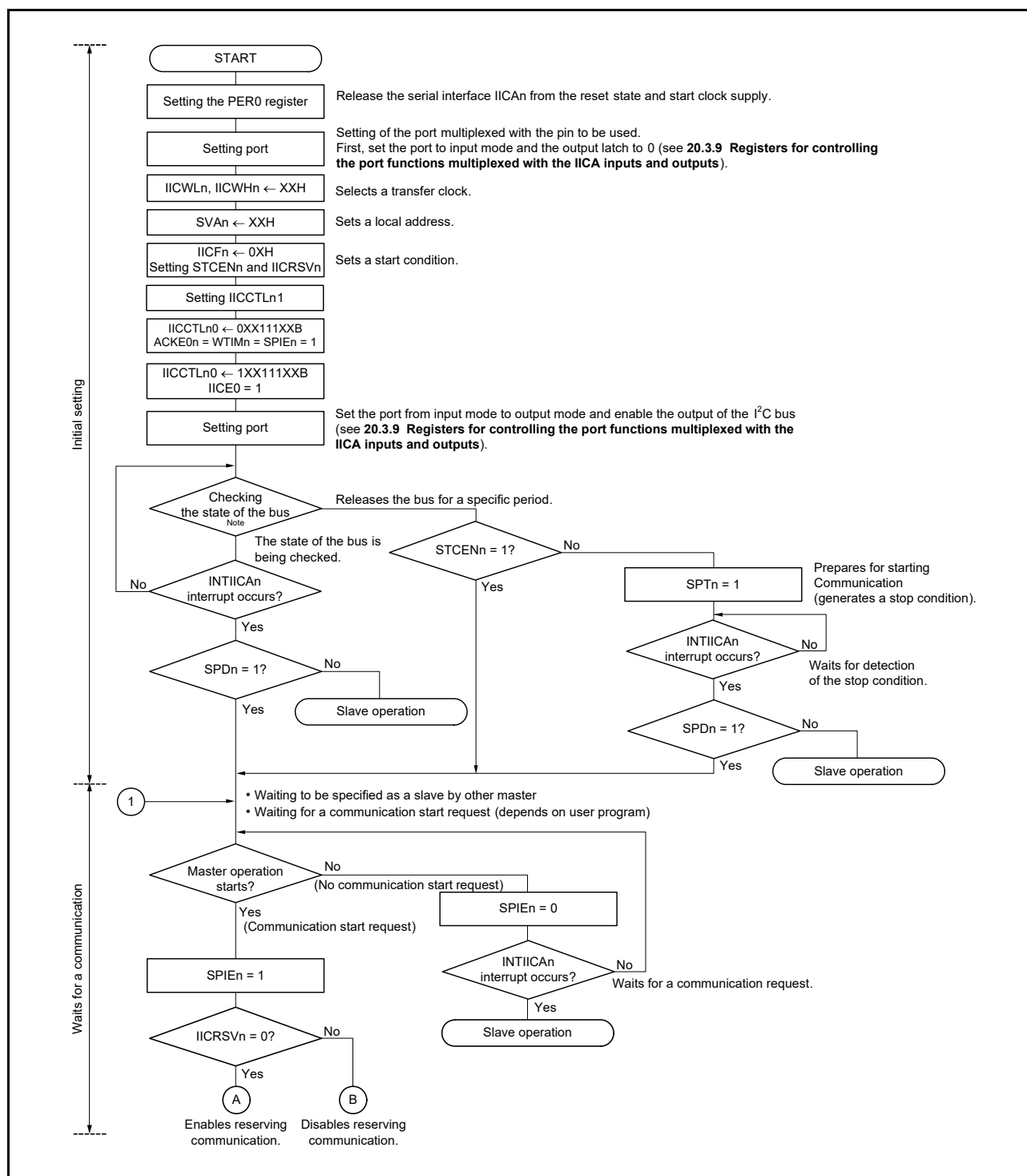
Note Release (SCLAn and SDAAn pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the 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, 1

2. Master operation in multi-master system

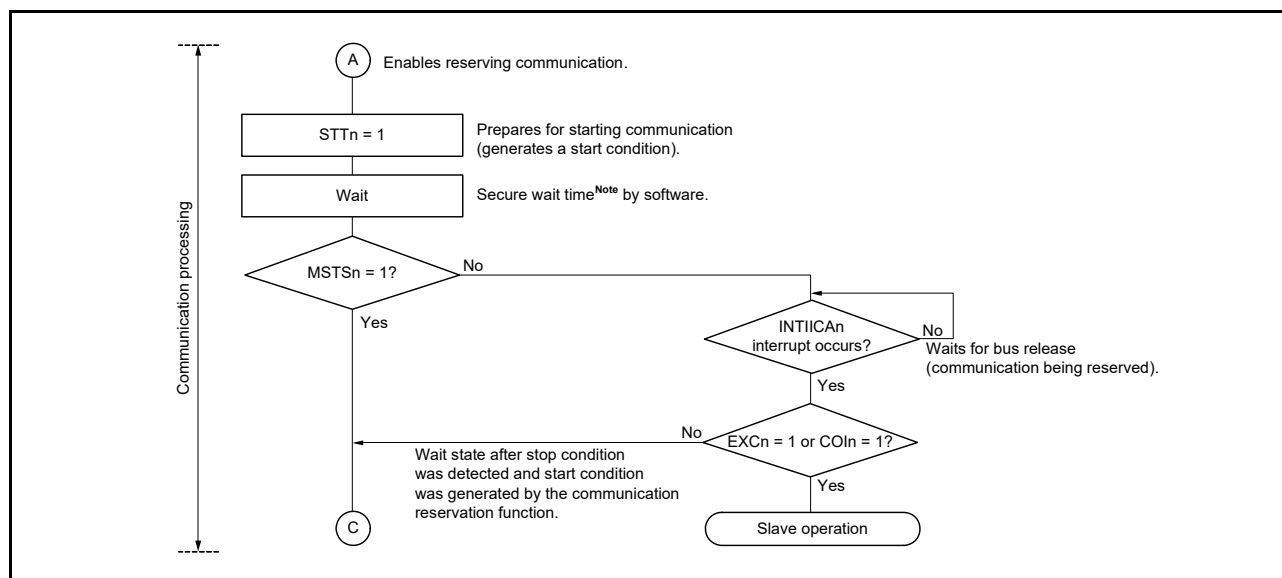
Figure 20 - 29 Master Operation in Multi-master System (1/3)



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.

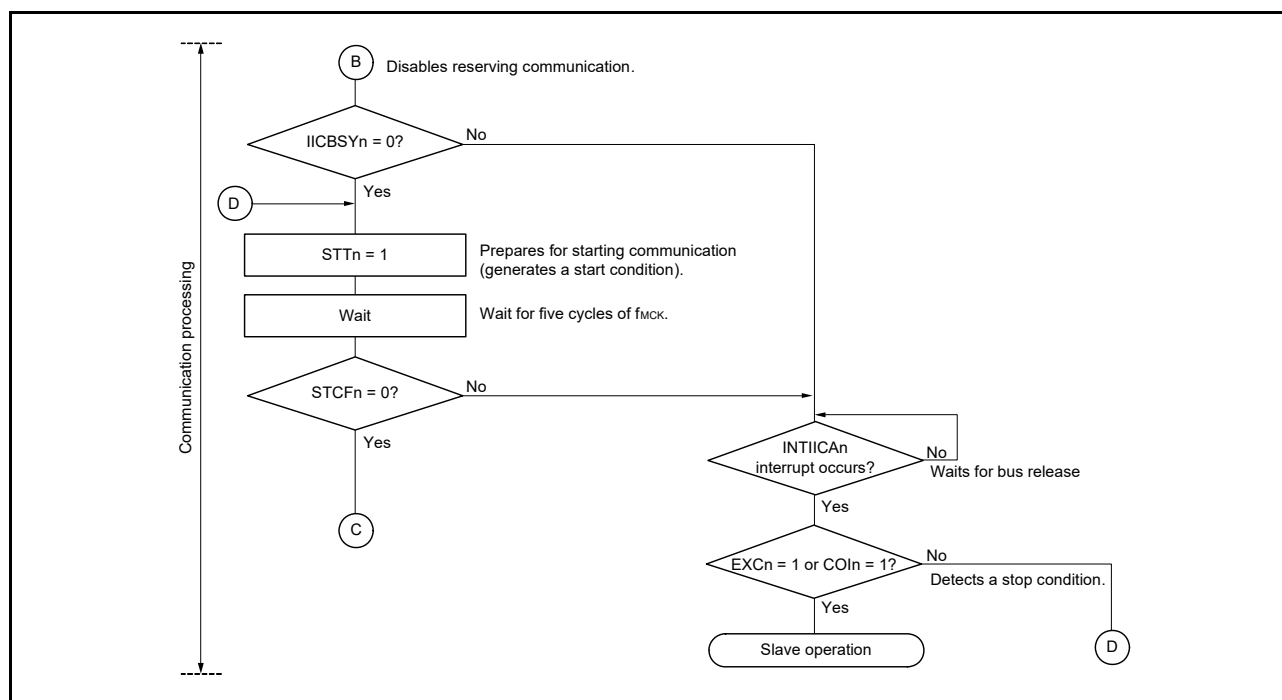
Remark n = 0, 1

Figure 20 - 29 Master Operation in Multi-master System (2/3)



Note The wait time is calculated as follows.

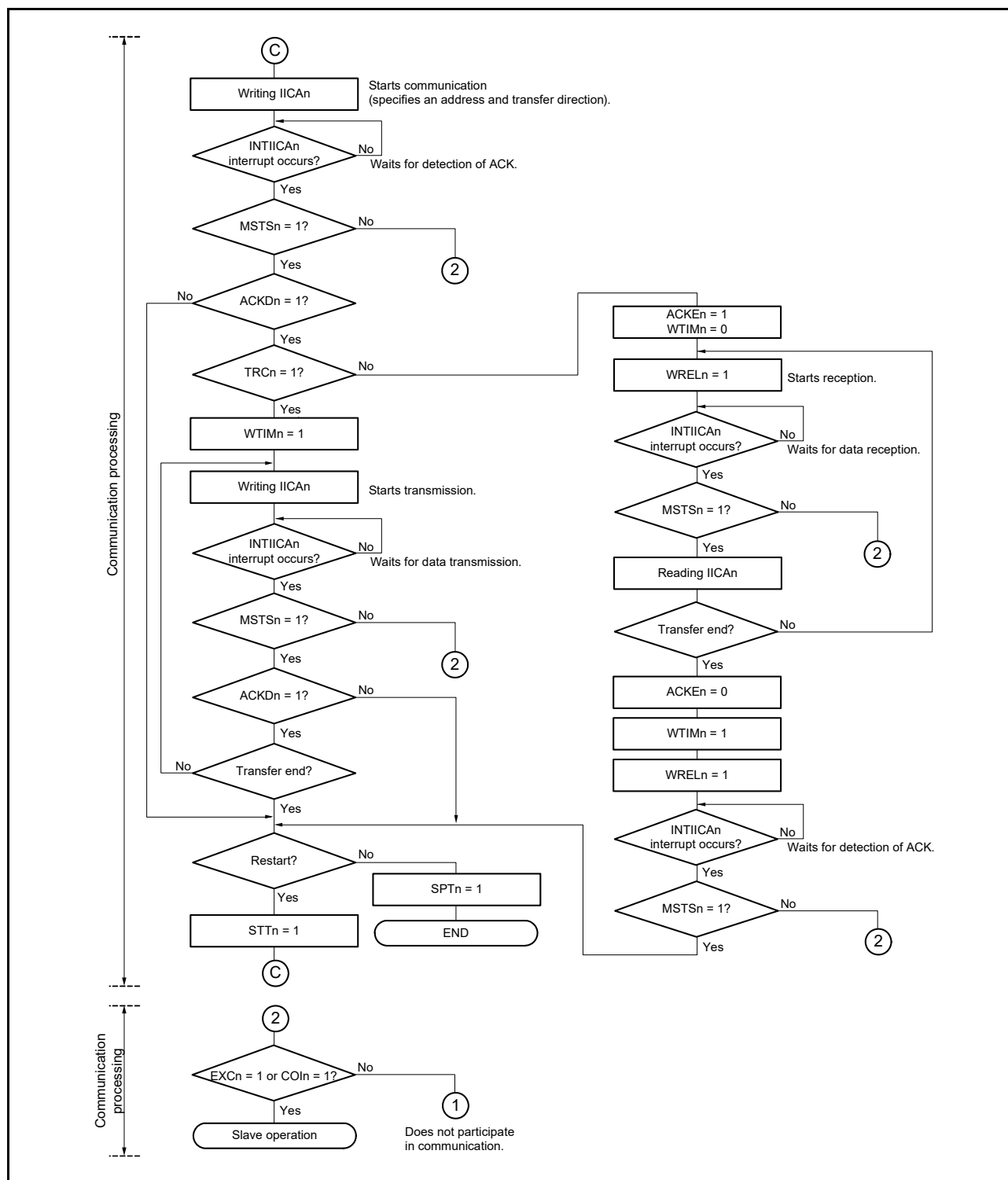
$$(IICWLn \text{ setting value} + IICWHn \text{ setting value} + 4)/f_{MCK} + t_F \times 2$$



Remark 1. IICWL_n: IICA low-level width setting register n
 IICWH_n: IICA high-level width setting register n
 t_F: SDA_An and SCL_An signal falling times
 f_{MCK}: IICA operation clock frequency

Remark 2. n = 0, 1

Figure 20 - 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 MSTS_n bit each time interrupt INTIICAn has occurred to check the arbitration result.

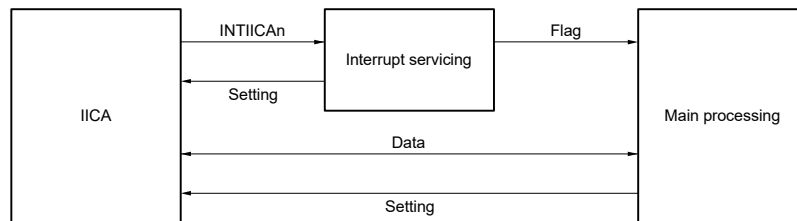
Remark 3. To use the device as a slave in a multi-master system, check the state by using the IICA status register *n* (IICS_n) and IICA flag register *n* (IICF_n) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

Remark 4. *n* = 0, 1

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 routine only handles state transition processing, and that the main processing handles actual data communications.



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).

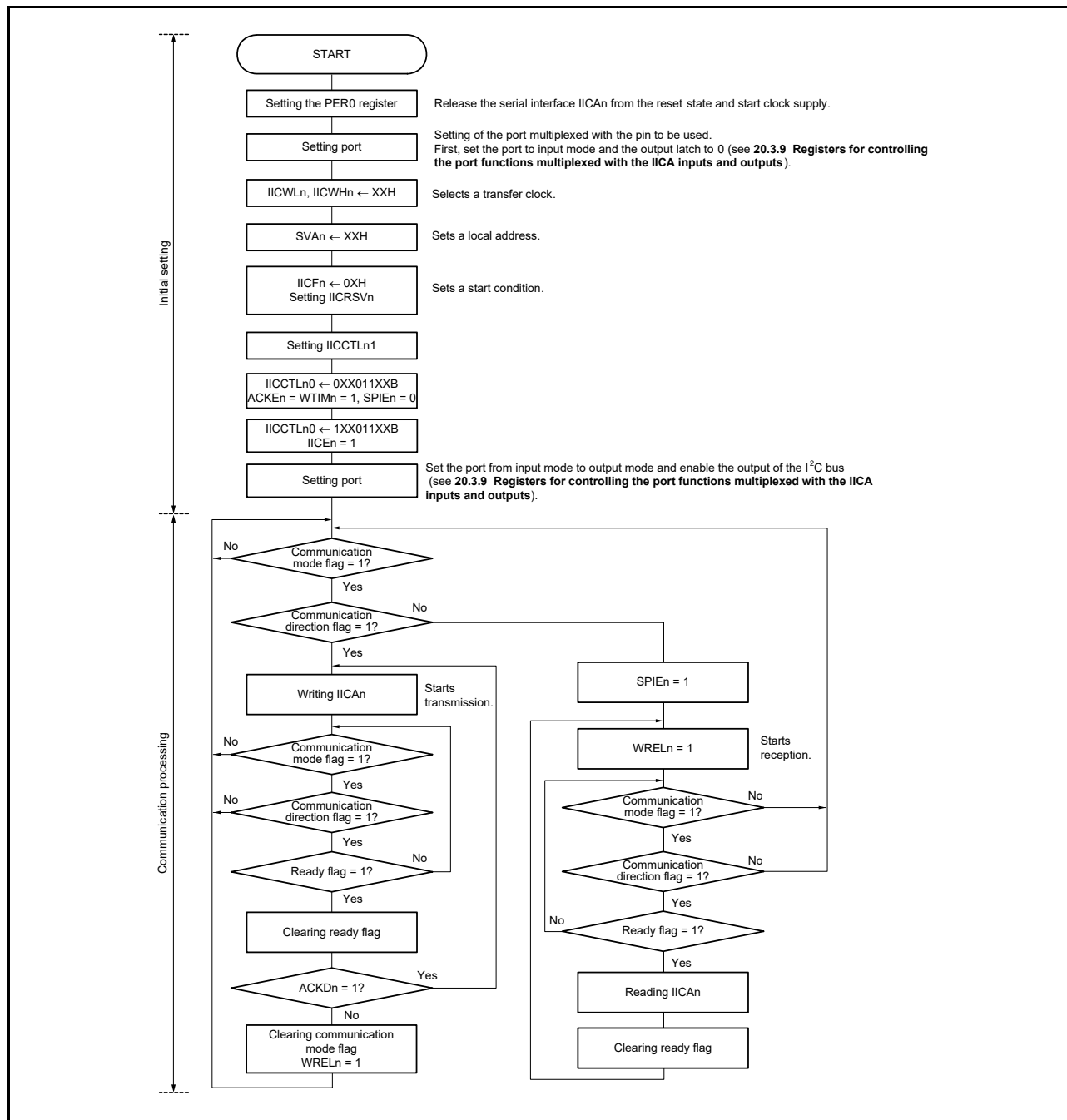
<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

Remark n = 0, 1

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 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 20 - 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.

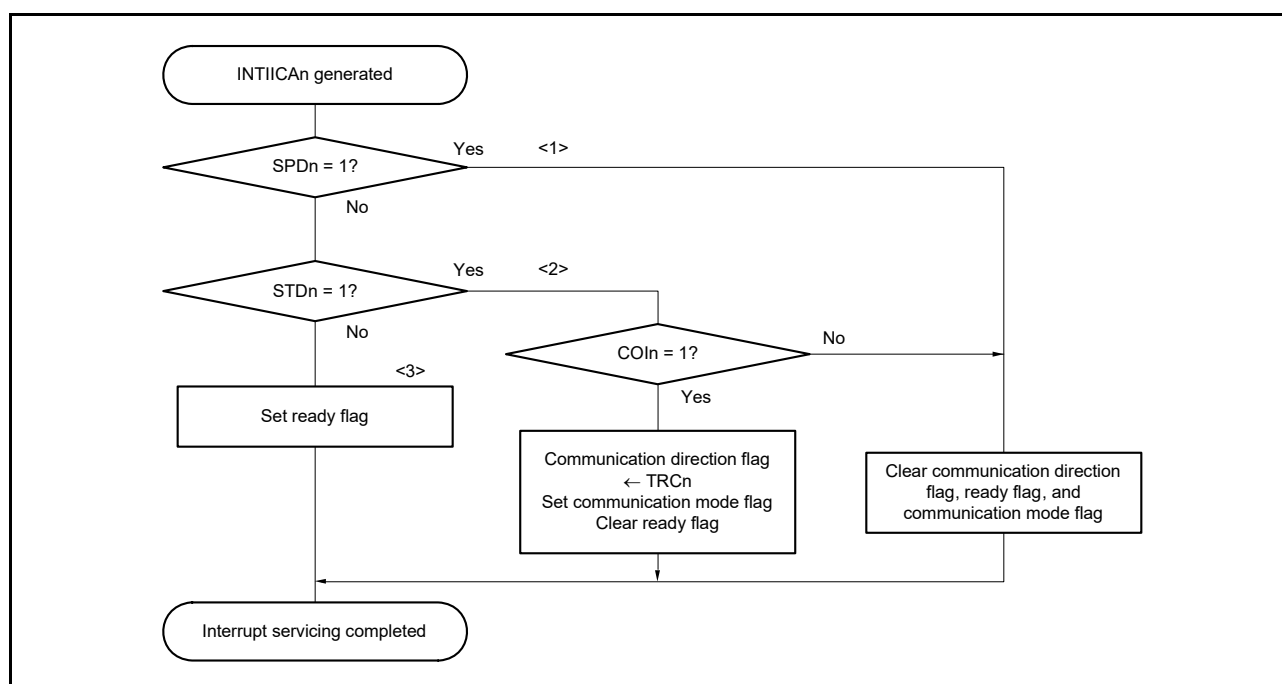
Remark 2. n = 0, 1

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 20 - 31 Slave Operation Flowchart (2)**.

Figure 20 - 31 Slave Operation Flowchart (2)



Remark n = 0, 1

20.5.17 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

R/W: Transfer direction specification

ACK: Acknowledge

D7 to D0: Data

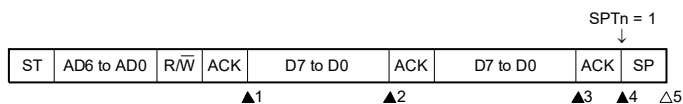
SP: Stop condition

Remark 2. n = 0, 1

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)

Δ 5: IICSn = 00000001B

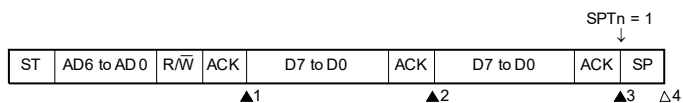
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

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)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated

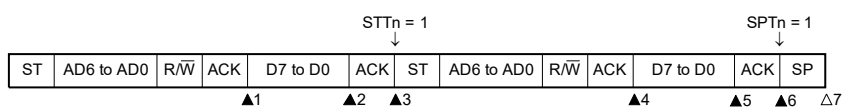
Δ: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

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)

Δ 7: 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)

Δ 5: IICSn = 00000001B

Remark ▲: Always generated

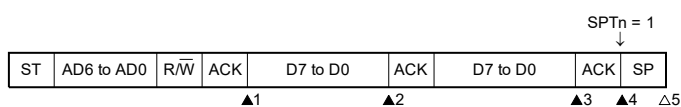
Δ: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

i) When WTIMn = 0



▲ 1: IICSn = 1010×110B

▲ 2: IICSn = 1010×000B

▲ 3: IICSn = 1010×000B (Sets the WTIMn bit to 1)^{Note}

▲ 4: IICSn = 1010××00B (Sets the SPTn bit to 1)

Δ 5: IICSn = 00000001B

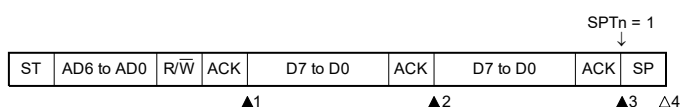
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

×: Don't care

ii) When WTIMn = 1



▲ 1: IICSn = 1010×110B

▲ 2: IICSn = 1010×100B

▲ 3: IICSn = 1010××00B (Sets the SPTn bit to 1)

Δ 4: IICSn = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn = 1

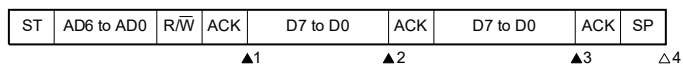
×: Don't care

Remark n = 0, 1

2. Slave device operation (slave address data reception)

a) Start ~ Address ~ Data ~ Data ~ Stop

i) When WTIMn = 0



▲ 1: IICSn = 0001×110B

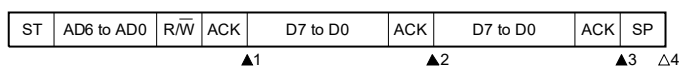
▲ 2: IICSn = 0001×000B

▲ 3: IICSn = 0001×000B

△ 4: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

ii) When WTIMn = 1



▲ 1: IICSn = 0001×110B

▲ 2: IICSn = 0001×100B

▲ 3: IICSn = 0001××00B

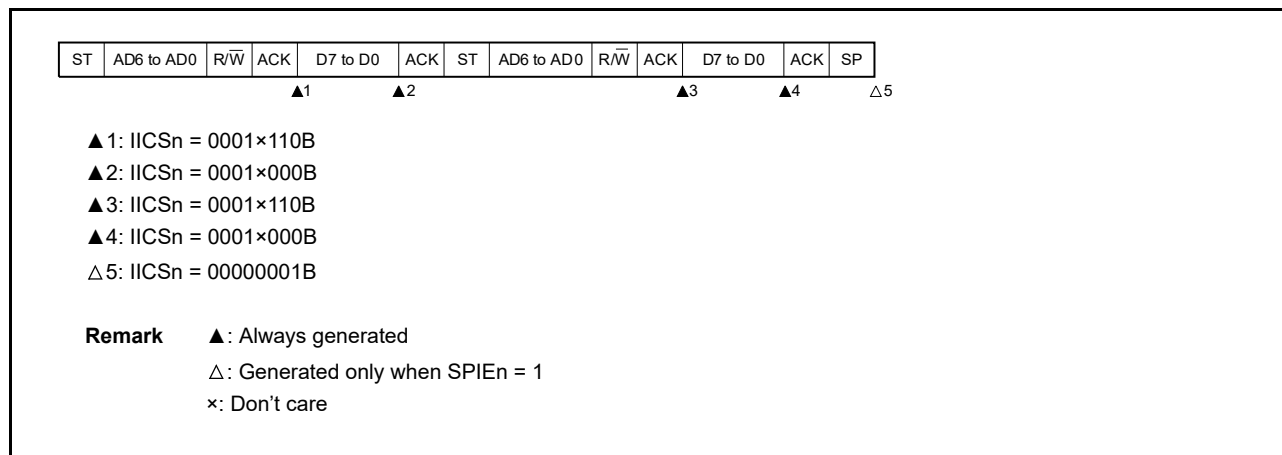
△ 4: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

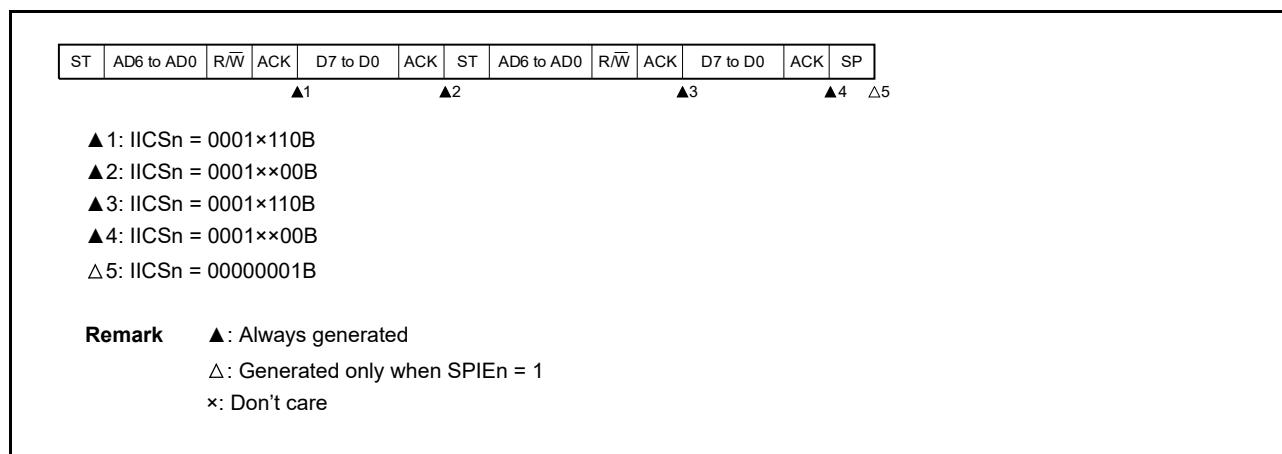
Remark n = 0, 1

b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, matches with SVAn, the all address match function is disabled)



ii) When WTIMn = 1 (after restart, matches with SVAn, the all address match function is disabled)



Remark n = 0, 1

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/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 = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×110B

▲5: IICSn = 0010××00B

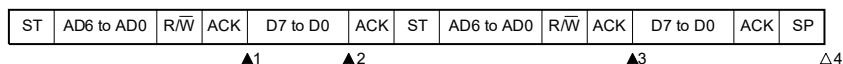
△6: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

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

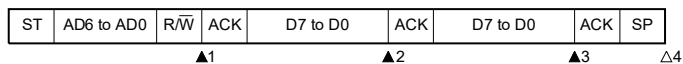
Remark n = 0, 1

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

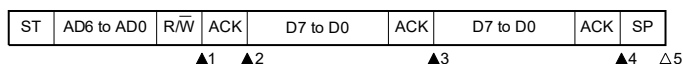
▲2: IICSn = 0010×000B

▲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, 1

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

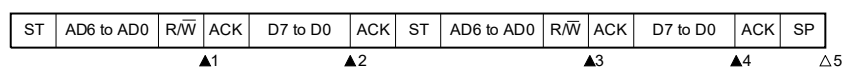
△6: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

i) When WTIMn = 0 (after restart, extension code reception, the all address match function is disabled)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲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, extension code reception, the all address match function is disabled)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0010×010B

▲5: IICSn = 0010×110B

▲6: IICSn = 0010××00B

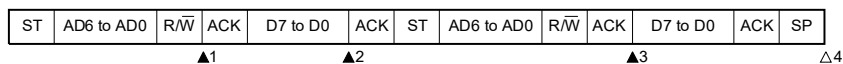
△7: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

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

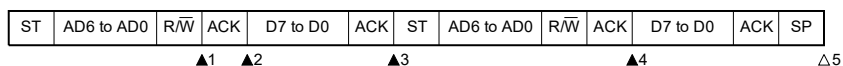
▲2: IICSn = 0010×000B

▲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

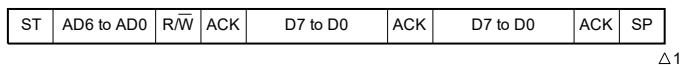
△5: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

4. Operation without communication

a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

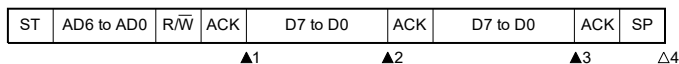
Remark △: Generated only when SPIEn = 1

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



▲1: IICSn = 0101×110B

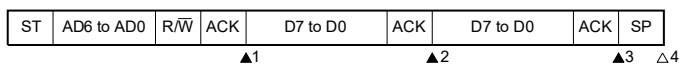
▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

△4: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

ii) When WTIMn = 1



▲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

Remark n = 0, 1

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

ii) When WTIMn = 1

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3		▲4 △5

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

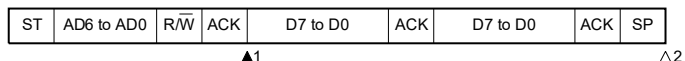
Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

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)

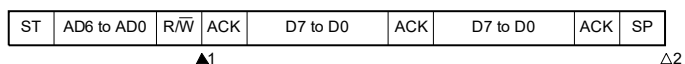


▲ 1: IICSn = 01000110B

△ 2: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1

b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)



▲ 1: IICSn = 0110×010B

Sets LRELn = 1 by software

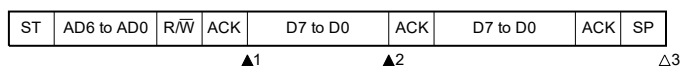
△ 2: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

c) When arbitration loss occurs during transmission of data

i) When WTIMn = 0



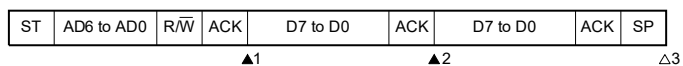
▲ 1: IICSn = 10001110B

▲ 2: IICSn = 01000000B

△ 3: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1

ii) When WTIMn = 1



▲ 1: IICSn = 10001110B

▲ 2: IICSn = 01000100B

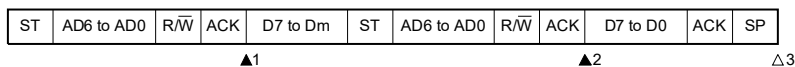
△ 3: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1

Remark n = 0, 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

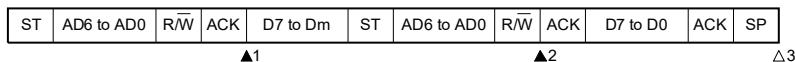
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

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

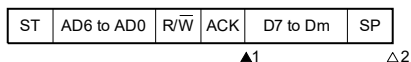
Remark ▲: Always generated

△: 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, 1

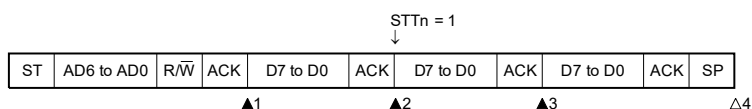
- f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition
 i) When WTIMn = 0



- ▲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



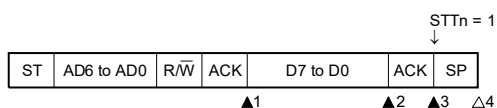
- ▲1: IICSn = 1000×110B
 ▲2: IICSn = 1000×100B (Sets the STTn bit to 1)
 ▲3: IICSn = 01000100B
 △4: IICSn = 00000001B

Remark ▲: Always generated
 △: Generated only when SPIEn = 1
 ×: Don't care

Remark n = 0, 1

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, 1

h) When arbitration loss occurs due to low-level data when attempting to generate a stop 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 = 01000100B

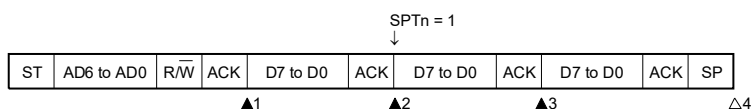
△ 5: IICSn = 00000001B

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 $SPTn$ bit to 1)

▲ 3: IICSn = 01000100B

△ 4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when $SPIEn = 1$

×: Don't care

Remark n = 0, 1

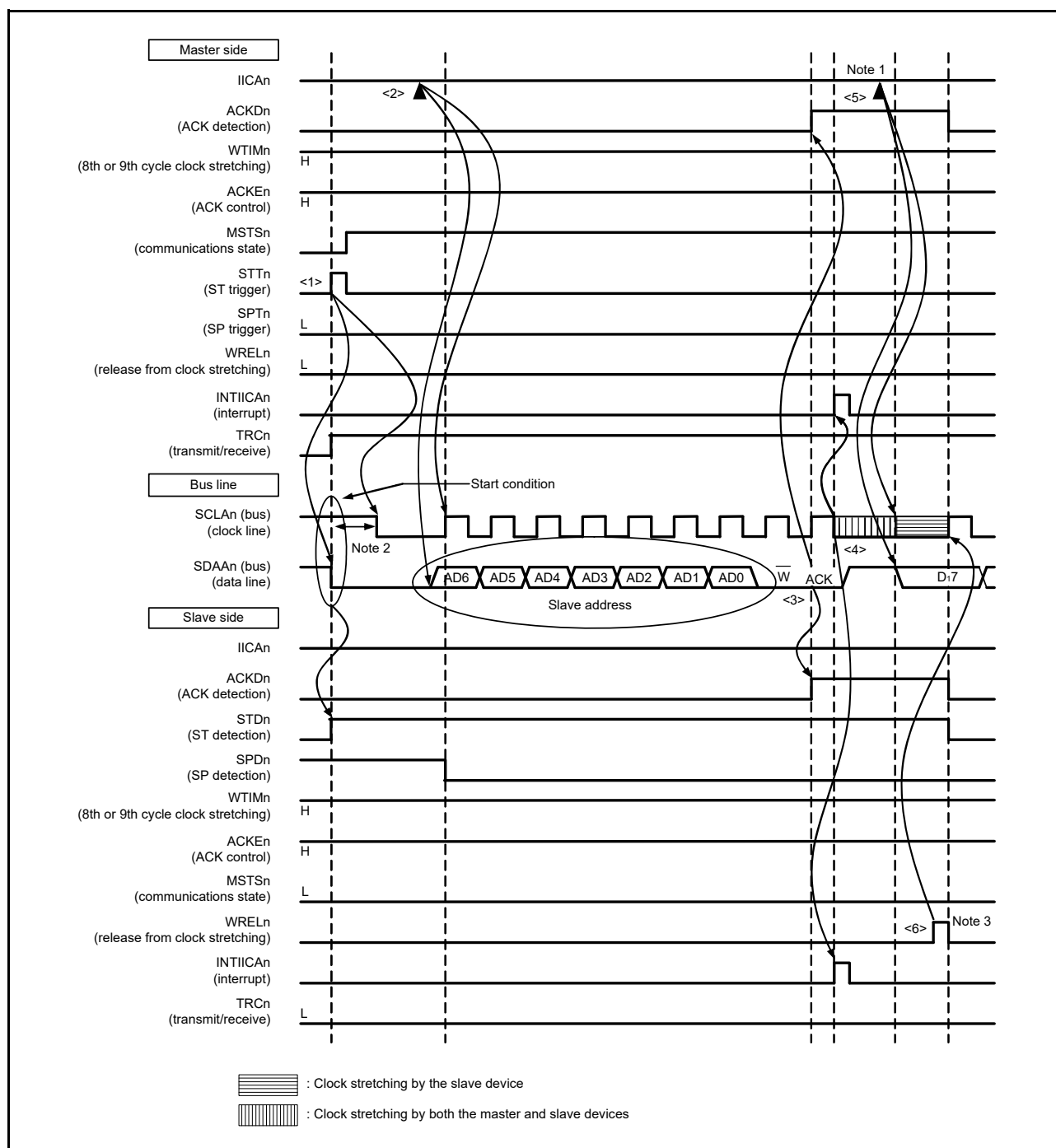
20.6 Timing Charts

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 20 - 32** and **Figure 20 - 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, 1

Figure 20 - 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 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 reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <1> to <6> in **1. Start condition to address to data** in **Figure 20 - 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.

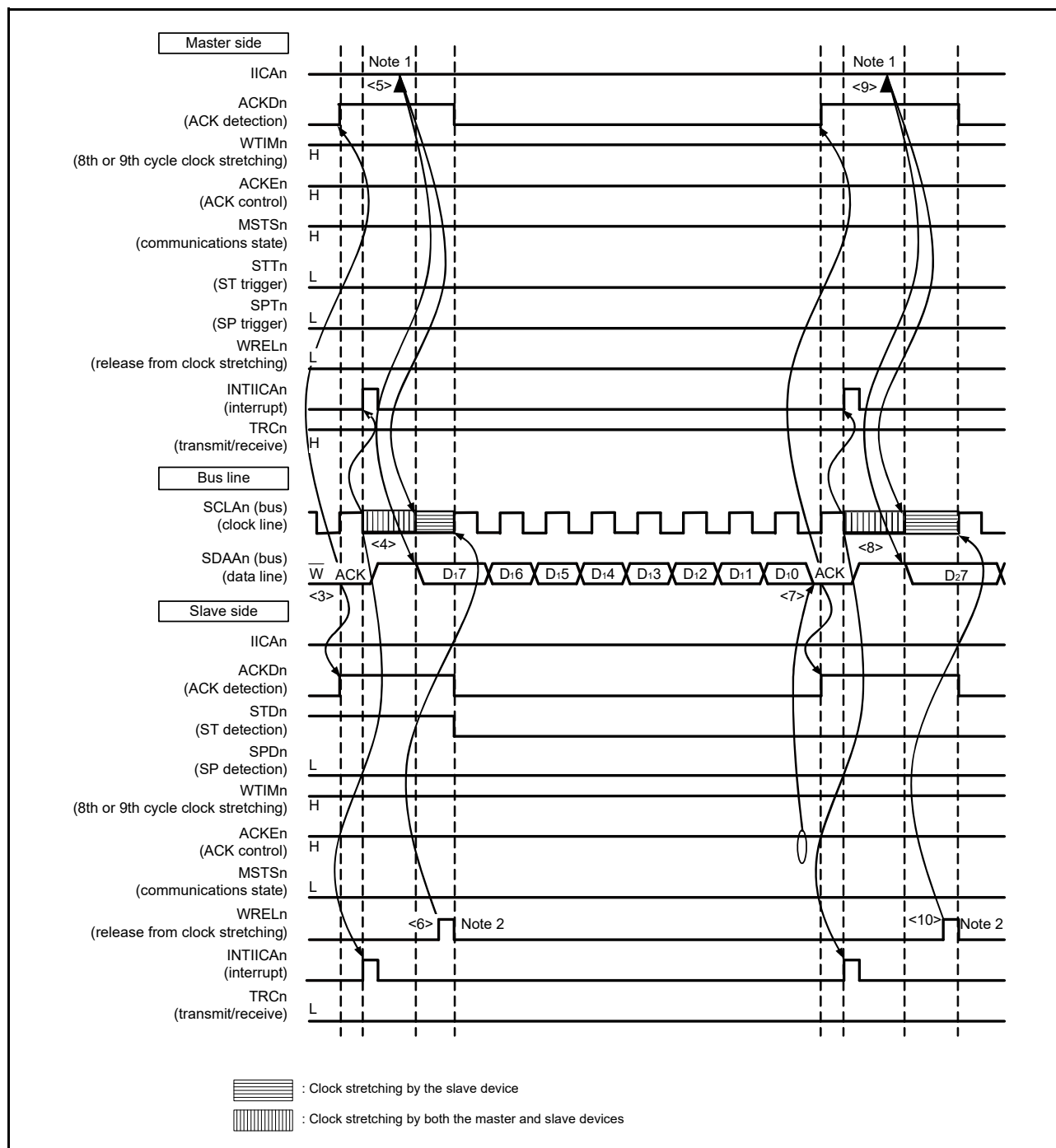
Remark 1. <1> to <15> in **Figure 20 - 32** represent the entire procedure for communicating data using the I²C bus.

- 1. Start condition to address to data** in **Figure 20 - 32**: Steps <1> to <6>
- 2. Address to data to data** in **Figure 20 - 32**: Steps <3> to <10>
- 3. Data to data to stop condition** in **Figure 20 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 20 - 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, 1

The meanings of <3> to <10> in **2. Address to data to data** in **Figure 20 - 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.

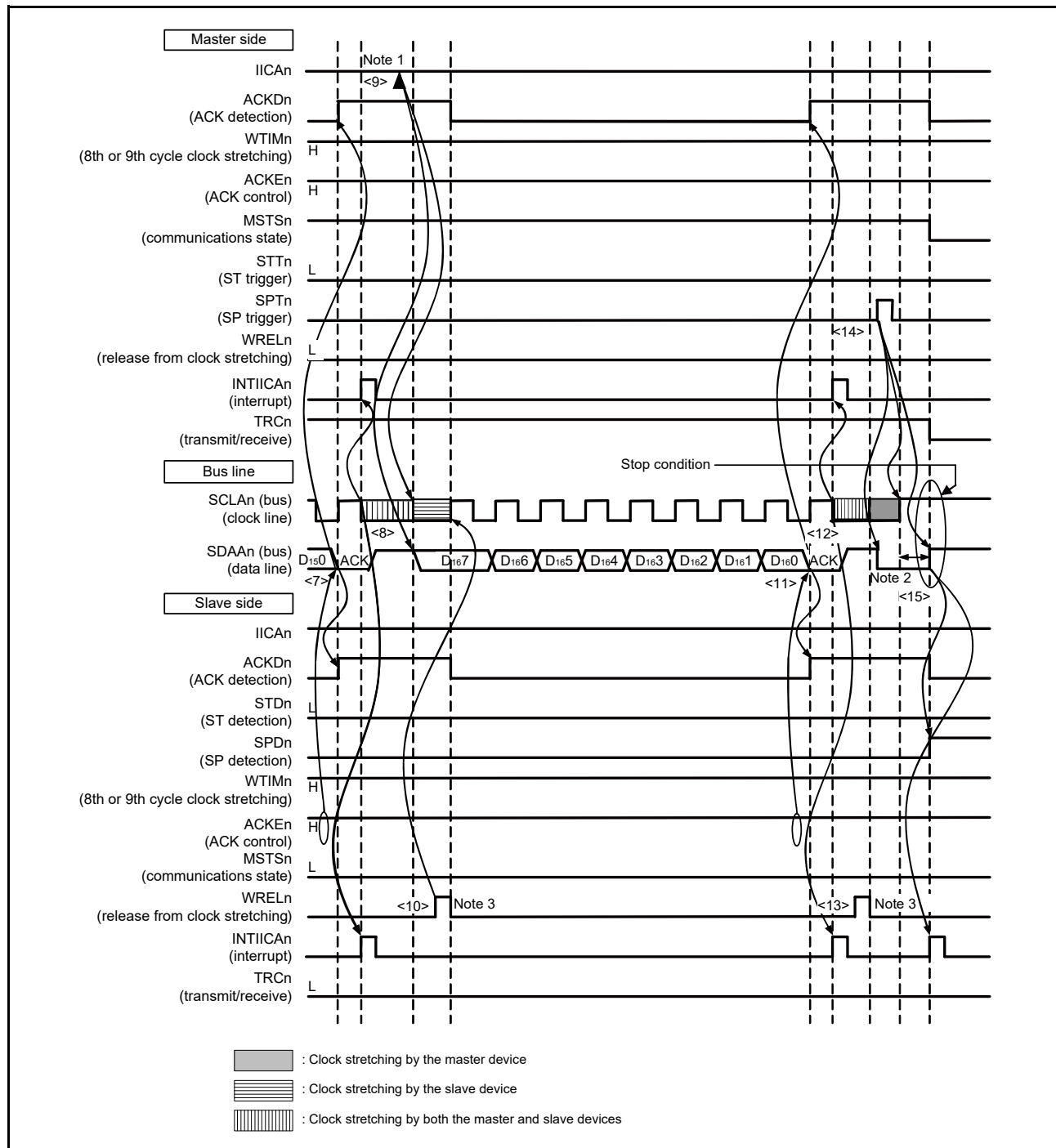
Remark 1. <1> to <15> in **Figure 20 - 32** represent the entire procedure for communicating data using the I²C bus.

1. **Start condition to address to data** in **Figure 20 - 32**: Steps <1> to <6>
2. **Address to data to data** in **Figure 20 - 32**: Steps <3> to <10>
3. **Data to data to stop condition** in **Figure 20 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 20 - 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 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 reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark $n = 0, 1$

The meanings of <7> to <15> in **3. Data to data to stop condition** in **Figure 20 - 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).

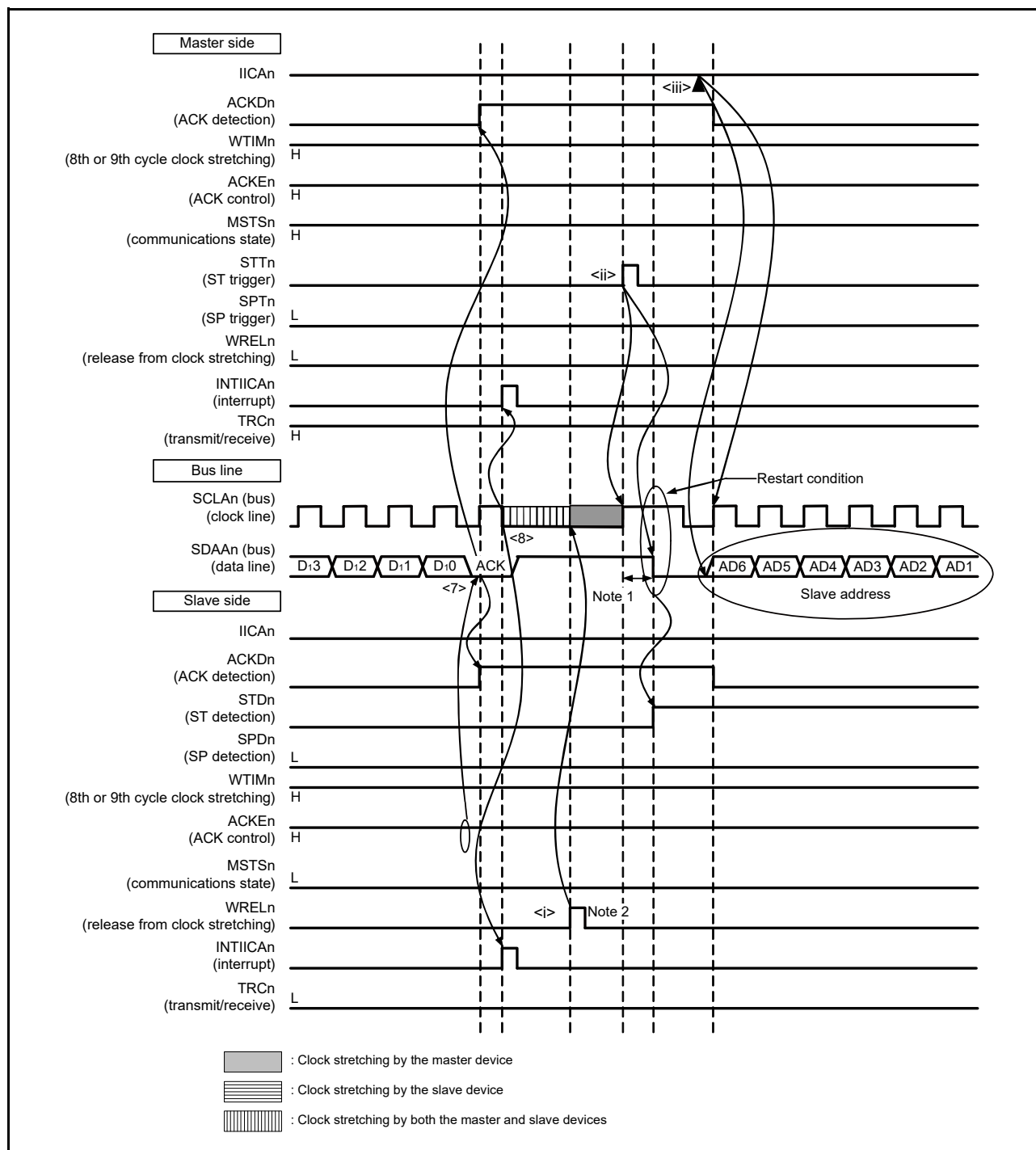
Remark 1. <1> to <15> in **Figure 20 - 32** represent the entire procedure for communicating data using the I²C bus.

- 1. Start condition to address to data** in **Figure 20 - 32**: Steps <1> to <6>
- 2. Address to data to data** in **Figure 20 - 32**: Steps <3> to <10>
- 3. Data to data to stop condition** in **Figure 20 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 20 - 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 SCLn 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 $n = 0, 1$

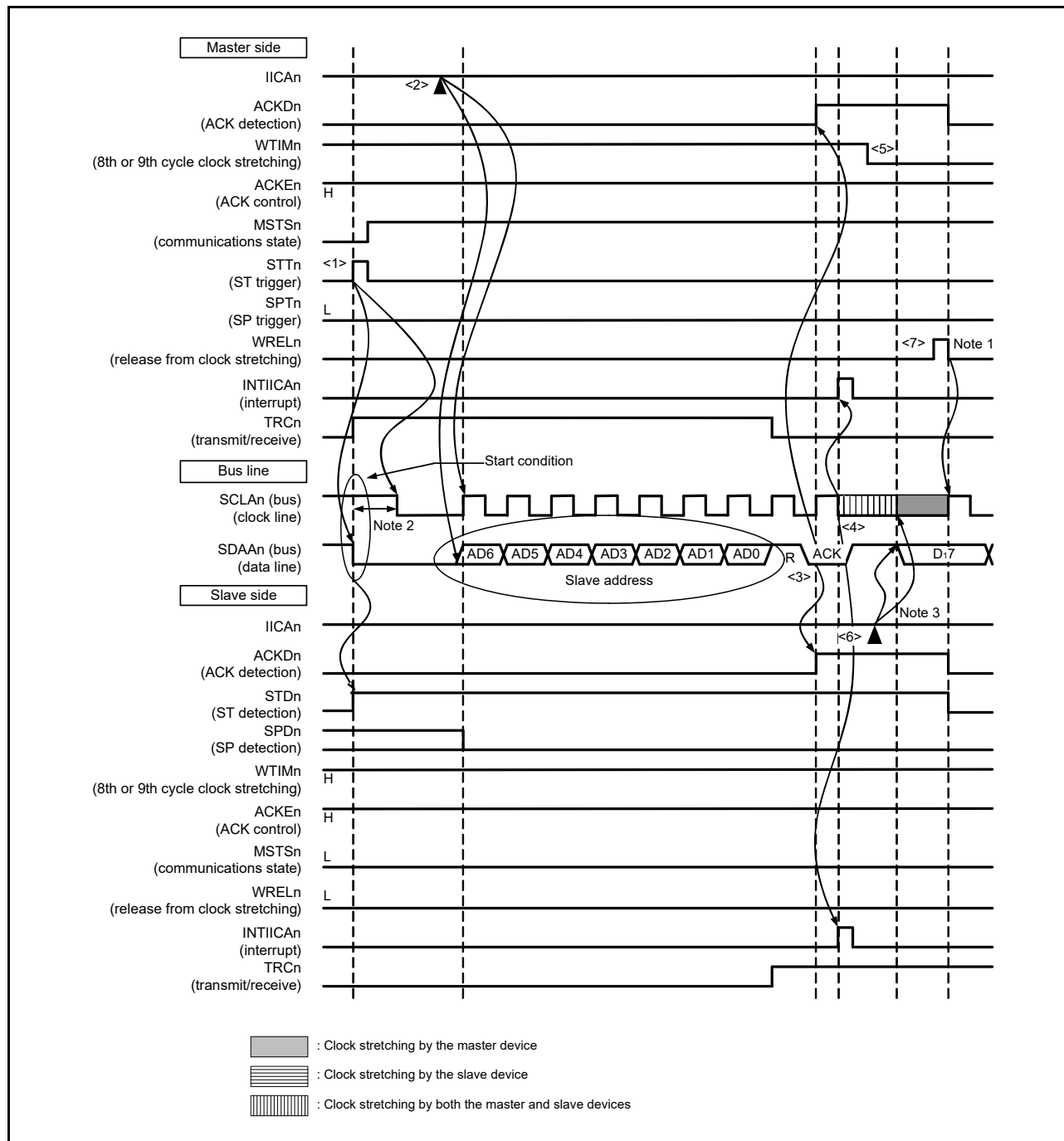
The following describes the operations in **4. Data to restart condition to address** in **Figure 20 - 32**. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <3>, the data transmission step.

- <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, 1

Figure 20 - 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 WREL_n 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 transmission by a slave device, write data to the IICAn register instead of setting the WREL_n bit.

Remark n = 0, 1

The meanings of <1> to <7> in **1. Start condition to address to data** in **Figure 20 - 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 (MSTS_n = 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.

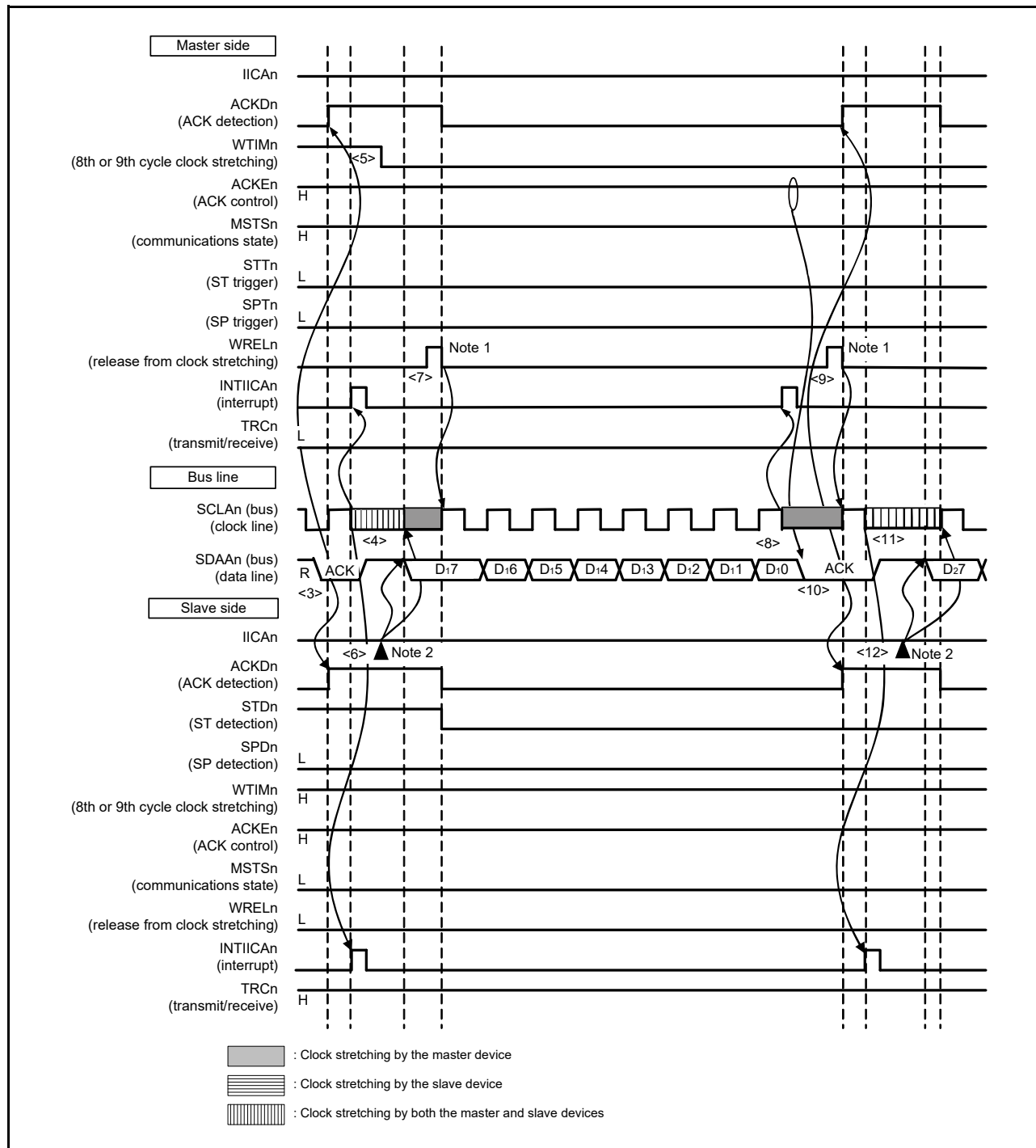
Remark 1. <1> to <19> in **Figure 20 - 33** represent the entire procedure for communicating data using the I²C bus.

- 1. Start condition to address to data** in **Figure 20 - 33**: Steps <1> to <7>
- 2. Address to data to data** in **Figure 20 - 33**: Steps <3> to <12>
- 3. Data to data to stop condition** in **Figure 20 - 33**: Steps <8> to <19>

Remark 2. n = 0, 1

Figure 20 - 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 $n = 0, 1$

The meanings of <3> to <12> in **2. Address to data to data** in **Figure 20 - 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.

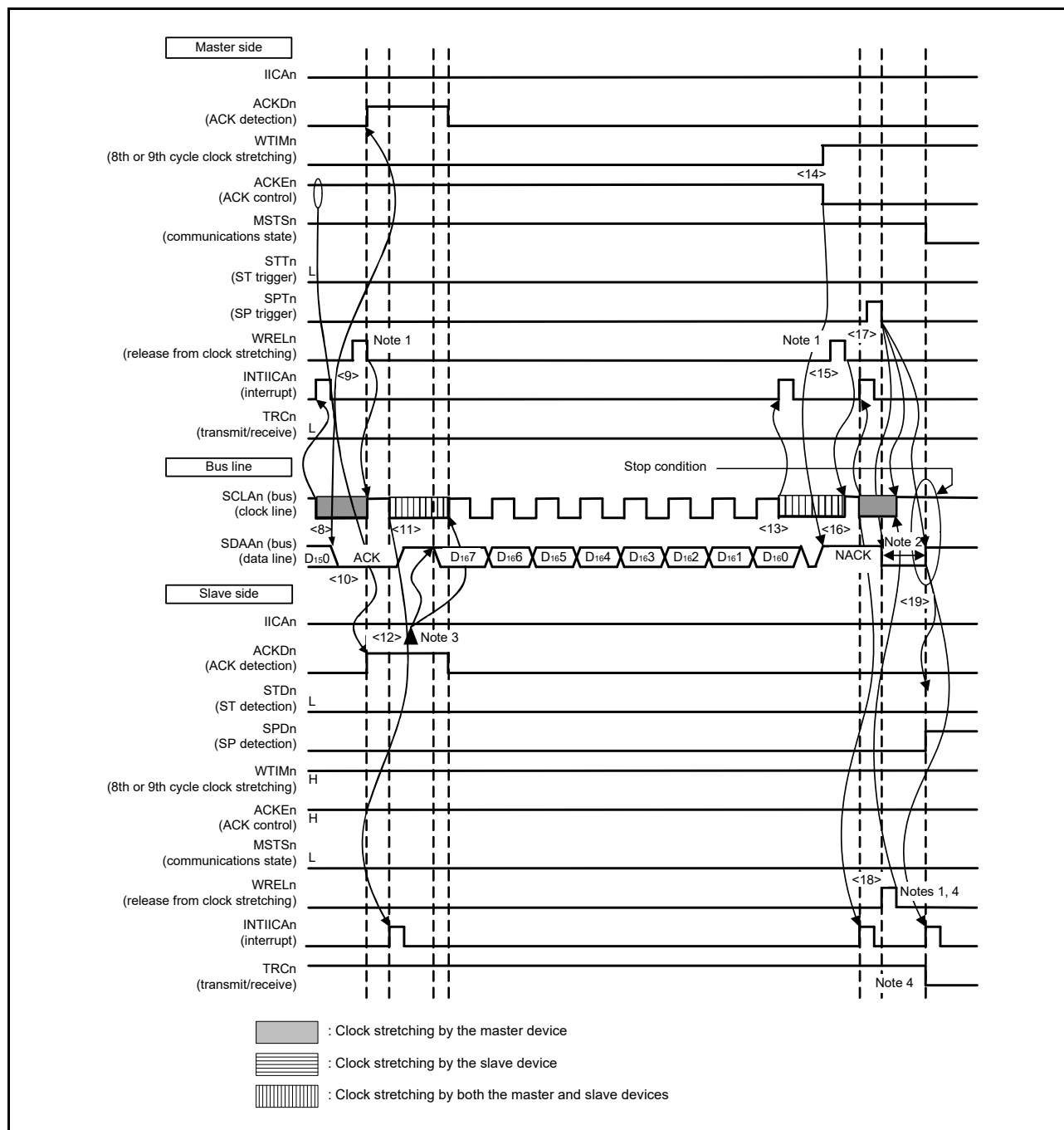
Remark 1. <1> to <19> in **Figure 20 - 33** represent the entire procedure for communicating data using the I²C bus.

1. **Start condition to address to data** in **Figure 20 - 33**: Steps <1> to <7>
2. **Address to data to data** in **Figure 20 - 33**: Steps <3> to <12>
3. **Data to data to stop condition** in **Figure 20 - 33**: Steps <8> to <19>

Remark 2. n = 0, 1

Figure 20 - 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, 1

The meanings of <8> to <19> in **3. Data to data to stop condition** in **Figure 20 - 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 20 - 33** represent the entire procedure for communicating data using the I²C bus.

- 1. Start condition to address to data** in **Figure 20 - 33**: Steps <1> to <7>
- 2. Address to data to data** in **Figure 20 - 33**: Steps <3> to <12>
- 3. Data to data to stop condition** in **Figure 20 - 33**: Steps <8> to <19>

Remark 2. $n = 0, 1$

Section 21 Serial Interface UARTA (UARTA)

The number of channels of the serial interface UARTA depends on the product.

Number of Pins	44	48 to 64		80 and 100
Flash Memory	64 to 512 Kbytes	64 to 128 Kbytes	256 to 512 Kbytes	128 to 512 Kbytes
Number of channels	2	2	3	4

21.1 Overview

The serial interface UARTAn 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:

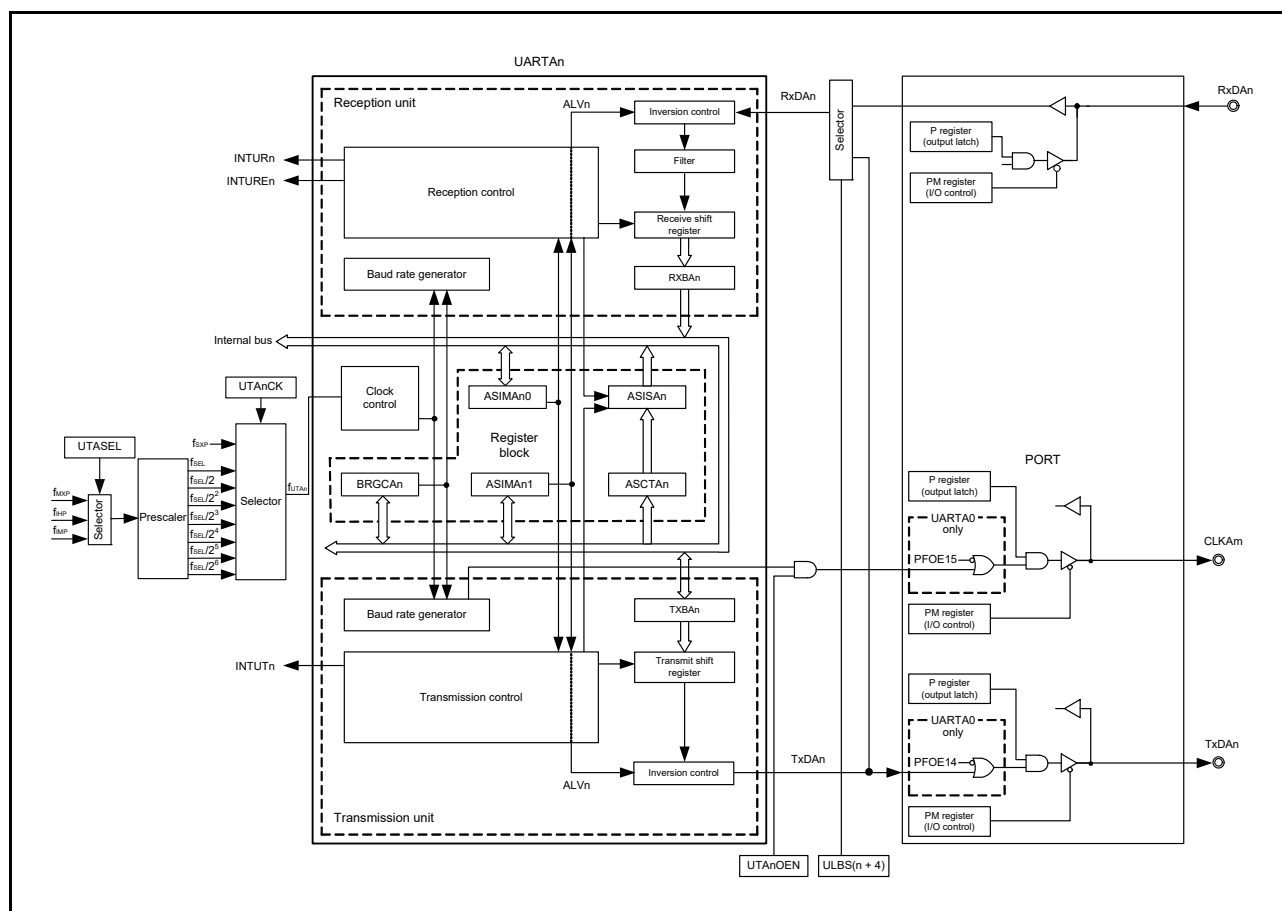
UARTAn handles asynchronous communications and has the following features.

- 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
- Transmit clock output is available (CLKAn)
- Operating clock independent of the CPU/peripheral hardware clock (fCLK) selectable

Remark n: Unit number n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

Figure 21 - 1 shows a block diagram of UARTAn and Table 21 - 1 shows the pin configuration of UARTAn.

Figure 21 - 1 Block Diagram of UARTAn



21.2 Registers for Controlling the Serial Interface UARTA

The following registers are used to control the serial interface UARTA.

- Peripheral enable registers 1, 2 (PER1, PER2)
- Transmit buffer register n (TXBAn)
- Receive buffer register n (RXBAn)
- Operation mode setting register n0 (ASIMAn0)
- Operation mode setting register n1 (ASIMAn1)
- Baud rate generator control register n (BRGCAn)
- Status register n (ASISAn)
- Status clear trigger register n (ASCTAn)
- UARTA clock select register n (UTAnCK)
- UART loopback select register (ULBS)
- Port mode registers (PMxx)
- Port registers (Pxx)
- 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 registers (PFOEx)
- LCD port function registers (PFSEGx)

Remark n: Unit number n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.1 Peripheral enable registers 1, 2 (PER1, PER2)

The PER1 or PER2 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 UARTAn, be sure to set bit 2 (UTA0EN) of the PER1 register and bits 4 to 2 (UTA3EN to UTA1EN) of the PER2 register to 1. The PER1 or PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PER1 or PER2 register following a reset is 00H.

Figure 21 - 2 Format of Peripheral Enable Registers 1, 2 (PER1, PER2)

Address: F00FAH (PER1), F00FCH (PER2)

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUEN

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN	UTA2EN	UTA1EN	TRJ1EN	TRJ0EN

UTAnEN	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 UTAnEN bit is 1 before using UARTAn. If UTAnEN = 0, writing to the registers which control UARTAn is ignored.

Caution 2. The functions mounted depend on the product. For details on the PER1 and PER2 registers, see Section 6 Clock Generator.

Caution 3. Be sure to clear bit 1 of the PER1 register to 0.

Caution 4. Be sure to clear bit 1 of the PER2 register to 0 in 44- to 64-pin products with 64 to 128 Kbytes of flash memory.

Caution 5. Be sure to clear bit 4 of the PER2 register to 0 in products with unit numbers (n) of 0 to 2.

Caution 6. Be sure to clear bits 4 and 3 of the PER2 register to 0 in products with unit numbers (n) of 0 and 1.

Remark n: Unit number
n = 0, 1 in 44-pin products
n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
n = 0 to 3 in 80- and 100-pin products

21.2.2 Transmit buffer register n (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 21 - 3 Format of Transmit Buffer Register n (TXBAn)

Address: F0300H (TXBA0), F0610H (TXBA1), F0630H (TXBA2), F0650H (TXBA3)
 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 1. 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.

Remark 2. n: Unit number
 n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.3 Receive buffer register n (RXBAn)

The RXBAn register holds 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.

Note The receive shift register converts the serial data that is input through the RxDA_n pin to parallel data. The receive shift register cannot be manipulated directly by a program.

Figure 21 - 4 Format of Receive Buffer Register n (RXBAn)

Address: F0301H (RXBA0), F0611H (RXBA1), F0631H (RXBA2), F0651H (RXBA3)

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.							

Caution If an overrun error (OVEAn) occurs, the data received at that time are not stored in the RXBAn register.

Remark n: Unit number

- n = 0, 1 in 44-pin products
- n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
- n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
- n = 0 to 3 in 80- and 100-pin products

21.2.4 Operation mode setting register n0 (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 a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 01H.

Figure 21 - 5 Format of Operation Mode Setting Register n0 (ASIMAn0)

Address: F0302H (ASIMA00), F0612H (ASIMA10), F0632H (ASIMA20), F0652H (ASIMA30)
 After reset: 01H
 R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
ASIMAn0	UARTAENn	TXEAn	RXEAn	0	0	0	ISSMAn	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.								
ISSMAn								
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 RxDAn 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 RXBAn registers are reset by clearing the UARTAENn bit to 0.

Caution 1. Be sure to clear bits 4 to 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 and Remark are listed 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 UARTAn operation clock (fUTAn). To enable transmission or reception again, set the TXEAn or RXEAn bit to 1 at least two cycles of the UARTAn operation clock after clearing the TXEAn or RXEAn bit to 0. If the bit is set to 1 within two cycles of the UARTAn 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 UARTAn 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.

Remark	n: Unit number	n = 0, 1 in 44-pin products n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory n = 0 to 3 in 80- and 100-pin products

21.2.5 Operation mode setting register n1 (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 a 1-bit or 8-bit memory manipulation instruction. The ASIMAn1 register must be modified while TXEAn = 0 and RXEAn = 0. The value of this register following a reset is 1AH.

Figure 21 - 6 Format of Operation Mode Setting Register n1 (ASIMAn1)

Address: F0303H (ASIMA01), F0613H (ASIMA11), F0633H (ASIMA21), F0653H (ASIMA31)

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 A parity check does not proceed when this setting is made. 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.

Remark n: Unit number
 n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.6 Baud rate generator control register n (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 21 - 7 Format of Baud Rate Generator Control Register n (BRGCAn)

Address: F0304H (BRGCA0), F0614H (BRGCA1), F0634H (BRGCA2), F0654H (BRGCA3)

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 21 - 2 .							

Caution Modify the BRGCAn bits while the TXEAn and RXEAn bits are 0 (in the transmission/reception stopped state).

Table 21 - 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	×	×	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, ..., 255)

Remark 2. ×: Don't care.

For an example of the baud rate setting, see **21.3.4 3. c) Baud rate setting example**.

Remark 3. n: Unit number
 n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.7 Status register n (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 UARTEAn 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 UARTEAn or TXEAn bit to 0.

Figure 21 - 8 Format of Status Register n (ASISAn) (1/2)

Address: F0305H (ASISA0), F0615H (ASISA1), F0635H (ASISA2), F0655H (ASISA3)

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 UARTEAn 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 UARTEAn 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 UARTEAn 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 UARTEAn or RXEAn bit is cleared to 0. • 1 is written to the FECTAn bit.

Figure 21 - 8 Format of Status Register n (ASISAn) (2/2)

OVEAn	Overrun 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. 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, checking of the TXBFAn flag is not required when continuous transmission proceeds by using the buffer empty interrupt (ISSMAAn bit = 1).

Caution 2. 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 3. The operation of the PEAn bit depends on the setting of the PSn1 and PSn0 bits of the ASIMAn1 register.

Caution 4. For the receive data, only the first 1 bit of the stop bits is checked regardless of the stop bit length.

Caution 5. When an overrun error occurs, the next receive data is not written to the RXBAn register and discarded.

Remark n: Unit number n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.8 Status clear trigger register n (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 PEAn, FEAn, and OVEAn bits of the ASISAn register, respectively. When writing 0, the corresponding error flags are not cleared.

Figure 21 - 9 Format of Status Clear Trigger Register n (ASCTAn)

Address: F0306H (ASCTA0), F0616H (ASCTA1), F0636H (ASCTA2), F0656H (ASCTA3)

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 PEAn flag. (The flag is retained.)
1	Clears the PEAn 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 (f_{UTA}n). Accordingly, if reading the ASISAn register immediately after writing 1 to the trigger bit, the corresponding error flag may not have been cleared yet.

Remark n: Unit number
 n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

21.2.9 UARTA clock select register n (UTAnCK)

The UTAnCK register selects the operating clock of the UARTAn. The UTASEL1 and UTASEL0 bits select the clock source, fSEL, for UARTAn from fMXP, fIH, and fIMP. The bits from UTA0CK3 to UTA0CK0 select the operating clock for UARTAn from fSEL/1 to fSEL/64, fSXP, and ELCL. Setting the UTAnOEN bit to 1 selects the UARTAn clock output mode, and the operating clock for UARTAn is output from the CLKAm pin. The UTAnCK 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 21 - 10 Format of UARTA Clock Select Register n (UTAnCK) (1/2)

Address: F0310H (UTA0CK), F0620H (UTA1CK), F0640H (UTA2CK), F0660H (UTA3CK)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
UTA0CK	UTA0OEN	0	UTA0SEL1	UTA0SEL0	UTA0CK3	UTA0CK2	UTA0CK1	UTA0CK0

Symbol	7	6	5	4	3	2	1	0
UTA1CK	UTA1OEN	0	UTA1SEL1	UTA1SEL0	UTA1CK3	UTA1CK2	UTA1CK1	UTA1CK0

Symbol	7	6	5	4	3	2	1	0
UTA2CK	0	0	UTA2SEL1	UTA2SEL0	UTA2CK3	UTA2CK2	UTA2CK1	UTA2CK0

Symbol	7	6	5	4	3	2	1	0
UTA3CK	0	0	UTA3SEL1	UTA3SEL0	UTA3CK3	UTA3CK2	UTA3CK1	UTA3CK0

UTAmOEN	UARTAn clock output function enable						
0	Disables CLKAm output.						
1	Enables CLKAm output.						

UTAnSEL1	UTAnSEL0	fSEL clock select
0	0	Stop
0	1	fMXP
1	0	fIHP
1	1	fIMP

Figure 21 - 10 Format of UARTA Clock Select Register n (UTAnCK) (2/2)

UTAnCK3	UTAnCK2	UTAnCK1	UTAnCK0	UARTAn operation clock select (fUTAn)
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
1	0	0	1	ELCL
Other than above				Setting prohibited

Caution 1. This register should be read or written when the TXEAn and RXEAn bits are 0 (in the transmission/reception stopped state).

Caution 2. Clear UTA1OEN to 0 because the CLKA1 output pin is not present in 44- to 64-pin products.

Caution 3. Be sure to clear bit 7 of the UTA2CK and UTA3CK registers to 0.

Caution 4. When the setting of the FRQSEL4 bit of the option byte is 1 and that of the UTASEL1 and UTASEL0 bits is for fHP as the fSEL clock, setting the UTA0CK3 to UTA0CK0 bits to 0000B is prohibited.

Remark 1. fMXP: High-speed peripheral clock
 fHP: High-speed on-chip oscillator peripheral clock
 fMP: Middle-speed on-chip oscillator peripheral clock
 fSXP: Low-speed peripheral clock
 fSEL: Selected clock to be divided for the UARTA

Remark 2. n: Unit number n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products
 m: CLKA pin number m = 0 in 44- to 64-pin products
 m = 0, 1 in 80- and 100-pin products

21.2.10 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 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 21 - 11 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	ULBS7	ULBS6	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0

ULBS7	Selection of the UART loopback function
0	The state of the RxDA3 pin of the serial interface UARTA3 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

ULBS6	Selection of the UART loopback function
0	The state of the RxDA2 pin of the serial interface UARTA2 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

ULBS5	Selection of the UART loopback function
0	The state of the RxDA1 pin of the serial interface UARTA1 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

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 When using the loopback function in UARTA0, set the PFOE14 bit of the port function output enable register 1 (PFOE1) to 1.

Remark The RxDA2 pin can only be used in 48- to 64-pin products with 256 to 512 Kbytes of flash memory or in 80- and 100-pin products. The RxDA3 pin can only be used in 80- and 100-pin products.

21.2.11 Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface.

- Port mode registers (PMxx)
- Port registers (Pxx)
- 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 registers (PFOEx)
- LCD port function registers (PFSEGx)

For details, 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)**
- **4.3.14 Port function output enable registers (PFOEx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting registers for controlling the port functions, see **4.5.4 Examples of register settings for port and alternate functions**.

When connecting 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**.

21.3 Operation

UARTAn operates in the following two modes.

- Operation stop mode
- UART mode

21.3.1 Operation stop mode

In the operation stop mode, serial communication does not proceed, 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 to 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 UTAnEN bit of the PER1 and PER2 registers to 0 after the above setting.

Remark n: Unit number n = 0, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

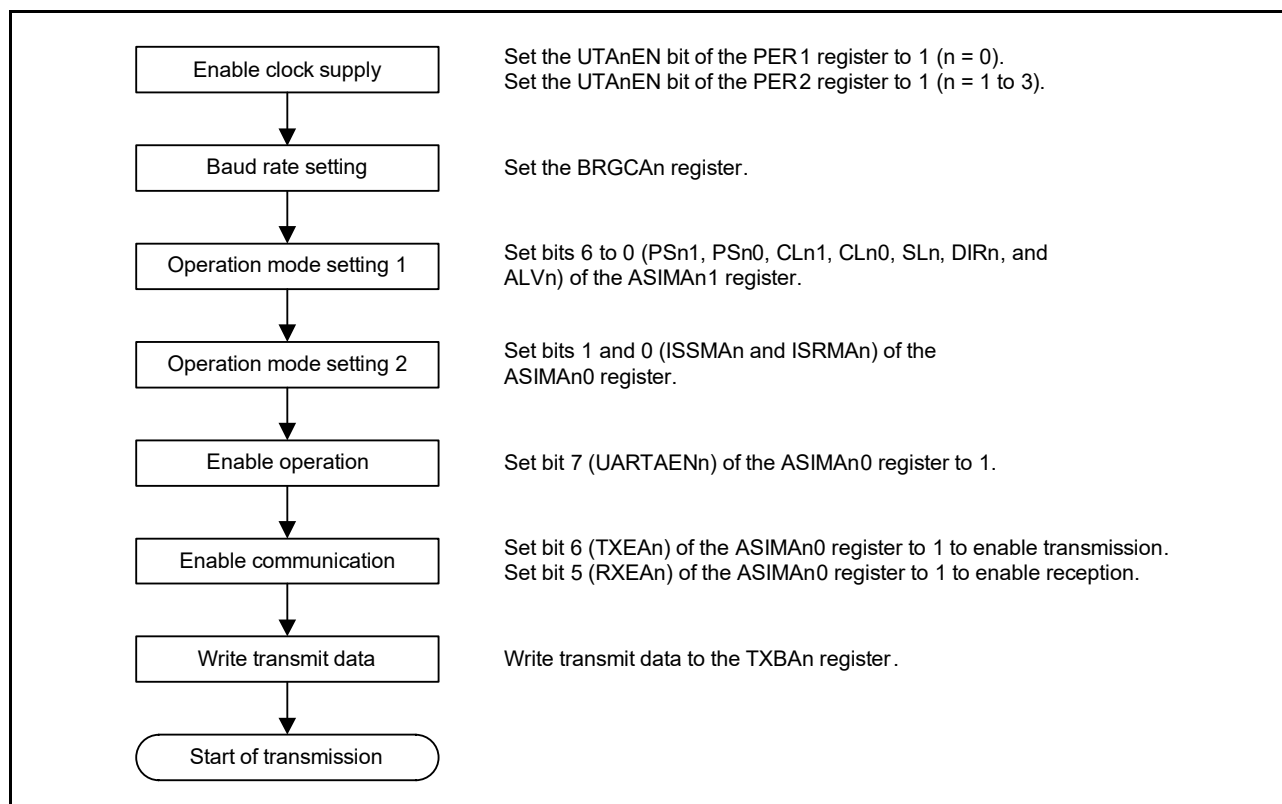
21.3.2 UART mode

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 21 - 12 shows the flowchart of communication procedure.

Figure 21 - 12 Flowchart of Communication Procedure



Caution When using the receiving function, set the port pin allocated for reception to the input mode by using port related registers. When using the transmitting function, set the initial output level of the port pin allocated for transmission by using the port registers (Pxx), then set that port pin to the output mode by using port related registers. These settings must be made before enabling transmission. The port related registers that need to be set differ depending on the channel or product to be used. For details, see 4.5 Register Settings When Using Alternate Function. When connecting 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.

Remark n: Unit number

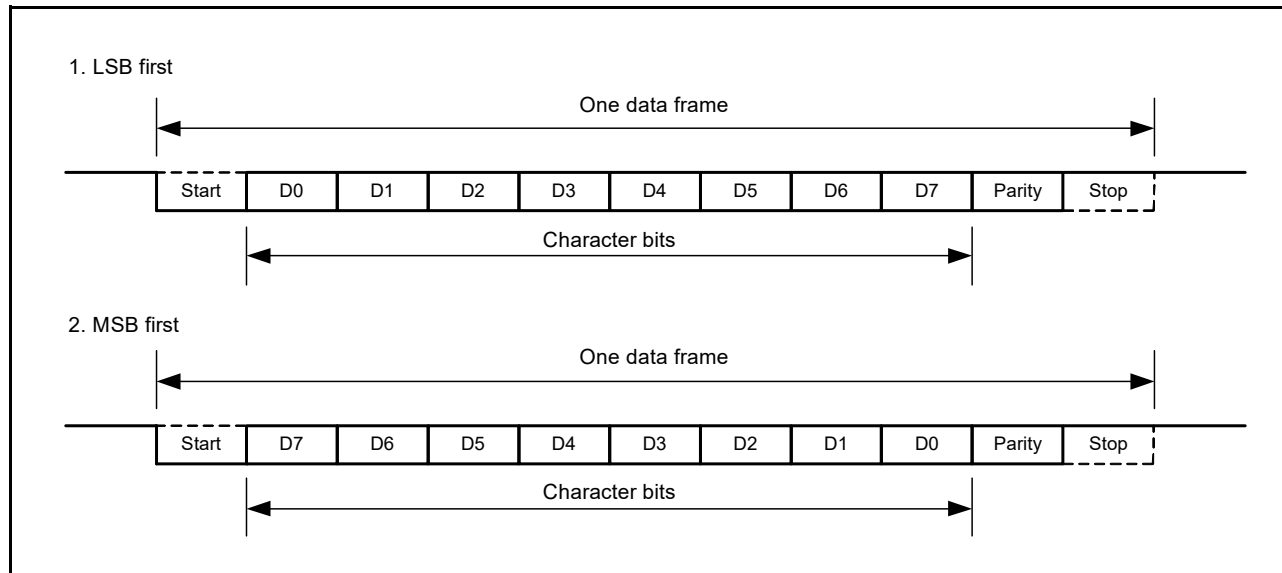
- n = 0, 1 in 44-pin products
- n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
- n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
- n = 0 to 3 in 80- and 100-pin products

2. Format and waveform example of transmit/receive data

The following describes the communication data format of UARTAn.

Figure 21 - 13 shows the data format.

Figure 21 - 13 Transmit/Receive Data Format



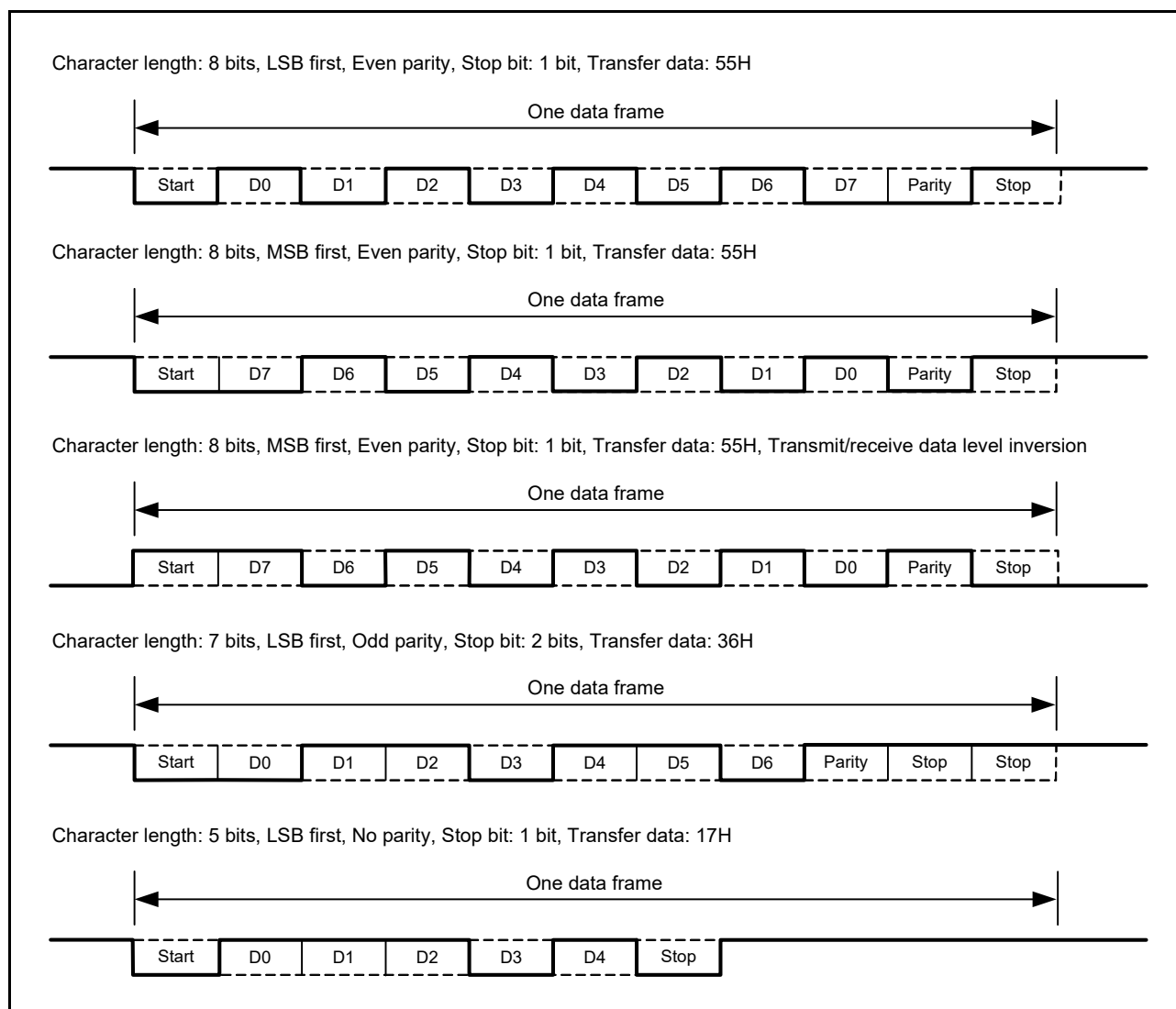
One data frame consists of the following bits.

- 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 21 - 14 shows the examples of transmit/receive data waveforms.

Figure 21 - 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 proceeds assuming that there is no parity bit. A parity error does not occur, because there is no parity bit.

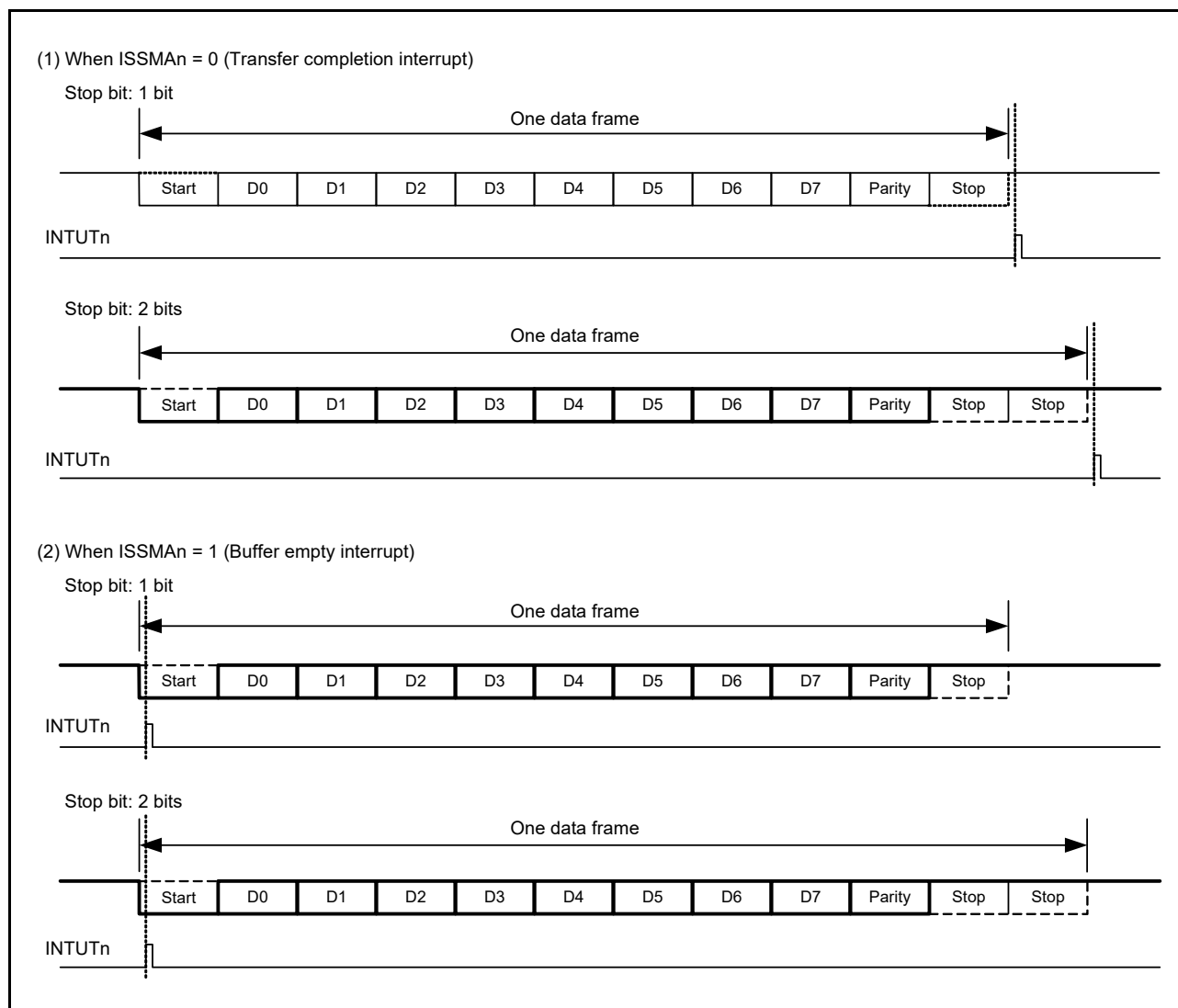
4. Normal transmission

Transmission is enabled by setting bit 7 (UARTAENn) of the operation mode setting register n0 (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 transmit buffer register n (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 21 - 15 shows the timing of the transfer completion interrupt request signal (INTUTn). INTUTn is issued at the following timing.

- a) When ISSMA_n = 0 (INTUT_n functions as a transfer completion interrupt.)
INTUT_n is issued after the output of the last stop bit.
- b) ISSMA_n = 1 (INTUT_n functions as a buffer empty interrupt.)
INTUT_n is issued when the start bit is output.

Figure 21 - 15 Interrupt Output Timing



5. Continuous transmission

UARTAn has two separate registers for continuous transmission: the transmit buffer register n (TXBAn) and the transmit shift register.

Once the transmit shift register starts a shift operation, the next transmit data can be written to the transmit buffer register n (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, ..., 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 n (ASISAn).

When using this method, clear bit 1 (ISSMAAn) of the operation mode setting register n0 (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.

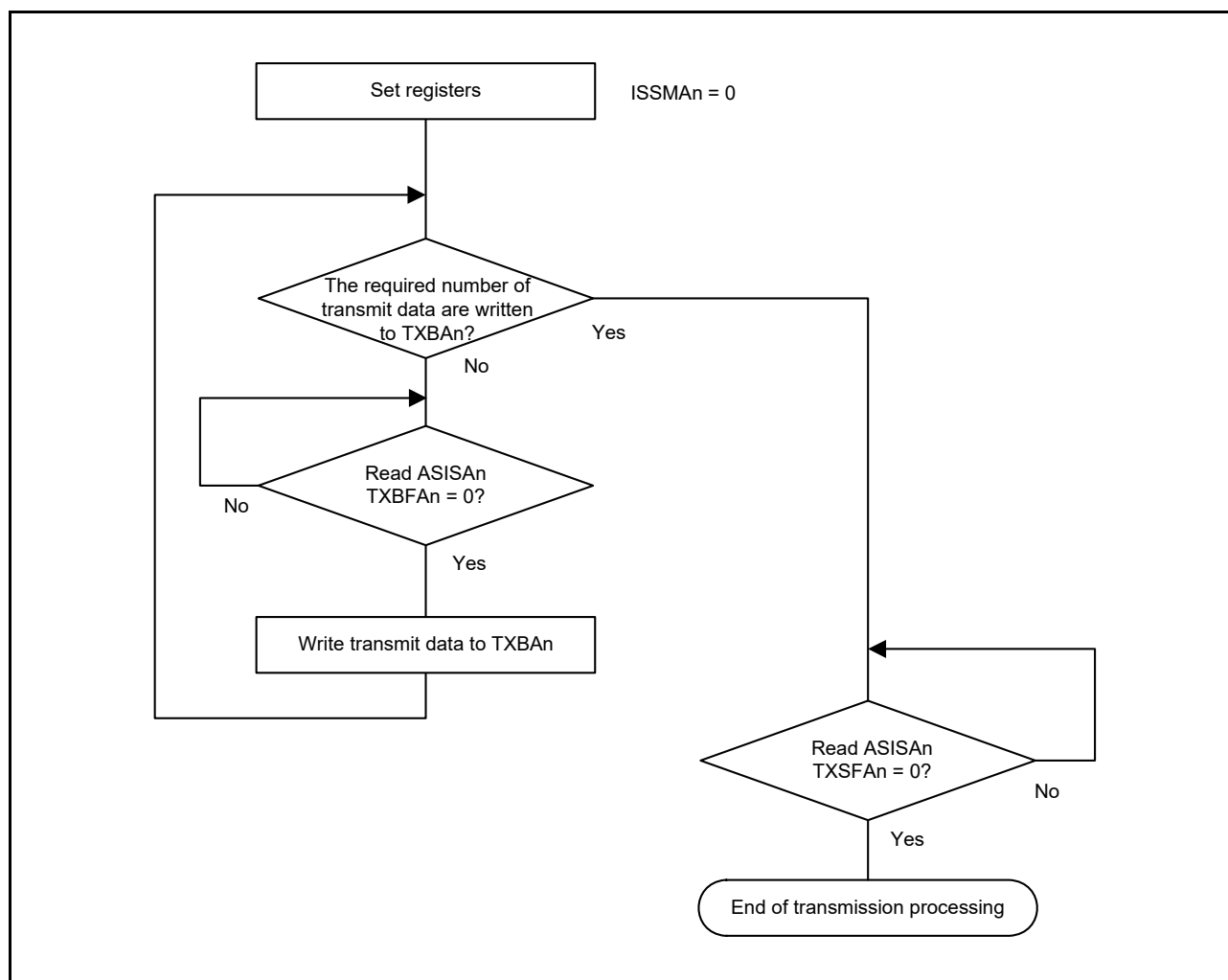
Caution 1. When initializing the transmission unit after completion of continuous transmission, check that the TXSFAn flag is 0 after the transfer completion interrupt is generated, and then initialize the unit.

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 21 - 16 shows a flow example of continuous transmission processing by polling.

Figure 21 - 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 n (TXBAn) are transferred to the transmit shift register by setting bit 1 (ISSMA_n) to 1 in the operation mode setting register n0 (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 21 - 17 shows a flow example of continuous transmission using interrupt.

Figure 21 - 17 Flow Example of Continuous Transmission Using Interrupt

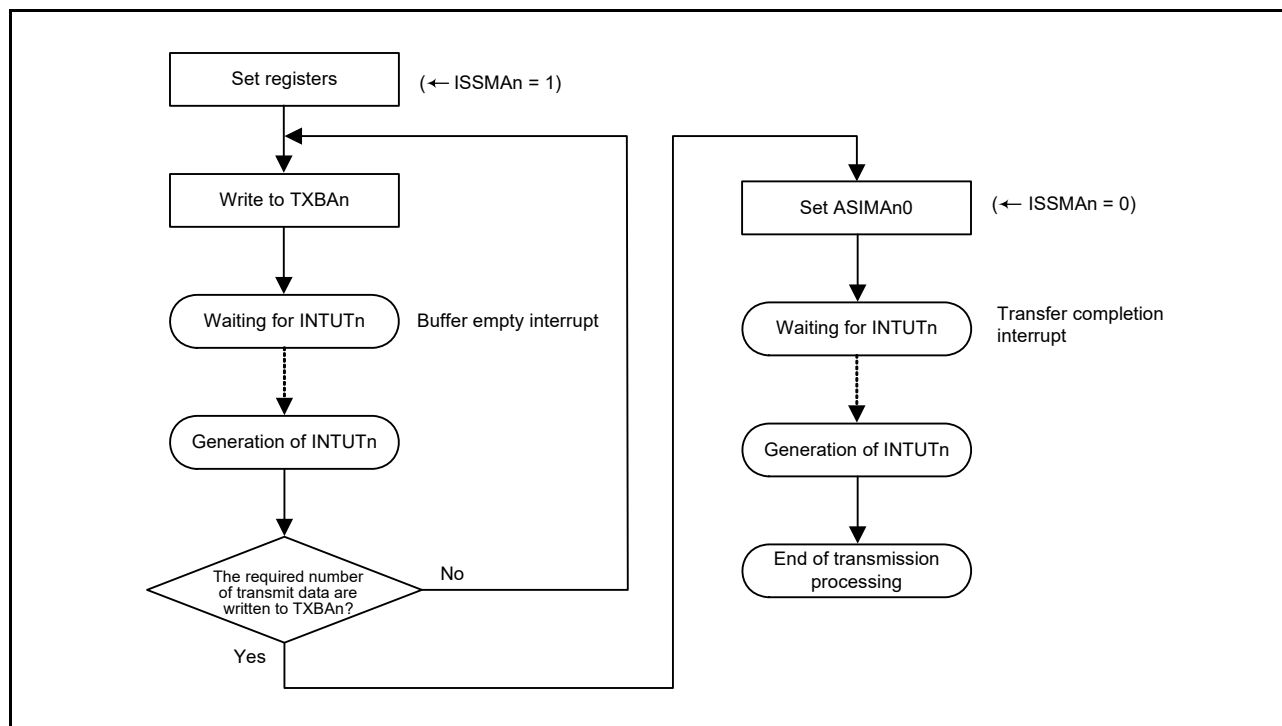
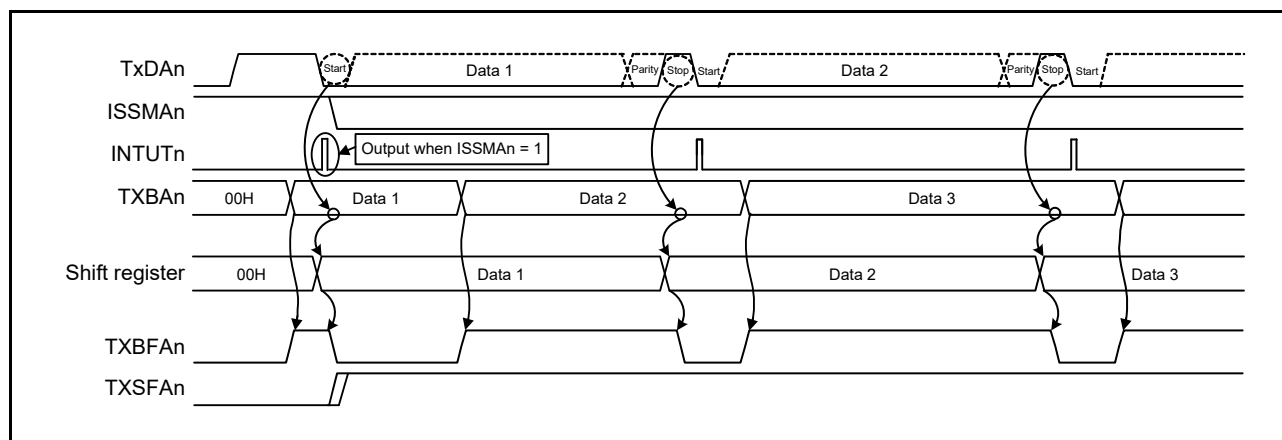


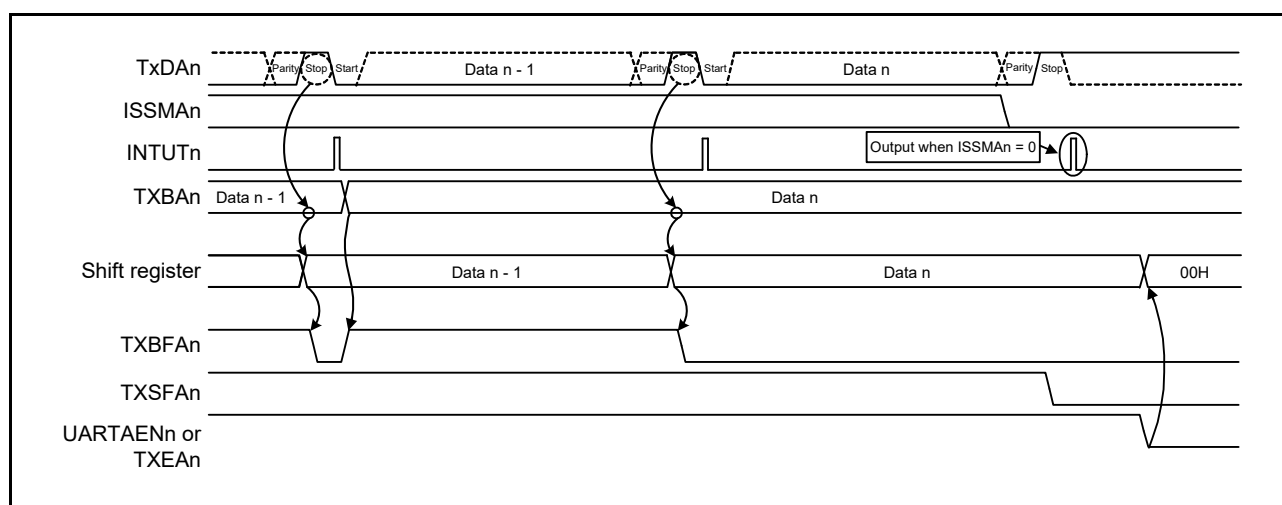
Figure 21 - 18 and **Figure 21 - 19** show the timing charts when the continuous transmission is started and completed, respectively.

Figure 21 - 18 Timing Chart When Continuous Transmission Is Started



Caution When the ASISAn register is read, both the TXBFA[n] and TXSFA[n] bits are read as 1 within this period. Accordingly, use only the TXBFA[n] flag to determine if writing is enabled or disabled.

Figure 21 - 19 Timing Chart When Continuous Transmission Is Completed



Remark n: Unit number

n = 0, 1 in 44-pin products

n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory

n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory

n = 0 to 3 in 80- and 100-pin products

6. Normal reception

Setting bit 7 (UARTAENn) of the operation mode setting register n0 (ASIMAn0) to 1, and setting bit 5 (RXEAn) of the ASIMAn0 register to 1 enables reception, and sampling of the input to the RxDAn pin proceeds.

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 n (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 21 - 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 n (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 21 - 20**). If the RxDAn pin is high, it is regarded as a start bit.

Figure 21 - 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 n (RXBAn).

Note that when an overrun error (OVEAn) occurs, the data received on occurrence of the error is not written to the RXBAn register.

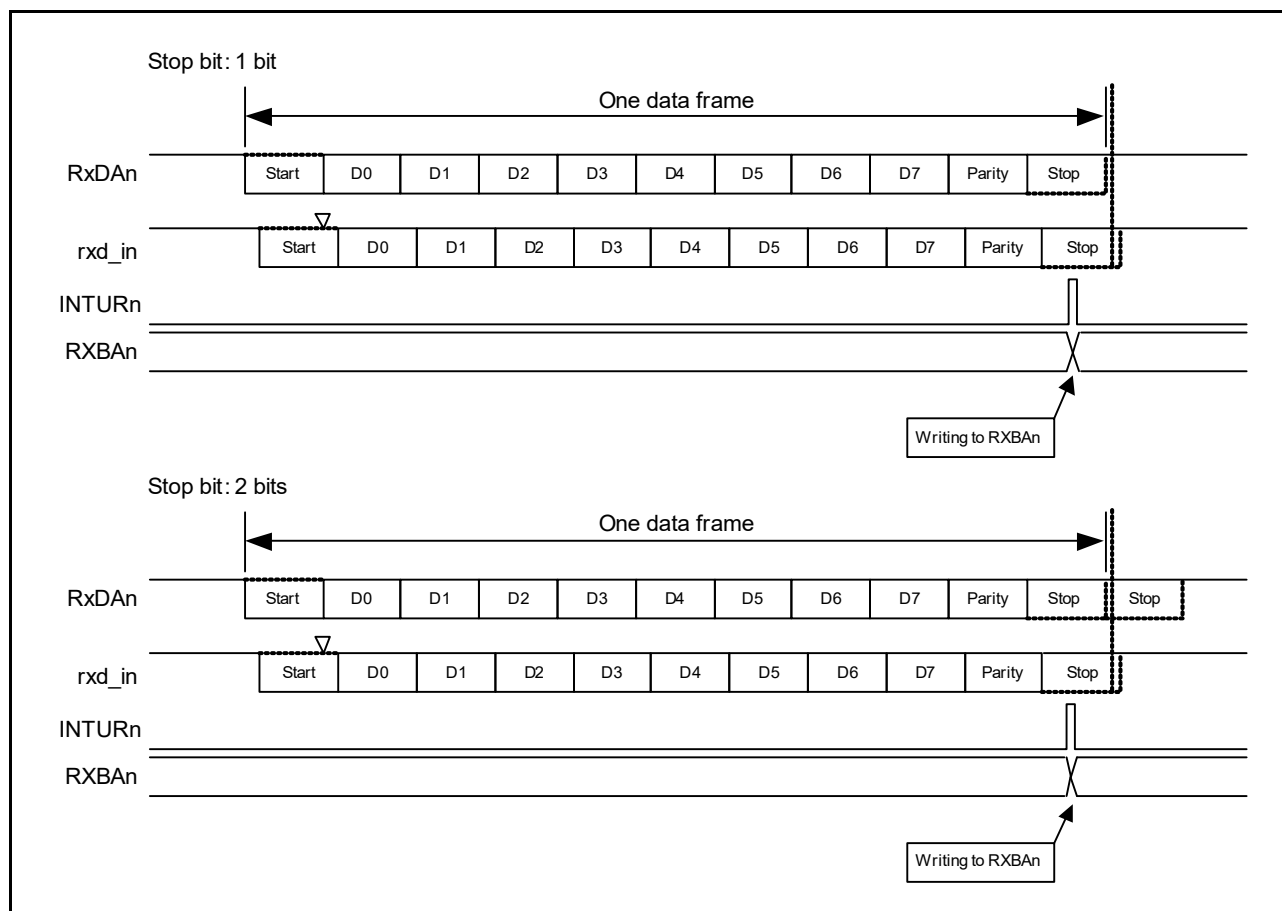
When a parity error (PEAn) or a framing error (FEAn) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (INTURn/INTUREn) set in the ISRMAAn bit is generated.

When a reception error occurs, read the status register n (ASISAn) and then read the receive buffer register n (RXBAn) to clear the error flag.

If the receive buffer register n (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 21 - 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 n (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 n (ASISAn).

The contents of the ASISAn register are cleared to 0 by setting the corresponding bit of the status clear trigger register n (ASCTAn) to 1.

Table 21 - 3 shows the causes of the reception errors.

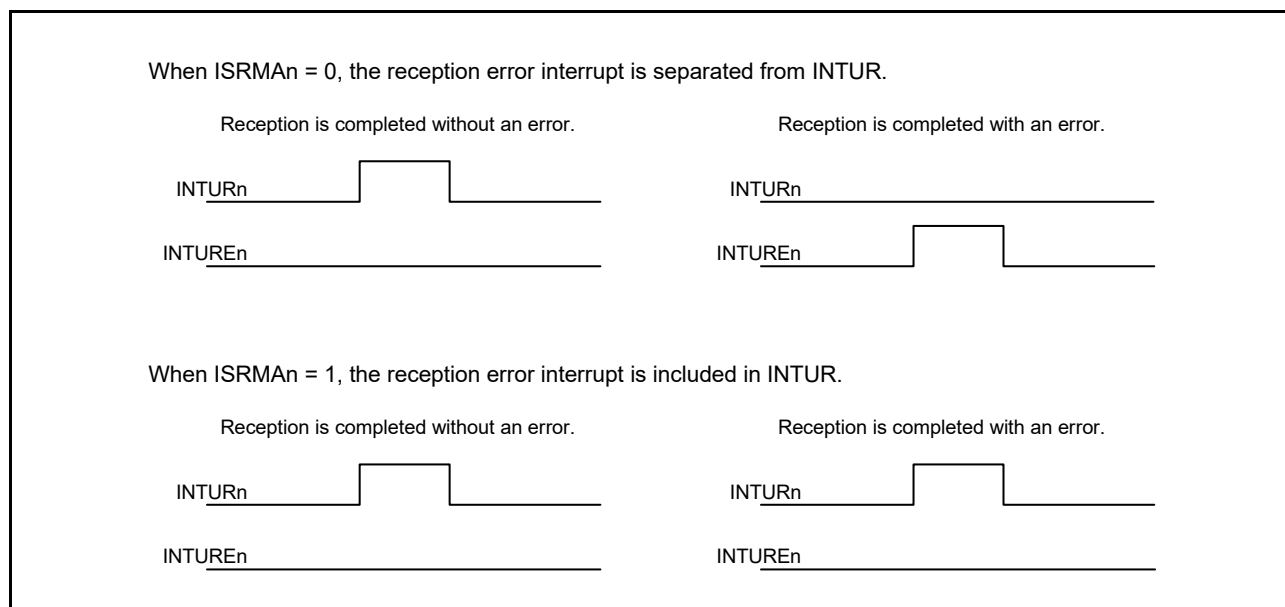
Table 21 - 3 Causes of Reception Errors

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 (ISRMA_n) of the operation mode setting register n0 (ASIMAn0) to 0 allows the reception error interrupt to be separated from INTUR_n and allows it to be generated as INTURE_n.

Figure 21 - 21 shows the interrupt output waveform which varies depending on the setting of the ISRMA_n bit.

Figure 21 - 21 Various Interrupt Output Waveforms Depending on ISRMA_n Setting



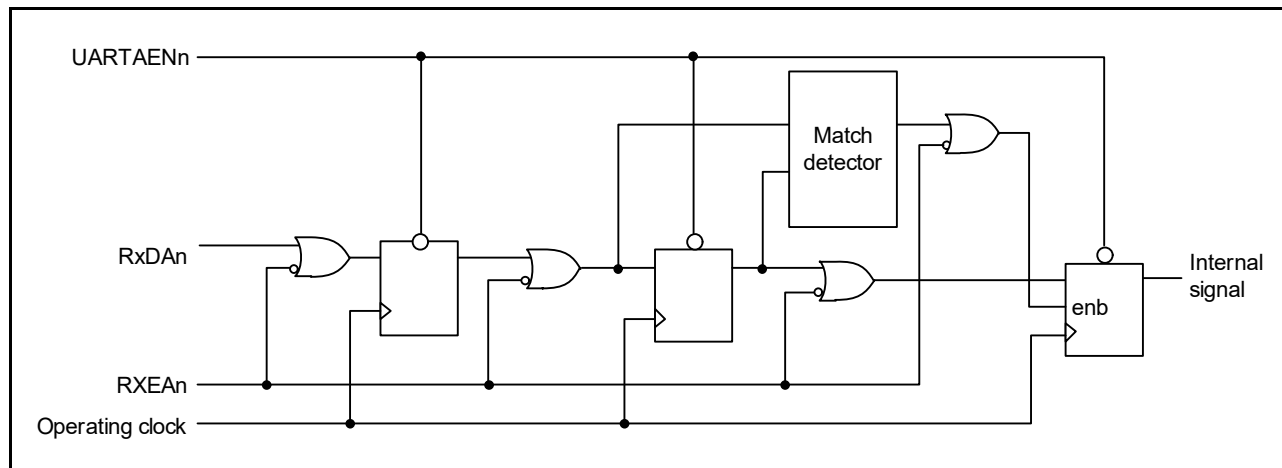
21.3.3 Receive data noise filter

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 21 - 22 shows the noise filter circuit.

Figure 21 - 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, 1 in 44-pin products

n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory

n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory

n = 0 to 3 in 80- and 100-pin products

21.3.4 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 n0 (ASIMAn0), the UARTAn operation clock (fUTAn) 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 n0 (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 n (TXBAn). When continuous transmission proceeds, 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

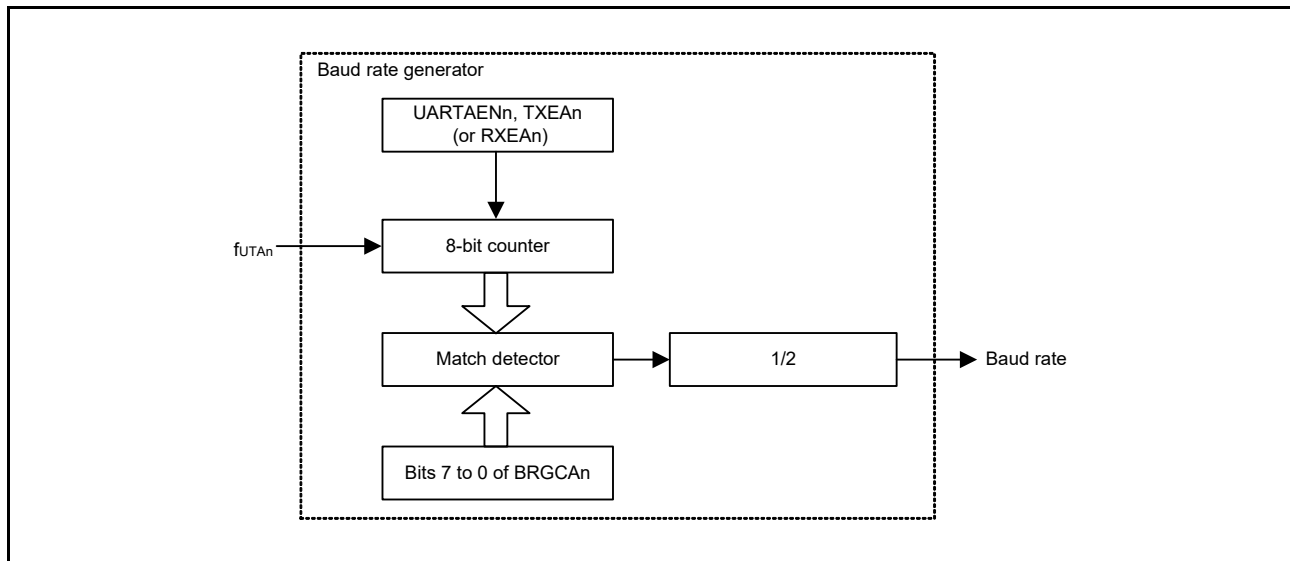
This counter is cleared to 0 and stops when bit 7 (UARTAENn) = 0 or bit 5 (RXEAn) = 0 in the operation mode setting register n0 (ASIMAn0). It starts counting when the start bit is detected.

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, 1 in 44-pin products
 n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
 n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
 n = 0 to 3 in 80- and 100-pin products

Figure 21 - 23 shows the configuration of the baud rate generator.

Figure 21 - 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 n (BRGCAn).

The baud rate generator divides the frequency of the input clock signal to the 8-bit counter (fUTAn) 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.

$$\text{Baud rate} = f_{UTAn} \div (2 \times k) \text{ [bps]}$$

fUTAn: Frequency of operating clock

k: Value set by bits 7 to 0 of the BRGCAn register (k = 2, 3, 4, ..., 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 21.3.4 3. d) Permissible baud rate range during reception.

c) Baud rate setting example

Table 21 - 4 Set Data of Baud Rate Generator (1/4)

Desired baud rate	In operation with fHP = 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 n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number
n = 0, 1 in 44-pin products
n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
n = 0 to 3 in 80- and 100-pin products

Table 21 - 4 Set Data of Baud Rate Generator (2/4)

Desired baud rate	In operation with fMIP = 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 n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number
n = 0, 1 in 44-pin products
n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
n = 0 to 3 in 80- and 100-pin products

Table 21 - 4 Set Data of Baud Rate Generator (3/4)

Desired baud rate	In operation with f _{MP} = 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 n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number
n = 0, 1 in 44-pin products
n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
n = 0 to 3 in 80- and 100-pin products

Table 21 - 4 Set Data of Baud Rate Generator (4/4)

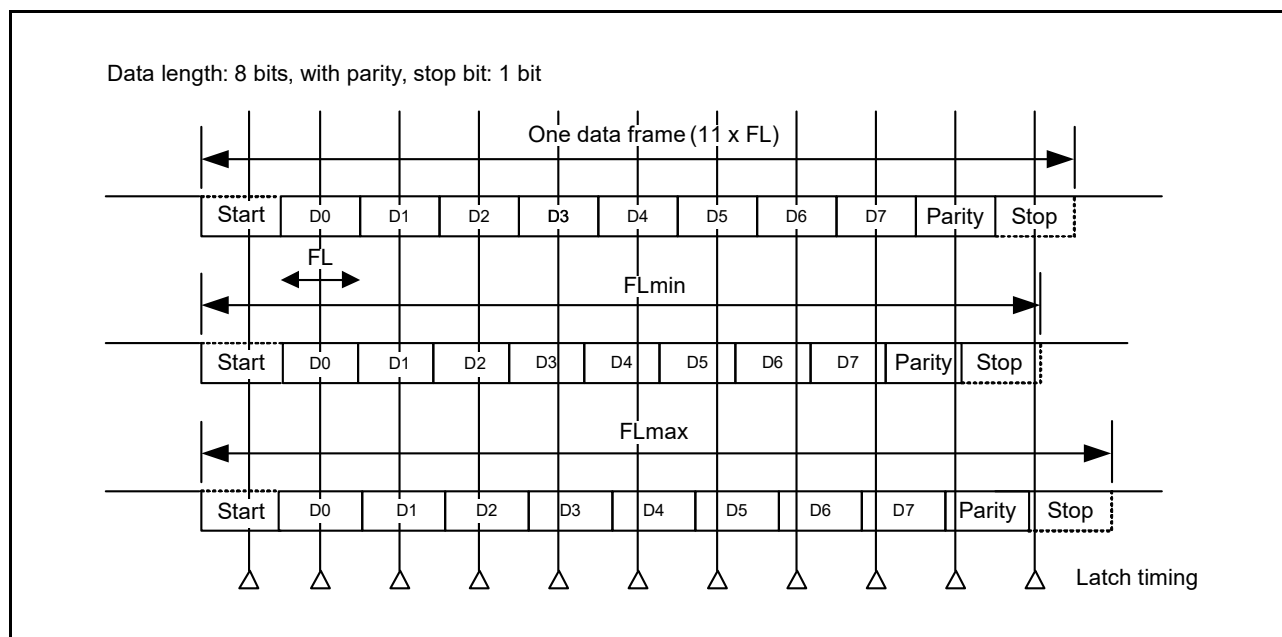
Desired baud rate	In operation with f _{SXP} = 32.768 kHz (UTAnCK3 to UTAnCK0 = 1000B)		In operation with ELCL = 32 MHz (UTAnCK3 to UTAnCK0 = 1001B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	82	-0.10%	Disabled	
300 bps	55	-0.70%	Disabled	
600 bps	27	-1.14%	Disabled	
1200 bps	14	-2.48%	Disabled	
2400 bps	7	-2.48%	Disabled	
4800 bps	Disabled		Disabled	
9600 bps	Disabled		Disabled	
19200 bps	Disabled		Disabled	
38400 bps	Disabled		Disabled	
76800 bps	Disabled		208	0.16%
115200 bps	Disabled		139	-0.08%
153600 bps	Disabled		104	0.16%

Remark k: Value set by bits 7 to 0 of the baud rate generator control register n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number
n = 0, 1 in 44-pin products
n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory
n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory
n = 0 to 3 in 80- and 100-pin products

- d) Permissible baud rate range during reception

Figure 21 - 24 shows the permissible error from the baud rate on the transmitting side during reception.

Figure 21 - 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 n (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

<R> Margin of latch timing: 2 clock cycles

- Minimum permissible data frame length (FLmin)

$$\text{When } k = 3 \text{ to } 255: FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

- Maximum permissible baud rate for reception on the transmitting side (BRmax)

$$\text{When } k = 2: BR_{\max} = Brate + \frac{1}{22k} Brate$$

$$\text{When } k = 3 \text{ to } 255: BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} Brate$$

- Maximum permissible data frame length (FLmax)

$$\text{When } k = 3 \text{ to } 255: FL_{\max} = \frac{21k+2}{20k} FL \times 11$$

- Minimum permissible baud rate for reception on the transmitting side (BRmin)

$$\text{When } k = 2: BR_{\min} = Brate - \frac{1}{22k} Brate$$

$$\text{When } k = 3 \text{ to } 255: BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} Brate$$

Table 21 - 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.

<R> Table 21 - 5 Maximum/Minimum Permissible Baud Rate Error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.27%	-2.27%
4	+2.33%	-2.44%
8	+3.53%	-3.61%
20	+4.27%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

(Remarks are listed on the next page.)

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, 1 in 44-pin products

n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory

n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory

n = 0 to 3 in 80- and 100-pin products

21.4 Points for Caution when the Serial Interface UARTA is to be Used

21.4.1 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 $UARTAEN_n = 1$.

21.4.2 Serial Interfaces

The serial data output (TxDA0) and serial clock output (CLKA0) of UARTA0 are selectable as inputs for the logic and event link controller (ELCL). If TxDA0 or CLKA0 is to be used as an input for the ELCL, you need to clear the corresponding bit in port function output enable register 1 (PFOE1) to 0 so that the signal is not output to the TxDA0 and CLKA0 pins. For details, see **4.3.14 Port function output enable registers (PFOEx)**.

21.4.3 Point for caution when selecting the UARTAn operation clock (fUTAn)

When the middle-speed on-chip oscillator peripheral clock (fIMP) 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, 1 in 44-pin products n = 0, 1 in 48- to 64-pin products with 64 to 128 Kbytes of flash memory n = 0 to 2 in 48- to 64-pin products with 256 to 512 Kbytes of flash memory n = 0 to 3 in 80- and 100-pin products
	m: CLKA pin number	m = 0 in 44- to 64-pin products m = 0, 1 in 80- and 100-pin products

Section 22 LCD Controller/Driver

The number of LCD display function pins of the RL78/L23 depends on the product. The following table shows the number of pins of each product.

Table 22 - 1 Number of LCD Display Function Pins of Each Product (1/2)

a) 80- and 100-pin products

Item	80 pins (R7F100LMx (x = L, J, G))								100 pins (R7F100LPx (x = L, J, G))							
LCD controller/ driver	Segment signal outputs: 49 (45) ^{Note} Common signal outputs: 8								Segment signal outputs: 56 (52) ^{Note} Common signal outputs: 8							
Multiplexed I/O port	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43
P1	SEG 42	SEG 41	SEG 40	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35	SEG 42	SEG 41	SEG 40	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35
P2	SEG 34	SEG 33	SEG 32	SEG 31	SEG 30	SEG 29	—	—	SEG 34	SEG 33	SEG 32	SEG 31	SEG 30	SEG 29	—	—
P3	—	—	SEG 25	SEG 24	—	—	SEG 21	SEG 20	—	—	SEG 25	SEG 24	SEG 23	SEG 22	SEG 21	SEG 20
P4	SEG 27	SEG 26	—	—	—	—	—	—	SEG 27	SEG 26	—	—	—	—	—	—
P5	SEG 11	SEG 10	SEG 9	SEG 8	SEG 7	SEG 6	SEG 5	SEG 4	SEG 11	SEG 10	SEG 9	SEG 8	SEG 7	SEG 6	SEG 5	SEG 4
P7	SEG 19	SEG 18	SEG 17	SEG 16	SEG 15	SEG 14	SEG 13	SEG 12	SEG 19	SEG 18	SEG 17	SEG 16	SEG 15	SEG 14	SEG 13	SEG 12
P9	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0
P13	—	—	—	—	—	—	—	SEG 28	—	—	—	—	—	—	—	SEG 28
P14	—	—	—	—	—	—	—	—	—	—	SEG 55	SEG 54	SEG 53	SEG 52	SEG 51	—

Note The number in parentheses indicates the number of signal outputs when 8 com is in use.

Table 22 - 1 Number of LCD Display Function Pins of Each Product (2/2)

b) 52- and 64-pin products

Item	52 pins (R7F100LJx (x = L, J, G, E))								64 pins (R7F100LLx (x = L, J, G, E))							
LCD controller/ driver	Segment signal outputs: 25 (21) ^{Note} Common signal outputs: 8								Segment signal outputs: 34 (30) ^{Note} Common signal outputs: 8							
Multiplexed I/O port	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	SEG 50	SEG 49	SEG 48	—	—	—	—	SEG 43	SEG 50	SEG 49	SEG 48	SEG 47	SEG 46	SEG 45	SEG 44	SEG 43
P1	SEG 42	SEG 41	—	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35	SEG 42	SEG 41	SEG 40	SEG 39	SEG 38	SEG 37	SEG 36	SEG 35
P2	SEG 34	SEG 33	—	—	—	SEG 29	—	—	SEG 34	SEG 33	—	—	—	SEG 29	—	—
P3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEG 21	SEG 20
P5	SEG 11	—	—	SEG 8	SEG 7	—	—	—	SEG 11	—	—	SEG 8	SEG 7	SEG 6	—	—
P7	SEG 19	SEG 18	SEG 17	SEG 16	—	—	—	—	SEG 19	SEG 18	SEG 17	SEG 16	—	—	—	SEG 12
P9	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0
P13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
P14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note The number in parentheses indicates the number of signal outputs when 8 com is in use.

c) 44- and 48-pin products

Item	44 pins (R7F100LFx (x = L, J, G, E))								48 pins (R7F100LGx (x = L, J, G, E))							
LCD controller/ driver	Segment signal outputs: 19 (15) ^{Note} Common signal outputs: 8								Segment signal outputs: 22 (18) ^{Note} Common signal outputs: 8							
Multiplexed I/O port	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	SEG 50	SEG 49	SEG 48	—	—	—	—	SEG 43	SEG 50	SEG 49	SEG 48	—	—	—	—	SEG 43
P1	SEG 42	SEG 41	—	—	—	SEG 37	SEG 36	SEG 35	SEG 42	SEG 41	—	—	SEG 38	SEG 37	SEG 36	SEG 35
P2	SEG 34	SEG 33	—	—	—	SEG 29	—	—	SEG 34	SEG 33	—	—	—	SEG 29	—	—
P3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
P5	SEG 11	—	—	—	—	—	—	—	SEG 11	—	—	SEG 8	—	—	—	—
P7	SEG 19	SEG 18	—	—	—	—	—	—	SEG 19	SEG 18	SEG 17	—	—	—	—	—
P9	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0	SEG3/ COM7	SEG2/ COM6	SEG1/ COM5	SEG0/ COM4	COM 3	COM 2	COM 1	COM 0
P13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
P14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note The number in parentheses indicates the number of signal outputs when 8 com is in use.

22.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the RL78/L23 microcontrollers are as follows.

1. Waveform A or B selectable
2. The LCD driver voltage generator can switch internal voltage boosting method, capacitor split method, and external resistance division method.
3. Automatic output of segment and common signals based on automatic display data register read
4. The reference voltage to be generated when operating the voltage boost circuit is selectable from among 23 levels (contrast adjustment).
5. Alternation of the image on the LCD is available. The time for switching in alternation of the display is selectable.

Table 22 - 2 lists the maximum number of pixels that can be displayed in each display mode.

Table 22 - 2 Maximum Number of Pixels (1/12)

a) 100-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	56 (56 segment signals, 1 common signal)
			1/2	2	112 (56 segment signals, 2 common signals)
				3	168 (56 segment signals, 3 common signals)
			1/3	3	
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
		VL2 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
		VL4 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (2/12)

a) 100-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
		VDD reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)
			1/4	8	
		VL4 reference	1/3	3	168 (56 segment signals, 3 common signals)
				4	224 (56 segment signals, 4 common signals)
				6	324 (54 segment signals, 6 common signals)
				8	416 (52 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (3/12)

b) 80-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	49 (49 segment signals, 1 common signal)
			1/2	2	98 (49 segment signals, 2 common signals)
				3	147 (49 segment signals, 3 common signals)
			1/3	3	
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
		VL2 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
		VL4 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (4/12)

b) 80-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
		VDD reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)
			1/4	8	
		VL4 reference	1/3	3	147 (49 segment signals, 3 common signals)
				4	196 (49 segment signals, 4 common signals)
				6	282 (47 segment signals, 6 common signals)
				8	360 (45 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (5/12)

c) 64-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	34 (34 segment signals, 1 common signal)
			1/2	2	68 (34 segment signals, 2 common signals)
				3	102 (34 segment signals, 3 common signals)
			1/3	3	
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
		VL2 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
		VL4 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (6/12)

c) 64-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
		VDD reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)
			1/4	8	
		VL4 reference	1/3	3	102 (34 segment signals, 3 common signals)
				4	136 (34 segment signals, 4 common signals)
				6	192 (32 segment signals, 6 common signals)
				8	240 (30 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (7/12)

d) 52-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	25 (25 segment signals, 1 common signal)
			1/2	2	50 (25 segment signals, 2 common signals)
				3	75 (25 segment signals, 3 common signals)
			1/3	3	
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
		VL2 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
		VL4 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (8/12)

d) 52-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
		VDD reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)
			1/4	8	
		VL4 reference	1/3	3	75 (25 segment signals, 3 common signals)
				4	100 (25 segment signals, 4 common signals)
				6	138 (23 segment signals, 6 common signals)
				8	168 (21 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (9/12)

e) 48-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	22 (22 segment signals, 1 common signal)
			1/2	2	44 (22 segment signals, 2 common signals)
				3	66 (22 segment signals, 3 common signals)
			1/3	3	
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
		VL2 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
		VL4 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (10/12)

e) 48-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
		VDD reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)
			1/4	8	
		VL4 reference	1/3	3	66 (22 segment signals, 3 common signals)
				4	88 (22 segment signals, 4 common signals)
				6	120 (20 segment signals, 6 common signals)
				8	144 (18 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (11/12)

f) 44-pin products (1/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division		—	Static	19 (19 segment signals, 1 common signal)
			1/2	2	38 (19 segment signals, 2 common signals)
				3	57 (19 segment signals, 3 common signals)
			1/3	3	
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
		VL2 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
	Capacitor split	VDD reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
		VL4 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)

Table 22 - 2 Maximum Number of Pixels (12/12)

f) 44-pin products (2/2)

Drive Waveform for LCD Driver	LCD Driver Voltage Generator		Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform B	External resistance division		1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	8	
	Internal voltage boosting	VL1 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	8	
		VL2 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
		VDD reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)
			1/4	8	
	Capacitor split	VL4 reference	1/3	3	57 (19 segment signals, 3 common signals)
				4	76 (19 segment signals, 4 common signals)
				6	102 (17 segment signals, 6 common signals)
				8	120 (15 segment signals, 8 common signals)

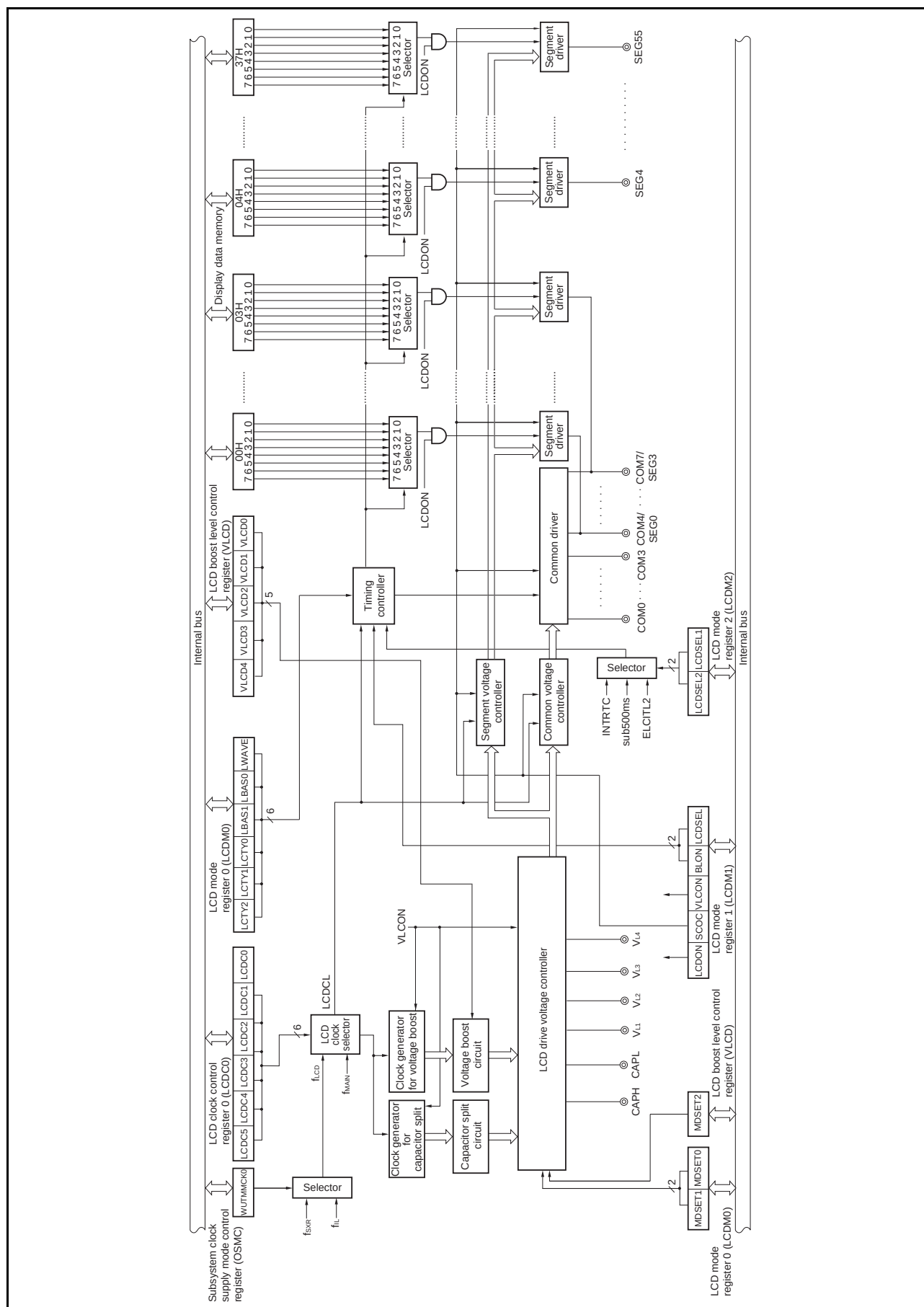
22.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 22 - 3 Configuration of LCD Controller/Driver

Item	Configuration
Control registers	• LCD mode register 0 (LCDM0) • LCD mode register 1 (LCDM1) • LCD mode register 2 (LCDM2) • Subsystem clock supply mode control register (OSMC) • LCD clock control register 0 (LCDC0) • LCD boost level control register (VLCD) • LCD input switch control register (ISCLCD) • Port mode registers (PMxx) • Port registers (Pxx) • Pull-up resistor option registers (PUxx) • Port input mode registers (PIMxx) • Port output mode registers (POMxx) • Port mode control A registers (PMCAxx) • Port mode control T registers (PMCTxx) • Port mode control E registers (PMCExx) • LCD port function registers (PFSEGx)

Figure 22 - 1 Block Diagram of LCD Controller/Driver



22.3 Registers for Controlling the LCD Controller/Driver

The following registers are used to control the LCD controller/driver.

- LCD mode register 0 (LCDM0)
- LCD mode register 1 (LCDM1)
- LCD mode register 2 (LCDM2)
- Subsystem clock supply mode control register (OSMC)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- LCD input switch control register (ISCLCD)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

22.3.1 LCD mode register 0 (LCDM0)

The LCDM0 register specifies the LCD operation. The MDSET1 and MDSET0 bits of this register is used in combination with the MDSET2 bit of the VLCD register. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 2 Format of LCD Mode Register 0 (LCDM0)

Address: FFF40H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

MDSET2 Note	MDSET1	MDSET0	LCD drive voltage generator selection
0	0	0	External resistance division method
0	0	1	Internal voltage boosting method on the VL1 reference
1	0	1	Internal voltage boosting method on the VL2 reference
0	1	0	Capacitor split method for the VDD reference
1	1	0	Capacitor split method for the VL4 reference
Other than above			Setting prohibited

LWAVE	LCD display waveform selection
0	Waveform A
1	Waveform B

LDTY2	LDTY1	LDTY0	Selection of time slice of LCD display
0	0	0	Static
0	0	1	2-time slice
0	1	0	3-time slice
0	1	1	4-time slice
1	0	0	6-time slice
1	0	1	8-time slice
Other than above			Setting prohibited

LBAS1	LBAS0	LCD display bias mode selection
0	0	1/2 bias method
0	1	1/3 bias method
1	0	1/4 bias method
1	1	Setting prohibited

Note MDSET2 is assigned to bit 7 of the VLCD register.

Caution 1. Do not rewrite the LCDM0 value while the SCOC bit of the LCDM1 register = 1.

Caution 2. When "Static" is selected (LDTY2 to LDTY0 bits = 000B), be sure to set the LBAS1 and LBAS0 bits to the default value (00B). Otherwise, the operation will not be guaranteed.

Caution 3. Only the combinations of display waveform, number of time slices, and bias method shown in Table 22 - 4 are supported. Combinations of settings not shown in Table 22 - 4 are prohibited.

Table 22 - 4 Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency

Display Mode			Set Value						Driving Voltage Generation Method				
Display Waveform	Number of Time Slices	Bias Mode	LWA VE	LDTY2[2:0]			LBAS[1:0]		External Resistance Division	Internal voltage boosting		Capacitor split	
										VL1	VL2	VDD	VL4
Waveform A	8	1/4	0	1	0	1	1	0	✓ (24 to 128 Hz)	✓ (24 to 64 Hz)	×	✓ (24 to 128 Hz)	×
Waveform A	6	1/4	0	1	0	0	1	0	×	✓ (32 to 86 Hz)	×	✓ (32 to 86 Hz)	×
Waveform A	8	1/3	0	1	0	1	0	1	✓ (32 to 128 Hz)	✓ (32 to 64 Hz)	✓ (32 to 64 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)
Waveform A	6	1/3	0	1	0	0	0	1	✓ (32 to 128 Hz)	✓ (32 to 86 Hz)	✓ (32 to 80 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)
Waveform A	4	1/3	0	0	1	1	0	1	✓ (24 to 128 Hz)	✓ (24 to 128 Hz)	✓ (24 to 80 Hz)	✓ (24 to 128 Hz)	✓ (24 to 128 Hz)
Waveform A	3	1/3	0	0	1	0	0	1	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)	✓ (32 to 80 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)
Waveform A	3	1/2	0	0	1	0	0	0	✓ (32 to 128 Hz)	×	×	×	×
Waveform A	2	1/2	0	0	0	1	0	0	✓ (24 to 128 Hz)	×	×	×	×
Waveform A	Static		0	0	0	0	0	0	✓ (24 to 128 Hz)	×	×	×	×
Waveform B	8	1/4	1	1	0	1	1	0	✓ (24 to 128 Hz)	✓ (24 to 64 Hz)	×	✓ (24 to 128 Hz)	×
Waveform B	8	1/3	1	1	0	1	0	1	✓ (32 to 128 Hz)	✓ (32 to 64 Hz)	✓ (32 to 64 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)
Waveform B	6	1/3	1	1	0	0	0	1	✓ (32 to 128 Hz)	✓ (32 to 86 Hz)	✓ (32 to 80 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)
Waveform B	4	1/3	1	0	1	1	0	1	✓ (24 to 128 Hz)	✓ (24 to 128 Hz)	✓ (24 to 80 Hz)	✓ (24 to 128 Hz)	✓ (24 to 128 Hz)
Waveform B	3	1/3	1	0	1	0	0	1	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)	✓ (32 to 80 Hz)	✓ (32 to 128 Hz)	✓ (32 to 128 Hz)

Remark ✓: Supported
 ×: Not supported

22.3.2 LCD mode register 1 (LCDM1)

The LCDM1 register enables or disables the operation of the display and of the voltage boost circuit or capacitor split circuit, and specifies the data area for display and the low-voltage mode. The LCDM1 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 22 - 3 Format of LCD Mode Register 1 (LCDM1) (1/2)

Address: FFF41H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM
SCOC								
LCDON								
Enabling or disabling the LCD (when waveform A or B is to be output)								
0								
Output ground level to the segment or common pin								
0								
1								
Display off (all segment outputs are deselected)								
1								
Display on								
VLCON								
Enabling or disabling the operation of the voltage boost circuit or capacitor split circuit								
0								
Stops the operation of the voltage boost circuit or capacitor split circuit								
1								
Enables the operation of the voltage boost circuit or capacitor split circuit								
Note 1								
BLON								
Note 2								
LCDSEL								
Display data area control								
0								
Displaying an A-pattern area data (lower four bits of LCD display data register)								
0								
1								
Displaying a B-pattern area data (higher four bits of LCD display data register)								
1								
Alternately displaying the data in the A-pattern area and the data in the B-pattern area (alternation of the display according to the timing set by the LCDSEL2 and LCDSEL1 bits of the LCDM2 register)								
1								
LCDVLM								
Note 3								
Control of default value of voltage boosting pin								
0								
Set when $V_{DD} \geq 2.7\text{ V}$								
1								
Set when $V_{DD} \leq 4.2\text{ V}$								

Note 1. Setting the VLCON bit to 1 is prohibited in the external resistance division mode.

Note 2. When f_{IL} is selected as the LCD source clock (f_{LCD}), be sure to set the BLON bit to 0.

Note 3. This function is used to shorten the boost stabilization time by setting the VL_x pin to the default state when the voltage boost circuit is used.

If the V_{DD} voltage is 2.7 V or higher when boosting is started, set the LCDVLM bit to 0; if the V_{DD} voltage is 4.2 V or less, set the LCDVLM bit to 1. However, when $2.7\text{ V} \leq V_{DD} \leq 4.2\text{ V}$, operation is possible with LCDVLM = 0 or LCDVLM = 1.

(Cautions are listed on the next page.)

- Caution 1. When the voltage boost circuit is used, set the SCOC bit = 0 and the VLCON bit = 0, and the MDSET1 and MDSET0 bits = 00B in order to reduce power consumption when the LCD is not used. When the MDSET1 and MDSET0 bits = 01B, power is consumed by the internal reference voltage generator.
- Caution 2. When the external resistance division method has been set (MDSET1 and MDSET0 of LCDM0 = 00B) or capacitor split method has been set (MDSET1 and MDSET0 = 10B), set the LCDVLM bit to 0.
- Caution 3. Do not rewrite the VLCON and LCDVLM bits while the SCOC bit = 1.
- Caution 4. Set the BLON and LCDSEL bits to 0 when 6 or 8 has been selected as the number of time slices for the display mode.
- Caution 5. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register, wait for the reference voltage setup time (at least 10 ms), and then set the VLCON bit to 1. If the default reference voltage is to be used, the internal voltage boosting method must be selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B).
- Caution 6. To use the capacitor split method when the VL4 reference is selected, after selecting the capacitor split method on the VL4 reference (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 10B and setting the MDSET2 bit of the VLCD register to 1), wait for the reference voltage setup time (at least 10 ms), and then set the VLCON bit to 1.

22.3.3 LCD mode register 2 (LCDM2)

The LCDM2 register is used to select the trigger for switching in alternation of the display on the LCD. The LCDM2 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 4 Format of LCD Mode Register 2 (LCDM2)

Address: F0299H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LCDM2	0	0	0	0	0	0	LCDSEL2	LCDSEL1

LCDSEL2	LCDSEL1	Selection of the time for switching in alternation of the display on the LCD
0	0	Alternation in response to INTRTC
1	0	Alternation at the interval of 0.5 seconds (in response to the sub500ms signal)
1	1	Alternation in response to ELCITL2
Other than above		Setting prohibited

22.3.4 Subsystem clock supply mode control register (OSMC)

The OSMC register is used to reduce power consumption by stopping unnecessary clock functions. The operating clock (f_{LCD}) of the LCD controller/driver can be selected by using the WUTMMCK0 bit. 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 22 - 5 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	WUTMMCK 0	x	x	0	HIPREC

WUTMMCK 0	Selection of the operating clock (f _{LCD}) of the LCD controller/driver
0	Subsystem clock XR (fs _{XR})
1	Low-speed on-chip oscillator clock (f _L)

Caution 1. It is prohibited to set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.

Caution 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, serial interface U_{ART}A, clock output/buzzer output controller, LCD controller/driver, 32-bit interval timer, 8-bit interval timer, and external signal sampler are all stopped.

22.3.5 LCD clock control register 0 (LCDC0)

The LCDC0 register specifies the LCD clock. The frame frequency is determined according to the LCD clock and the number of time slices. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 6 Format of LCD Clock Control Register 0 (LCDC0)

Address: FFF42H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LCDC0	0	0	LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD clock (LCDCL)	
						WUTMMCK0 = 0	WUTMMCK0 = 1
0	0	0	0	0	1	$f_{\text{SXR}}/2^2$	$f_{\text{IL}}/2^2$
0	0	0	0	1	0	$f_{\text{SXR}}/2^3$	$f_{\text{IL}}/2^3$
0	0	0	0	1	1	$f_{\text{SXR}}/2^4$	$f_{\text{IL}}/2^4$
0	0	0	1	0	0	$f_{\text{SXR}}/2^5$	$f_{\text{IL}}/2^5$
0	0	0	1	0	1	$f_{\text{SXR}}/2^6$	$f_{\text{IL}}/2^6$
0	0	0	1	1	0	$f_{\text{SXR}}/2^7$	$f_{\text{IL}}/2^7$
0	0	0	1	1	1	$f_{\text{SXR}}/2^8$	$f_{\text{IL}}/2^8$
0	0	1	0	0	0	$f_{\text{SXR}}/2^9$	$f_{\text{IL}}/2^9$
0	0	1	0	0	1	$f_{\text{SXR}}/2^{10}$	$f_{\text{IL}}/2^{10}$
0	1	0	0	0	1	$f_{\text{MAIN}}/2^8$	
0	1	0	0	1	0	$f_{\text{MAIN}}/2^9$	
0	1	0	0	1	1	$f_{\text{MAIN}}/2^{10}$	
0	1	0	1	0	0	$f_{\text{MAIN}}/2^{11}$	
0	1	0	1	0	1	$f_{\text{MAIN}}/2^{12}$	
0	1	0	1	1	0	$f_{\text{MAIN}}/2^{13}$	
0	1	0	1	1	1	$f_{\text{MAIN}}/2^{14}$	
0	1	1	0	0	0	$f_{\text{MAIN}}/2^{15}$	
0	1	1	0	0	1	$f_{\text{MAIN}}/2^{16}$	
0	1	1	0	1	0	$f_{\text{MAIN}}/2^{17}$	
0	1	1	0	1	1	$f_{\text{MAIN}}/2^{18}$	
1	0	1	0	1	1	$f_{\text{MAIN}}/2^{19}$	
Other than above						Setting prohibited	

Caution 1. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.

Caution 2. Be sure to set bits 7 and 6 to 0.

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

Caution 3. When the internal voltage boosting method or capacitor split method is set, set the LCD clock (LCDCL) as follows:

- 512 Hz or less when fsXR is selected.
- 512 Hz or less when fIL is selected.

For details, see Table 22 - 4 Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency.

Remark fMAIN: Main system clock frequency
fIL: Low-speed on-chip oscillator clock frequency
fsXR: Subsystem clock XR frequency

22.3.6 LCD boost level control register (VLCD)

The VLCD register selects the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage is selectable from among 23 levels. The MDSET2 bit of this register is used in combination with the MDSET1 and MDSET0 bits of the LCDM0 register. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 04H.

Figure 22 - 7 Format of LCD Boost Level Control Register (VLCD) (1/3)

Address: FFF43H

After reset: 04H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
VLCD	MDSET2	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0

MDSET2	MDSET1 Note	MDSET0 Note	LCD drive voltage generator selection
0	0	0	External resistance division method
0	0	1	Internal voltage boosting method on the VL1 reference
1	0	1	Internal voltage boosting method on the VL2 reference
0	1	0	Capacitor split method for the VDD reference
1	1	0	Capacitor split method for the VL4 reference
Other than above			Setting prohibited

Note MDSET1 and MDSET0 are assigned to bits 7 and 6 of the LCDM0 register.

Caution Be sure to set bits 6 and 5 to 0.

Figure 22 - 7 Format of LCD Boost Level Control Register (VLCD) (2/3)

<When the internal voltage boosting method on the VL1 reference is selected
(MDSET2 to MDSET0 = 3'b001)>

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	VL1 reference voltage selected (contrast adjustment)	VL4 voltage	
						1/3 bias method	1/4 bias method
0	0	1	0	0	1.01 V (default)	3.03 V	4.04 V
0	0	1	0	1	1.04 V	3.12 V	4.16 V
0	0	1	1	0	1.07 V	3.21 V	4.28 V
0	0	1	1	1	1.11 V	3.33 V	4.44 V
0	1	0	0	0	1.14 V	3.42 V	4.56 V
0	1	0	0	1	1.17 V	3.51 V	4.68 V
0	1	0	1	0	1.21 V	3.63 V	4.84 V
0	1	0	1	1	1.24 V	3.72 V	4.96 V
0	1	1	0	0	1.27 V	3.81 V	5.01 V
0	1	1	0	1	1.31 V	3.93 V	5.24 V
0	1	1	1	0	1.34 V	4.02 V	Setting prohibited
0	1	1	1	1	1.37 V	4.11 V	Setting prohibited
1	0	0	0	0	1.40 V	4.20 V	Setting prohibited
1	0	0	0	1	1.44 V	4.32 V	Setting prohibited
1	0	0	1	0	1.47 V	4.41 V	Setting prohibited
1	0	0	1	1	1.50 V	4.50 V	Setting prohibited
1	0	1	0	0	1.54 V	4.62 V	Setting prohibited
1	0	1	0	1	1.57 V	4.71 V	Setting prohibited
1	0	1	1	0	1.60 V	4.80 V	Setting prohibited
1	0	1	1	1	1.64 V	4.92 V	Setting prohibited
1	1	0	0	0	1.67 V	5.01 V	Setting prohibited
1	1	0	0	1	1.70 V	5.10 V	Setting prohibited
1	1	0	1	0	1.74 V	5.22 V	Setting prohibited
Other than above					Setting prohibited		

Figure 22 - 7 Format of LCD Boost Level Control Register (VLCD) (3/3)

<When the internal voltage boosting method on the VL2 reference is selected
(MDSET2 to MDSET0 = 3'b101)>

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	VL2 reference voltage selected (contrast adjustment)	VL4 voltage
						1/3 bias method
0	0	1	0	0	2.02 V (default)	3.03 V
0	0	1	0	1	2.09 V	3.14 V
0	0	1	1	0	2.16 V	3.24 V
0	0	1	1	1	2.22 V	3.33 V
0	1	0	0	0	2.29 V	3.44 V
0	1	0	0	1	2.36 V	3.54 V
0	1	0	1	0	2.42 V	3.63 V
0	1	0	1	1	2.49 V	3.74 V
0	1	1	0	0	2.56 V	3.84 V
0	1	1	0	1	2.62 V	3.93 V
0	1	1	1	0	2.69 V	4.07 V
0	1	1	1	1	2.76 V	4.14 V
1	0	0	0	0	2.82 V	4.23 V
1	0	0	0	1	2.89 V	4.34 V
1	0	0	1	0	2.96 V	4.44 V
1	0	0	1	1	3.02 V	4.53 V
1	0	1	0	0	3.09 V	4.64 V
1	0	1	0	1	3.15 V	4.73 V
1	0	1	1	0	3.22 V	4.83 V
1	0	1	1	1	3.29 V	4.94 V
1	1	0	0	0	3.35 V	5.03 V
1	1	0	0	1	3.42 V	5.13 V
1	1	0	1	0	3.49 V	5.24 V
Other than above					Setting prohibited	

Caution 1. The VLCD setting is valid only when the voltage boost circuit is operating.

Caution 2. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).

Caution 3. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register, wait for the reference voltage setup time (at least 10 ms), and then set the VLCON bit to 1. If the default reference voltage is to be used, the internal voltage boosting method must be selected (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B).

Caution 4. If the external resistance division method or capacitor split method is to be used, do not change the setting of the VLCD4 to VLCD0 bits of the VLCD register from the initial value (04H).

22.3.7 LCD input switch control register (ISCLCD)

Input to the Schmitt trigger buffer must be disabled until the VL4/P85, VL3/P125, VL2/P86, VL1/P87, CAPL/P126, and CAPH/P127 pins are set to operate as LCD function pins in order to prevent through-current from entering. 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.

<R> Figure 22 - 8 Format of LCD Input Switch Control Register (ISCLCD)

Address: F0298H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
ISCLCD	0	0	0	ISCVL4	ISCVL3	ISCVL2	ISCVL1	ISCCAP
ISCVL4 Control of the Schmitt trigger buffer of the VL4/P85 pin								
0 The LCD function enabled (i.e. the pin is used as the VL4 pin).								
1 The LCD function disabled (i.e. digital input enabled).								
ISCVL3 Control of the Schmitt trigger buffer of the VL3/P125 pin								
0 The LCD function enabled (i.e. the pin is used as the VL3 pin).								
1 The LCD function disabled (i.e. digital input enabled).								
ISCVL2 Control of the Schmitt trigger buffer of the VL2/P86 pin								
0 The LCD function enabled (i.e. the pin is used as the VL2 pin).								
1 The LCD function disabled (i.e. digital input enabled).								
ISCVL1 Control of the Schmitt trigger buffer of the VL1/P87 pin								
0 The LCD function enabled (i.e. the pin is used as the VL1 pin).								
1 The LCD function disabled (i.e. digital input enabled).								
ISCCAP Control of the Schmitt trigger buffer of the CAPL/P126 and CAPH/P127 pins								
0 The LCD function enabled (i.e. the pins are used as the CAPL and CAPH pins).								
1 The LCD function disabled (i.e. digital input enabled).								

Caution 1. If the ISCVL4 bit = 0, set the corresponding port control register as follows.

PU85 bit of PU8 register = 0, P85 bit of P8 register = 0

Caution 2. If the ISCVL3 bit = 0, set the corresponding port control register as follows.

PU125 bit of PU12 register = 0, P125 bit of P12 register = 0

Caution 3. If the ISCVL2 bit = 0, set the corresponding port control register as follows.

PU86 bit of PU8 register = 0, P86 bit of P8 register = 0

Caution 4. If the ISCVL1 bit = 0, set the corresponding port control register as follows.

PU87 bit of PU8 register = 0, P87 bit of P8 register = 0

Caution 5. If the ISCCAP bit = 0, set the corresponding port control register as follows.

PU126 bit of PU12 register = 0, P126 bit of P12 register = 0

PU127 bit of PU12 register = 0, P127 bit of P12 register = 0

a) Operation of ports that alternately function as VL4 to VL1, CAPL, and CAPH pins

The functions of the VL4/P85, VL3/P125, VL2/P86, VL1/P87, CAPL/P126, and CAPH/P127 pins are determined by the settings of the following registers: LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), port mode register 8 (PM8), and port mode register 12 (PM12).

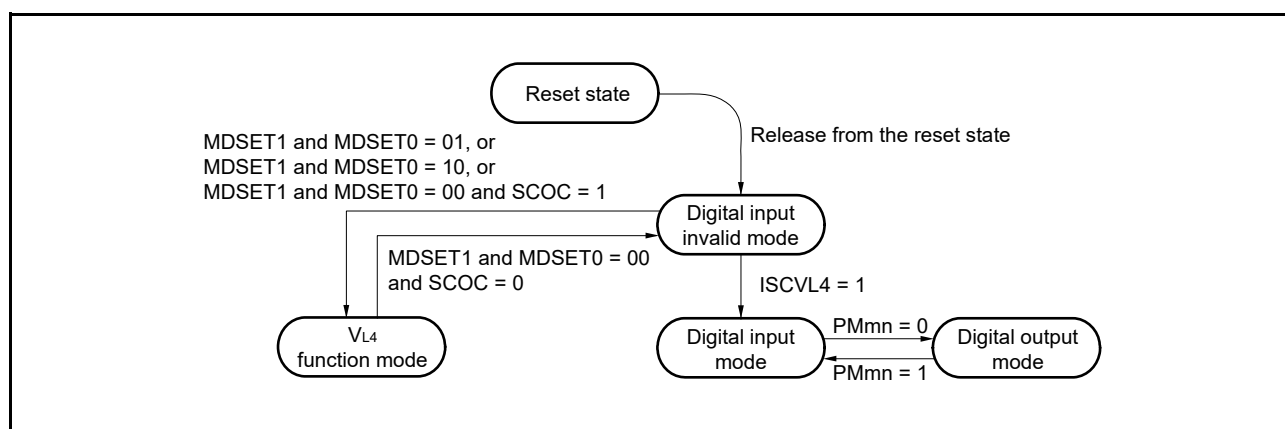
• VL4/P85

Table 22 - 5 Settings of VL4/P85 Pin Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	SCOC Bit of LCDM1 Register	ISCVL4 Bit of ISCLCD Register	PM85 Bit of PM8 Register	Pin Function	Initial State
External resistance division (MDSET1 and MDSET0 = 00)	0	0	1	Digital input invalid mode	✓
	0	1	0	Digital output mode	—
	0	1	1	Digital input mode	—
	1	0	1	VL4 function mode	—
Internal voltage boosting or capacitor split (MDSET1 and MDSET0 = 01 or 10)	0/1	0	1	VL4 function mode	—
Other than above				Setting prohibited	

The following shows the VL4/P85 pin function state transitions.

Figure 22 - 9 VL4/P85 Pin Function State Transitions



• TS02/VL3/P125

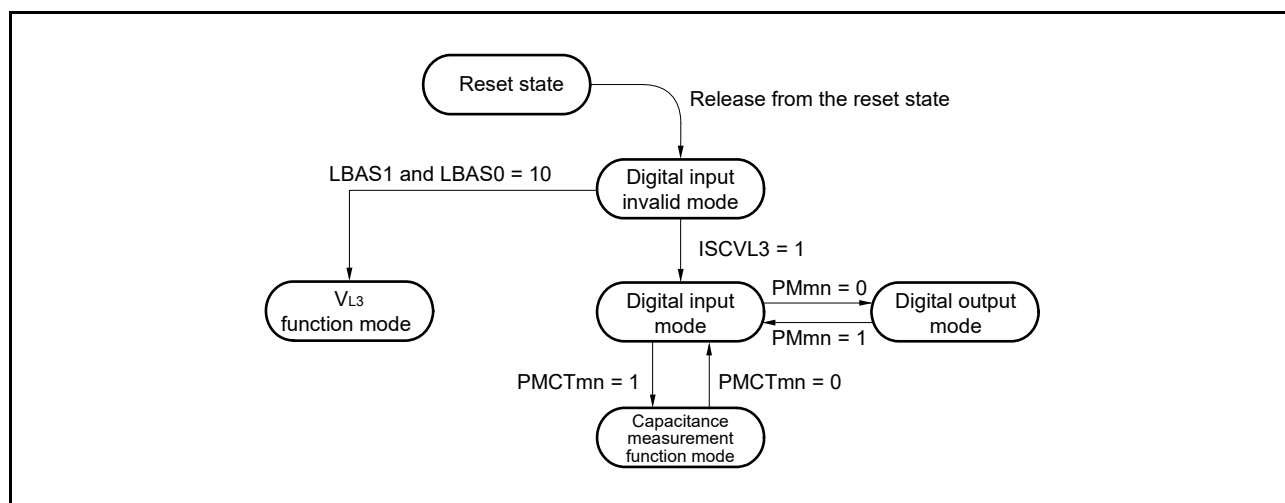
Table 22 - 6 Settings of TS02/VL3/P125 Pin Function

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register)	PMCT125 Bit of PMCT12 Register	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial State
Other than 1/4 bias method (LBAS1 and LBAS0 = 00 or 01)	0	0	1	Digital input invalid mode	✓
	0	1	0	Digital output mode	—
	0	1	1	Digital input mode	—
	1	1	×	Capacitance measurement function mode	—
1/4 bias method (LBAS1 and LBAS0 = 10)	0	0	1	VL3 function mode	—
Other than above				Setting prohibited	

Remark ×: Don't care

The following shows the TS02/VL3/P125 pin function state transitions.

Figure 22 - 10 TS02/VL3/P125 Pin Function State Transitions



Caution Be sure to set the VL3 function mode before segment output starts (while the SCOC bit of LCD mode register 1 (LCDM1) is 0).

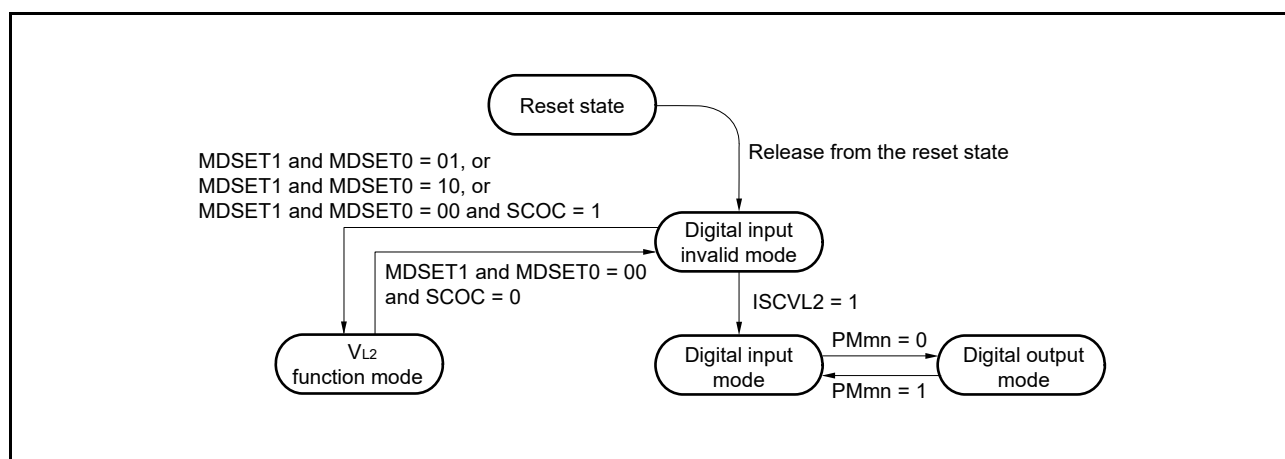
- VL2/P86

Table 22 - 7 Settings of VL2/P86 Pin Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	SCOC Bit of LCDM1 Register	ISCVL2 Bit of ISCLCD Register	PM86 Bit of PM8 Register	Pin Function	Initial State
External resistance division (MDSET1 and MDSET0 = 00)	0	0	1	Digital input invalid mode	✓
	0/1	1	0	Digital output mode	—
	0/1	1	1	Digital input mode	—
	1	0	1	VL2 function mode	—
Internal voltage boosting or capacitor split (MDSET1 and MDSET0 = 01 or 10)	0/1	0	1	VL2 function mode	—
Other than above				Setting prohibited	

The following shows the VL2/P86 pin function state transitions.

Figure 22 - 11 VL2/P86 Pin Function State Transitions



Caution To set the digital input mode, do so before changing the settings of the MDSET1, MDSET0, and SCOC bits.

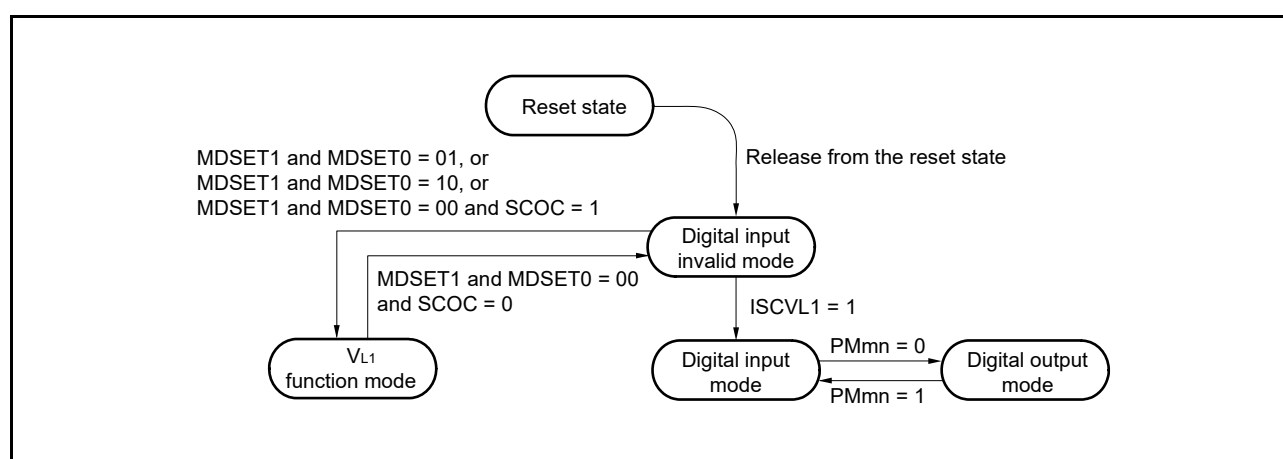
- VL1/P87

Table 22 - 8 Settings of VL1/P87 Pin Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	SCOC Bit of LCDM1 Register	ISCVL1 Bit of ISCLCD Register	PM87 Bit of PM8 Register	Pin Function	Initial State
External resistance division (MDSET1 and MDSET0 = 00)	0	0	1	Digital input invalid mode	✓
	0/1	1	0	Digital output mode	—
	0/1	1	1	Digital input mode	—
	1	0	1	VL1 function mode	—
Internal voltage boosting or capacitor split (MDSET1 and MDSET0 = 01 or 10)	0/1	0	1	VL1 function mode	—
Other than above				Setting prohibited	

The following shows the VL1/P87 pin function state transitions.

Figure 22 - 12 VL1/P87 Pin Function State Transitions



Caution To set the digital input mode, do so before changing the settings of the MDSET1, MDSET0, and SCOC bits.

- TS01/CAPL/P126, TS00/CAPH/P127

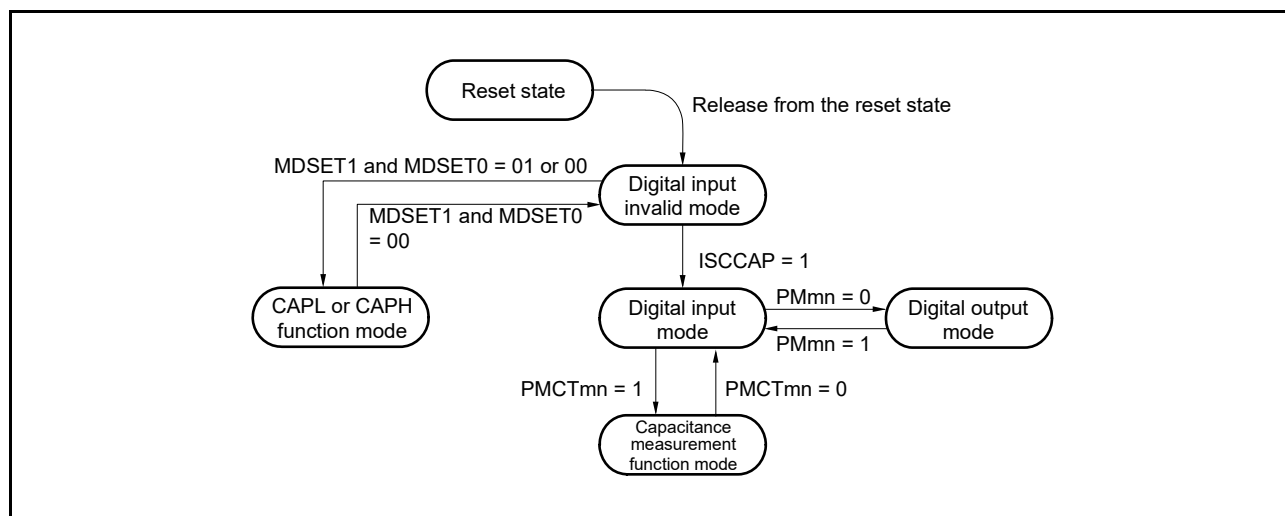
Table 22 - 9 Settings of TS01/CAPL/P126 and TS00/CAPH/P127 Pin Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	PMCT126 and PMCT127 Bits of PMCT12 Register	ISCCAP Bit of ISCLCD Register	PM126 and PM127 Bits of PM12 Register	Pin Function	Initial State
External resistance division (MDSET1 and MDSET0 = 00)	0	0	1	Digital input invalid mode	✓
	0	1	0	Digital output mode	—
	0	1	1	Digital input mode	—
	1	1	×	Capacitance measurement function mode	—
Internal voltage boosting or capacitor split (MDSET1 and MDSET0 = 01 or 10)	0	0	1	CAPL or CAPH function mode	—
Other than above				Setting prohibited	

Remark ×: Don't care

The following shows the state transitions of the TS01/CAPL/P126 and TS00/CAPH/P127 pin functions.

Figure 22 - 13 State Transitions of the TS01/CAPL/P126 and TS00/CAPH/P127 Pin Functions



Caution Be sure to set the CAPL or CAPH function mode before segment output starts (while the SCOC bit of LCD mode register 1 (LCDM1) is 0).

22.3.8 LCD port function registers (PFSEGx)

The PFSEGx registers can be used to set whether the P90 to P97 pins are to be used as port pins (other than common output pins) or as common output pins. These registers can also be used to set whether the P00 to P07, P10 to P17, P22 to P27, P30 to P35, P46, P47, P50 to P57, P70 to P77, P94 to P97, P130, and P141 to P145 pins are to be used as port pins (other than segment output pins) or as segment output pins. The PFSEGx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of PFSEG1 to PFSEG7 following a reset is FFH, and that of PFSEG0 is 0FH.

Remark The correspondence between the COM and SEG output pins (COMxx and SEGxx) and the PFSEGx registers (PFCOMy and PFSEGz bits), and presence or absence of the COMxx and SEGxx pins in each product are shown in **Table 22 - 10 Common and Segment Output Pins in Each Product and the Corresponding PFSEGx Registers (PFCOMy and PFSEGz Bits)**.

Figure 22 - 14 Format of LCD Port Function Registers (PFSEGx) (1/2)

Address: F0290H

After reset: 0FH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	0	0	0	0	PFCOM3	PFCOM2	PFCOM1	PFCOM0

Address: F0291H

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG07	PFSEG06	PFSEG05	PFSEG04	PFSEG03	PFSEG02	PFSEG01	PFSEG00

Address: F0292H

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0293H

After reset: FFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0294H

After reset: EFH

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG31	PFSEG30	PFSEG29	PFSEG28	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Figure 22 - 14 Format of LCD Port Function Registers (PFSEGx) (2/2)

Address: F0295H
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG39	PFSEG38	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0296H
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6	PFSEG47	PFSEG46	PFSEG45	PFSEG44	PFSEG43	PFSEG42	PFSEG41	PFSEG40

Address: F0297H
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG7	PFSEG55	PFSEG54	PFSEG53	PFSEG52	PFSEG51	PFSEG50	PFSEG49	PFSEG48

PFCOMy (y = 0 to 3)	Specifies whether to use the Pmn pins as port pins (other than common output pins) or as common output pins. (mn = 90 to 93)
0	Uses the pins as port pins (other than common output pins).
1	Uses the pins as common output pins.

PFSEGz (z = 00 to 55)	Specifies whether to use the Pmn pins as port pins (other than segment output pins) or as segment output pins. (mn = 00 to 07, 10 to 17, 22 to 27, 30 to 35, 46, 47, 50 to 57, 70 to 77, 94 to 97, 130, and 141 to 145)
0	Uses the pins as port pins (other than segment output pins).
1	Uses the pins as segment output pins.

Remark If the pins are used as common or segment output pins (PFCOMy or PFSEGz = 1), be sure to set the following bits to 0:

- PUMn bit of the PUM register
- POMmn bit of the POM register
- PIMmn bit of the PIM register

Table 22 - 10 Common and Segment Output Pins in Each Product and the Corresponding PFSEGx Registers
(PFCOMy and PFSEGz Bits) (1/2)

Bit Name of PFSEG Register	Corresponding Pins	Alternate Port	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
PFCOM0	COM0	P90	✓	✓	✓	✓	✓	✓
PFCOM1	COM1	P91	✓	✓	✓	✓	✓	✓
PFCOM2	COM2	P92	✓	✓	✓	✓	✓	✓
PFCOM3	COM3	P93	✓	✓	✓	✓	✓	✓
PFSEG00	COM4/SEG0	P94	✓	✓	✓	✓	✓	✓
PFSEG01	COM5/SEG1	P95	✓	✓	✓	✓	✓	✓
PFSEG02	COM6/SEG2	P96	✓	✓	✓	✓	✓	✓
PFSEG03	COM7/SEG3	P97	✓	✓	✓	✓	✓	✓
PFSEG04	SEG4	P50	✓	✓	—	—	—	—
PFSEG05	SEG5	P51	✓	✓	—	—	—	—
PFSEG06	SEG6	P52	✓	✓	✓	—	—	—
PFSEG07	SEG7	P53	✓	✓	✓	✓	—	—
PFSEG08	SEG8	P54	✓	✓	✓	✓	✓	—
PFSEG09	SEG9	P55	✓	✓	—	—	—	—
PFSEG10	SEG10	P56	✓	✓	—	—	—	—
PFSEG11	SEG11	P57	✓	✓	✓	✓	✓	✓
PFSEG12	SEG12	P70	✓	✓	✓	—	—	—
PFSEG13	SEG13	P71	✓	✓	—	—	—	—
PFSEG14	SEG14	P72	✓	✓	—	—	—	—
PFSEG15	SEG15	P73	✓	✓	—	—	—	—
PFSEG16	SEG16	P74	✓	✓	✓	✓	—	—
PFSEG17	SEG17	P75	✓	✓	✓	✓	✓	—
PFSEG18	SEG18	P76	✓	✓	✓	✓	✓	✓
PFSEG19	SEG19	P77	✓	✓	✓	✓	✓	✓
PFSEG20	SEG20	P30	✓	✓	✓	—	—	—
PFSEG21	SEG21	P31	✓	✓	✓	—	—	—
PFSEG22	SEG22	P32	✓	—	—	—	—	—
PFSEG23	SEG23	P33	✓	—	—	—	—	—
PFSEG24	SEG24	P34	✓	✓	—	—	—	—
PFSEG25	SEG25	P35	✓	✓	—	—	—	—
PFSEG26	SEG26	P46	✓	✓	—	—	—	—
PFSEG27	SEG27	P47	✓	✓	—	—	—	—
PFSEG28	SEG28	P130	✓	✓	—	—	—	—
PFSEG29	SEG29	P22	✓	✓	✓	✓	✓	✓
PFSEG30	SEG30	P23	✓	✓	—	—	—	—
PFSEG31	SEG31	P24	✓	✓	—	—	—	—
PFSEG32	SEG32	P25	✓	✓	—	—	—	—
PFSEG33	SEG33	P26	✓	✓	✓	✓	✓	✓

Table 22 - 10 Common and Segment Output Pins in Each Product and the Corresponding PFSEGx Registers
(PFCOMy and PFSEGz Bits) (2/2)

Bit Name of PFSEG Register	Corresponding Pins	Alternate Port	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
PFSEG34	SEG34	P27	✓	✓	✓	✓	✓	✓
PFSEG35	SEG35	P10	✓	✓	✓	✓	✓	✓
PFSEG36	SEG36	P11	✓	✓	✓	✓	✓	✓
PFSEG37	SEG37	P12	✓	✓	✓	✓	✓	✓
PFSEG38	SEG38	P13	✓	✓	✓	✓	✓	—
PFSEG39	SEG39	P14	✓	✓	✓	✓	—	—
PFSEG40	SEG40	P15	✓	✓	✓	—	—	—
PFSEG41	SEG41	P16	✓	✓	✓	✓	✓	✓
PFSEG42	SEG42	P17	✓	✓	✓	✓	✓	✓
PFSEG43	SEG43	P00	✓	✓	✓	✓	✓	✓
PFSEG44	SEG44	P01	✓	✓	✓	—	—	—
PFSEG45	SEG45	P02	✓	✓	✓	—	—	—
PFSEG46	SEG46	P03	✓	✓	✓	—	—	—
PFSEG47	SEG47	P04	✓	✓	✓	—	—	—
PFSEG48	SEG48	P05	✓	✓	✓	✓	✓	✓
PFSEG49	SEG49	P06	✓	✓	✓	✓	✓	✓
PFSEG50	SEG50	P07	✓	✓	✓	✓	✓	✓
PFSEG51	SEG51	P141	✓	—	—	—	—	—
PFSEG52	SEG52	P142	✓	—	—	—	—	—
PFSEG53	SEG53	P143	✓	—	—	—	—	—
PFSEG54	SEG54	P144	✓	—	—	—	—	—
PFSEG55	SEG55	P145	✓	—	—	—	—	—

a) Operation of ports that alternately function as COMxx/SEGxx pins

The functions of the common pins (COMxx) and segment output pins (SEGxx) are determined by the settings of the following registers: port mode control A registers (PMCAxx), port mode control T registers (PMCTxx), port mode registers (PMxx), and LCD port function registers (PFSEGx).

- P00 to P07, P15 to P17, P30 to P35, P46, P47, P50 to P57, P70 to P77, P90 to P97, and P141 to P145 (ports that do not serve as analog input pins (ANLxx))

Table 22 - 11 Settings of TSxx/COMxx/SEGxx Port Pin Function

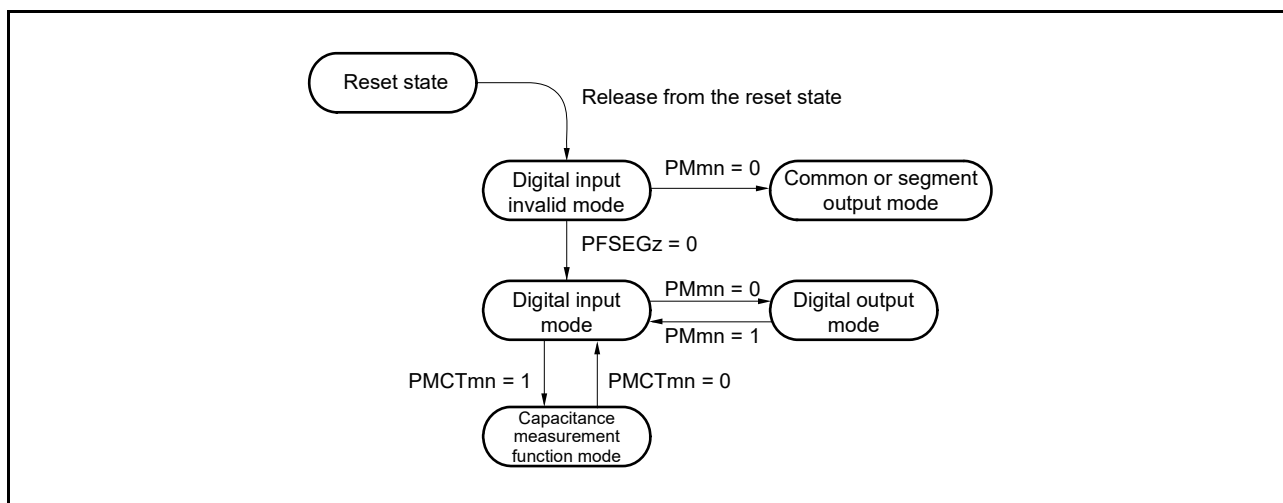
PMCTxx Bit of PMCTxx Register ^{Note}	PFCOMy or PFSEGz Bit of PFSEGx Register	PMxx Bit of PMxx Register	Pin Function	Initial State
0	1	1	Digital input invalid mode	✓
0	0	0	Digital output mode	—
0	0	1	Digital input mode	—
0	1	0	Common or segment output mode	—
1	0	×	Capacitance measurement function mode	—
Other than above			Setting prohibited	

Note PMCT00 to PMCT03, PMCT15 to PMCT17, PMCT46, PMCT47, PMCT130, and PMCT141 to PMCT145 are not provided.

Remark ×: Don't care

The following shows the TSxx/COMxx/SEGxx port pin function state transitions.

Figure 22 - 15 TSxx/COMxx/SEGxx Port Pin Function State Transitions



Caution Be sure to set the common or segment output mode before segment output starts (while the SCOC bit of LCD mode register 1 (LCDM1) is 0).

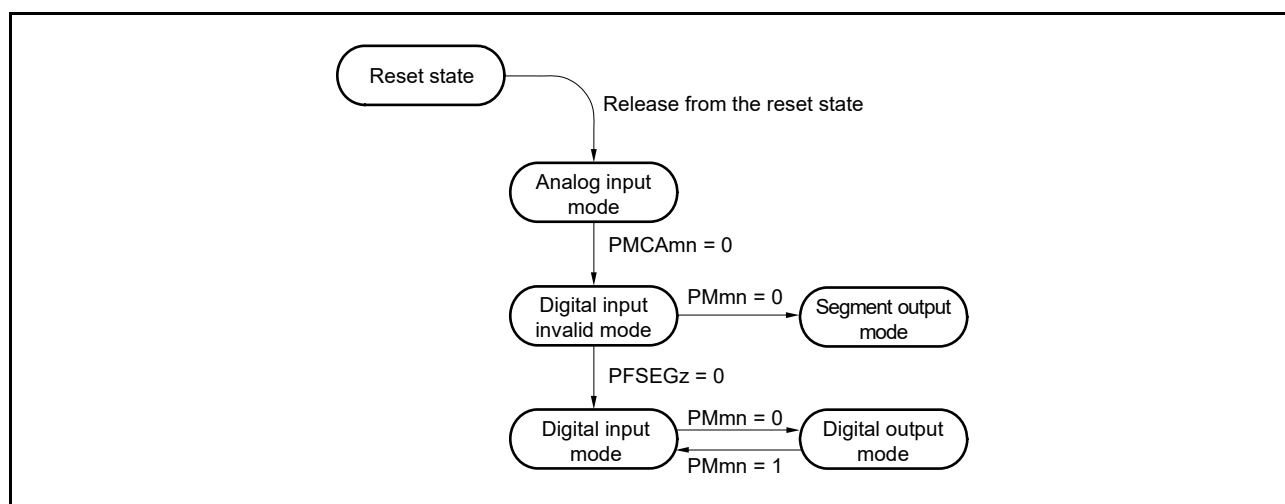
- P10 to P14 and P22 to P27 (ports that serve as analog input pins (ANLxx))

Table 22 - 12 Settings of ANLxx/SEGxx Port Pin Function

PMCAxx Bit of PMCAxx Register	PFSEGz Bit of PFSEG4 and PFSEG5 Registers	PMxx Bit of PMxx Register	Pin Function	Initial State
1	1	1	Analog input mode	✓
0	0	0	Digital output mode	—
0	0	1	Digital input mode	—
0	1	0	Segment output mode	—
0	1	1	Digital input invalid mode	—
Other than above			Setting prohibited	

The following shows the ANLxx/SEGxx port pin function state transitions.

Figure 22 - 16 ANLxx/SEGxx Port Pin Function State Transitions



Caution Be sure to set the segment output mode before segment output starts (while the SCOC bit of LCD mode register 1 (LCDM1) is 0).

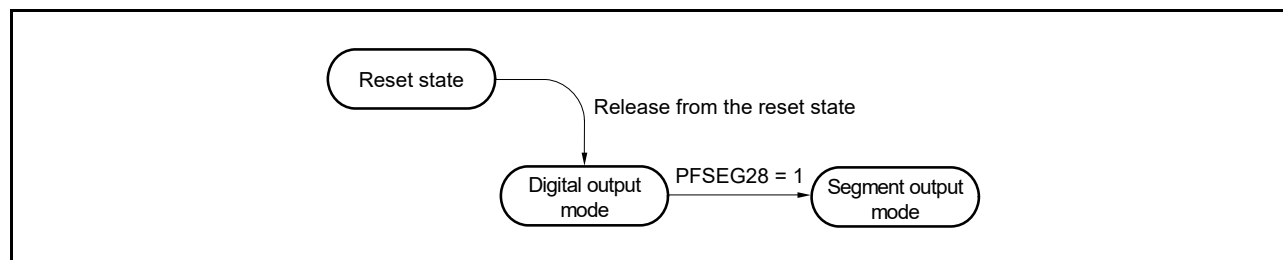
- SEG28/P130

Table 22 - 13 Settings of SEG28/P130 Port Pin Function

PFSEG28 Bit of PFSEG4 Register	Pin Function	Initial State
0	Digital output mode	✓
1	Segment output mode	—

The following shows the SEG28/P130 port pin function state transitions.

Figure 22 - 17 SEG28/P130 Port Pin Function State Transitions



Caution Be sure to set the segment output mode before segment output starts (while the SCOC bit of LCD mode register 1 (LCDM1) is 0).

22.3.9 Registers for controlling the port functions multiplexed with the common outputs and segment outputs of the LCD controller/driver

Set the following registers for controlling the port functions multiplexed with the common outputs and segment outputs of the LCD controller/driver.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.3 Pull-up resistor option registers (PUxx)**
- **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)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

22.4 LCD Display Data Registers

The LCD display data registers are mapped as shown in **Table 22 - 14**. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (1/8)

a) Other than six-time-slice and eight-time-slice (static, two-time-slice, three-time-slice, and four-time-slice)
(100 to 64 pins) (1/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	100-pin	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG0	F0400H	SEG0 (B-pattern area)				SEG0 (A-pattern area)				✓	✓	✓
SEG1	F0401H	SEG1 (B-pattern area)				SEG1 (A-pattern area)				✓	✓	✓
SEG2	F0402H	SEG2 (B-pattern area)				SEG2 (A-pattern area)				✓	✓	✓
SEG3	F0403H	SEG3 (B-pattern area)				SEG3 (A-pattern area)				✓	✓	✓
SEG4	F0404H	SEG4 (B-pattern area)				SEG4 (A-pattern area)				✓	✓	—
SEG5	F0405H	SEG5 (B-pattern area)				SEG5 (A-pattern area)				✓	✓	—
SEG6	F0406H	SEG6 (B-pattern area)				SEG6 (A-pattern area)				✓	✓	✓
SEG7	F0407H	SEG7 (B-pattern area)				SEG7 (A-pattern area)				✓	✓	✓
SEG8	F0408H	SEG8 (B-pattern area)				SEG8 (A-pattern area)				✓	✓	✓
SEG9	F0409H	SEG9 (B-pattern area)				SEG9 (A-pattern area)				✓	✓	—
SEG10	F040AH	SEG10 (B-pattern area)				SEG10 (A-pattern area)				✓	✓	—
SEG11	F040BH	SEG11 (B-pattern area)				SEG11 (A-pattern area)				✓	✓	✓
SEG12	F040CH	SEG12 (B-pattern area)				SEG12 (A-pattern area)				✓	✓	✓
SEG13	F040DH	SEG13 (B-pattern area)				SEG13 (A-pattern area)				✓	✓	—
SEG14	F040EH	SEG14 (B-pattern area)				SEG14 (A-pattern area)				✓	✓	—
SEG15	F040FH	SEG15 (B-pattern area)				SEG15 (A-pattern area)				✓	✓	—
SEG16	F0410H	SEG16 (B-pattern area)				SEG16 (A-pattern area)				✓	✓	✓
SEG17	F0411H	SEG17 (B-pattern area)				SEG17 (A-pattern area)				✓	✓	✓
SEG18	F0412H	SEG18 (B-pattern area)				SEG18 (A-pattern area)				✓	✓	✓
SEG19	F0413H	SEG19 (B-pattern area)				SEG19 (A-pattern area)				✓	✓	✓
SEG20	F0414H	SEG20 (B-pattern area)				SEG20 (A-pattern area)				✓	✓	✓
SEG21	F0415H	SEG21 (B-pattern area)				SEG21 (A-pattern area)				✓	✓	✓
SEG22	F0416H	SEG22 (B-pattern area)				SEG22 (A-pattern area)				✓	—	—
SEG23	F0417H	SEG23 (B-pattern area)				SEG23 (A-pattern area)				✓	—	—
SEG24	F0418H	SEG24 (B-pattern area)				SEG24 (A-pattern area)				✓	✓	—
SEG25	F0419H	SEG25 (B-pattern area)				SEG25 (A-pattern area)				✓	✓	—
SEG26	F041AH	SEG26 (B-pattern area)				SEG26 (A-pattern area)				✓	✓	—
SEG27	F041BH	SEG27 (B-pattern area)				SEG27 (A-pattern area)				✓	✓	—
SEG28	F041CH	SEG28 (B-pattern area)				SEG28 (A-pattern area)				✓	✓	—
SEG29	F041DH	SEG29 (B-pattern area)				SEG29 (A-pattern area)				✓	✓	✓

(Remark is listed on the next page.)

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (2/8)

a) Other than six-time-slice and eight-time-slice (static, two-time-slice, three-time-slice, and four-time-slice)
(100 to 64 pins) (2/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	100-pin	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG30	F041EH	SEG30 (B-pattern area)				SEG30 (A-pattern area)				✓	✓	—
SEG31	F041FH	SEG31 (B-pattern area)				SEG31 (A-pattern area)				✓	✓	—
SEG32	F0420H	SEG32 (B-pattern area)				SEG32 (A-pattern area)				✓	✓	—
SEG33	F0421H	SEG33 (B-pattern area)				SEG33 (A-pattern area)				✓	✓	✓
SEG34	F0422H	SEG34 (B-pattern area)				SEG34 (A-pattern area)				✓	✓	✓
SEG35	F0423H	SEG35 (B-pattern area)				SEG35 (A-pattern area)				✓	✓	✓
SEG36	F0424H	SEG36 (B-pattern area)				SEG36 (A-pattern area)				✓	✓	✓
SEG37	F0425H	SEG37 (B-pattern area)				SEG37 (A-pattern area)				✓	✓	✓
SEG38	F0426H	SEG38 (B-pattern area)				SEG38 (A-pattern area)				✓	✓	✓
SEG39	F0427H	SEG39 (B-pattern area)				SEG39 (A-pattern area)				✓	✓	✓
SEG40	F0428H	SEG40 (B-pattern area)				SEG40 (A-pattern area)				✓	✓	✓
SEG41	F0429H	SEG41 (B-pattern area)				SEG41 (A-pattern area)				✓	✓	✓
SEG42	F042AH	SEG42 (B-pattern area)				SEG42 (A-pattern area)				✓	✓	✓
SEG43	F042BH	SEG43 (B-pattern area)				SEG43 (A-pattern area)				✓	✓	✓
SEG44	F042CH	SEG44 (B-pattern area)				SEG44 (A-pattern area)				✓	✓	✓
SEG45	F042DH	SEG45 (B-pattern area)				SEG45 (A-pattern area)				✓	✓	✓
SEG46	F042EH	SEG46 (B-pattern area)				SEG46 (A-pattern area)				✓	✓	✓
SEG47	F042FH	SEG47 (B-pattern area)				SEG47 (A-pattern area)				✓	✓	✓
SEG48	F0430H	SEG48 (B-pattern area)				SEG48 (A-pattern area)				✓	✓	✓
SEG49	F0431H	SEG49 (B-pattern area)				SEG49 (A-pattern area)				✓	✓	✓
SEG50	F0432H	SEG50 (B-pattern area)				SEG50 (A-pattern area)				✓	✓	✓
SEG51	F0433H	SEG51 (B-pattern area)				SEG51 (A-pattern area)				✓	—	—
SEG52	F0434H	SEG52 (B-pattern area)				SEG52 (A-pattern area)				✓	—	—
SEG53	F0435H	SEG53 (B-pattern area)				SEG53 (A-pattern area)				✓	—	—
SEG54	F0436H	SEG54 (B-pattern area)				SEG54 (A-pattern area)				✓	—	—
SEG55	F0437H	SEG55 (B-pattern area)				SEG55 (A-pattern area)				✓	—	—

Remark ✓: Supported; —: Not supported

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (3/8)

b) Other than six-time-slice and eight-time-slice (static, two-time-slice, three-time-slice, and four-time-slice)
(52 to 44 pins) (1/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	52-pin	48-pin	44-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG0	F0400H	SEG0 (B-pattern area)				SEG0 (A-pattern area)				✓	✓	✓
SEG1	F0401H	SEG1 (B-pattern area)				SEG1 (A-pattern area)				✓	✓	✓
SEG2	F0402H	SEG2 (B-pattern area)				SEG2 (A-pattern area)				✓	✓	✓
SEG3	F0403H	SEG3 (B-pattern area)				SEG3 (A-pattern area)				✓	✓	✓
SEG4	F0404H	SEG4 (B-pattern area)				SEG4 (A-pattern area)				—	—	—
SEG5	F0405H	SEG5 (B-pattern area)				SEG5 (A-pattern area)				—	—	—
SEG6	F0406H	SEG6 (B-pattern area)				SEG6 (A-pattern area)				—	—	—
SEG7	F0407H	SEG7 (B-pattern area)				SEG7 (A-pattern area)				✓	—	—
SEG8	F0408H	SEG8 (B-pattern area)				SEG8 (A-pattern area)				✓	✓	—
SEG9	F0409H	SEG9 (B-pattern area)				SEG9 (A-pattern area)				—	—	—
SEG10	F040AH	SEG10 (B-pattern area)				SEG10 (A-pattern area)				—	—	—
SEG11	F040BH	SEG11 (B-pattern area)				SEG11 (A-pattern area)				✓	✓	✓
SEG12	F040CH	SEG12 (B-pattern area)				SEG12 (A-pattern area)				—	—	—
SEG13	F040DH	SEG13 (B-pattern area)				SEG13 (A-pattern area)				—	—	—
SEG14	F040EH	SEG14 (B-pattern area)				SEG14 (A-pattern area)				—	—	—
SEG15	F040FH	SEG15 (B-pattern area)				SEG15 (A-pattern area)				—	—	—
SEG16	F0410H	SEG16 (B-pattern area)				SEG16 (A-pattern area)				✓	—	—
SEG17	F0411H	SEG17 (B-pattern area)				SEG17 (A-pattern area)				✓	✓	—
SEG18	F0412H	SEG18 (B-pattern area)				SEG18 (A-pattern area)				✓	✓	✓
SEG19	F0413H	SEG19 (B-pattern area)				SEG19 (A-pattern area)				✓	✓	✓
SEG20	F0414H	SEG20 (B-pattern area)				SEG20 (A-pattern area)				—	—	—
SEG21	F0415H	SEG21 (B-pattern area)				SEG21 (A-pattern area)				—	—	—
SEG22	F0416H	SEG22 (B-pattern area)				SEG22 (A-pattern area)				—	—	—
SEG23	F0417H	SEG23 (B-pattern area)				SEG23 (A-pattern area)				—	—	—
SEG24	F0418H	SEG24 (B-pattern area)				SEG24 (A-pattern area)				—	—	—
SEG25	F0419H	SEG25 (B-pattern area)				SEG25 (A-pattern area)				—	—	—
SEG26	F041AH	SEG26 (B-pattern area)				SEG26 (A-pattern area)				—	—	—
SEG27	F041BH	SEG27 (B-pattern area)				SEG27 (A-pattern area)				—	—	—
SEG28	F041CH	SEG28 (B-pattern area)				SEG28 (A-pattern area)				—	—	—
SEG29	F041DH	SEG29 (B-pattern area)				SEG29 (A-pattern area)				✓	✓	✓
SEG30	F041EH	SEG30 (B-pattern area)				SEG30 (A-pattern area)				—	—	—
SEG31	F041FH	SEG31 (B-pattern area)				SEG31 (A-pattern area)				—	—	—
SEG32	F0420H	SEG32 (B-pattern area)				SEG32 (A-pattern area)				—	—	—
SEG33	F0421H	SEG33 (B-pattern area)				SEG33 (A-pattern area)				✓	✓	✓

(Remark is listed on the next page.)

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (4/8)

b) Other than six-time-slice and eight-time-slice (static, two-time-slice, three-time-slice, and four-time-slice)
(52 to 44 pins) (2/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	52-pin	48-pin	44-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG34	F0422H	SEG34 (B-pattern area)				SEG34 (A-pattern area)				✓	✓	✓
SEG35	F0423H	SEG35 (B-pattern area)				SEG35 (A-pattern area)				✓	✓	✓
SEG36	F0424H	SEG36 (B-pattern area)				SEG36 (A-pattern area)				✓	✓	✓
SEG37	F0425H	SEG37 (B-pattern area)				SEG37 (A-pattern area)				✓	✓	✓
SEG38	F0426H	SEG38 (B-pattern area)				SEG38 (A-pattern area)				✓	✓	—
SEG39	F0427H	SEG39 (B-pattern area)				SEG39 (A-pattern area)				✓	—	—
SEG40	F0428H	SEG40 (B-pattern area)				SEG40 (A-pattern area)				—	—	—
SEG41	F0429H	SEG41 (B-pattern area)				SEG41 (A-pattern area)				✓	✓	✓
SEG42	F042AH	SEG42 (B-pattern area)				SEG42 (A-pattern area)				✓	✓	✓
SEG43	F042BH	SEG43 (B-pattern area)				SEG43 (A-pattern area)				✓	✓	✓
SEG44	F042CH	SEG44 (B-pattern area)				SEG44 (A-pattern area)				—	—	—
SEG45	F042DH	SEG45 (B-pattern area)				SEG45 (A-pattern area)				—	—	—
SEG46	F042EH	SEG46 (B-pattern area)				SEG46 (A-pattern area)				—	—	—
SEG47	F042FH	SEG47 (B-pattern area)				SEG47 (A-pattern area)				—	—	—
SEG48	F0430H	SEG48 (B-pattern area)				SEG48 (A-pattern area)				✓	✓	✓
SEG49	F0431H	SEG49 (B-pattern area)				SEG49 (A-pattern area)				✓	✓	✓
SEG50	F0432H	SEG50 (B-pattern area)				SEG50 (A-pattern area)				✓	✓	✓
SEG51	F0433H	SEG51 (B-pattern area)				SEG51 (A-pattern area)				—	—	—
SEG52	F0434H	SEG52 (B-pattern area)				SEG52 (A-pattern area)				—	—	—
SEG53	F0435H	SEG53 (B-pattern area)				SEG53 (A-pattern area)				—	—	—
SEG54	F0436H	SEG54 (B-pattern area)				SEG54 (A-pattern area)				—	—	—
SEG55	F0437H	SEG55 (B-pattern area)				SEG55 (A-pattern area)				—	—	—

Remark ✓: Supported; —: Not supported

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (5/8)

<R> c) Six-time-slice and eight-time-slice (100 to 64 pins) (1/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	100-pin	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG0	F0400H	SEG0Note								✓	✓	✓
SEG1	F0401H	SEG1Note								✓	✓	✓
SEG2	F0402H	SEG2Note								✓	✓	✓
SEG3	F0403H	SEG3Note								✓	✓	✓
SEG4	F0404H	SEG4								✓	✓	—
SEG5	F0405H	SEG5								✓	✓	—
SEG6	F0406H	SEG6								✓	✓	✓
SEG7	F0407H	SEG7								✓	✓	✓
SEG8	F0408H	SEG8								✓	✓	✓
SEG9	F0409H	SEG9								✓	✓	—
SEG10	F040AH	SEG10								✓	✓	—
SEG11	F040BH	SEG11								✓	✓	✓
SEG12	F040CH	SEG12								✓	✓	✓
SEG13	F040DH	SEG13								✓	✓	—
SEG14	F040EH	SEG14								✓	✓	—
SEG15	F040FH	SEG15								✓	✓	—
SEG16	F0410H	SEG16								✓	✓	✓
SEG17	F0411H	SEG17								✓	✓	✓
SEG18	F0412H	SEG18								✓	✓	✓
SEG19	F0413H	SEG19								✓	✓	✓
SEG20	F0414H	SEG20								✓	✓	✓
SEG21	F0415H	SEG21								✓	✓	✓
SEG22	F0416H	SEG22								✓	—	—
SEG23	F0417H	SEG23								✓	—	—
SEG24	F0418H	SEG24								✓	✓	—
SEG25	F0419H	SEG25								✓	✓	—
SEG26	F041AH	SEG26								✓	✓	—
SEG27	F041BH	SEG27								✓	✓	—
SEG28	F041CH	SEG28								✓	✓	—
SEG29	F041DH	SEG29								✓	✓	—
SEG30	F041EH	SEG30								✓	✓	✓
SEG31	F041FH	SEG31								✓	✓	—
SEG32	F0420H	SEG32								✓	✓	—
SEG33	F0421H	SEG33								✓	✓	—
SEG34	F0422H	SEG34								✓	✓	✓
SEG35	F0423H	SEG35								✓	✓	✓

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

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (6/8)

<R> c) Six-time-slice and eight-time-slice (100 to 64 pins) (2/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	100-pin	80-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG36	F0424H	SEG36								✓	✓	✓
SEG37	F0425H	SEG37								✓	✓	✓
SEG38	F0426H	SEG38								✓	✓	✓
SEG39	F0427H	SEG39								✓	✓	✓
SEG40	F0428H	SEG40								✓	✓	✓
SEG41	F0429H	SEG41								✓	✓	✓
SEG42	F042AH	SEG42								✓	✓	✓
SEG43	F042BH	SEG43								✓	✓	✓
SEG44	F042CH	SEG44								✓	✓	✓
SEG45	F042DH	SEG45								✓	✓	✓
SEG46	F042EH	SEG46								✓	✓	✓
SEG47	F042FH	SEG47								✓	✓	✓
SEG48	F0430H	SEG48								✓	✓	✓
SEG49	F0431H	SEG49								✓	✓	✓
SEG50	F0432H	SEG50								✓	✓	✓
SEG51	F0433H	SEG51								✓	—	—
SEG52	F0434H	SEG52								✓	—	—
SEG53	F0435H	SEG53								✓	—	—
SEG54	F0436H	SEG54								✓	—	—
SEG55	F0437H	SEG55								✓	—	—

Note The COM4 to COM7 and SEG0 to SEG3 pin functions are respectively multiplexed with each other.

Remark ✓: Supported; —: Not supported

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (7/8)

<R> d) Six-time-slice and eight-time-slice (52 to 44 pins) (1/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	52-pin	48-pin	44-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG0	F0400H	SEG0Note								✓	✓	✓
SEG1	F0401H	SEG1Note								✓	✓	✓
SEG2	F0402H	SEG2Note								✓	✓	✓
SEG3	F0403H	SEG3Note								✓	✓	✓
SEG4	F0404H	SEG4								—	—	—
SEG5	F0405H	SEG5								—	—	—
SEG6	F0406H	SEG6								—	—	—
SEG7	F0407H	SEG7								✓	—	—
SEG8	F0408H	SEG8								✓	✓	—
SEG9	F0409H	SEG9								—	—	—
SEG10	F040AH	SEG10								—	—	—
SEG11	F040BH	SEG11								✓	✓	✓
SEG12	F040CH	SEG12								—	—	—
SEG13	F040DH	SEG13								—	—	—
SEG14	F040EH	SEG14								—	—	—
SEG15	F040FH	SEG15								—	—	—
SEG16	F0410H	SEG16								✓	—	—
SEG17	F0411H	SEG17								✓	✓	—
SEG18	F0412H	SEG18								✓	✓	✓
SEG19	F0413H	SEG19								✓	✓	✓
SEG20	F0414H	SEG20								—	—	—
SEG21	F0415H	SEG21								—	—	—
SEG22	F0416H	SEG22								—	—	—
SEG23	F0417H	SEG23								—	—	—
SEG24	F0418H	SEG24								—	—	—
SEG25	F0419H	SEG25								—	—	—
SEG26	F041AH	SEG26								—	—	—
SEG27	F041BH	SEG27								—	—	—
SEG28	F041CH	SEG28								—	—	—
SEG29	F041DH	SEG29								✓	✓	✓
SEG30	F041EH	SEG30								—	—	—
SEG31	F041FH	SEG31								—	—	—
SEG32	F0420H	SEG32								—	—	—
SEG33	F0421H	SEG33								✓	✓	✓
SEG34	F0422H	SEG34								✓	✓	✓
SEG35	F0423H	SEG35								✓	✓	✓

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

Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs (8/8)

<R> d) Six-time-slice and eight-time-slice (52 to 44 pins) (2/2)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	52-pin	48-pin	44-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0			
SEG36	F0424H	SEG36								✓	✓	✓
SEG37	F0425H	SEG37								✓	✓	✓
SEG38	F0426H	SEG38								✓	✓	—
SEG39	F0427H	SEG39								✓	—	—
SEG40	F0428H	SEG40								—	—	—
SEG41	F0429H	SEG41								✓	✓	✓
SEG42	F042AH	SEG42								✓	✓	✓
SEG43	F042BH	SEG43								✓	✓	✓
SEG44	F042CH	SEG44								—	—	—
SEG45	F042DH	SEG45								—	—	—
SEG46	F042EH	SEG46								—	—	—
SEG47	F042FH	SEG47								—	—	—
SEG48	F0430H	SEG48								✓	✓	✓
SEG49	F0431H	SEG49								✓	✓	✓
SEG50	F0432H	SEG50								✓	✓	✓
SEG51	F0433H	SEG51								—	—	—
SEG52	F0434H	SEG52								—	—	—
SEG53	F0435H	SEG53								—	—	—
SEG54	F0436H	SEG54								—	—	—
SEG55	F0437H	SEG55								—	—	—

Note The COM4 to COM7 and SEG0 to SEG3 pin functions are respectively multiplexed with each other.

Remark ✓: Supported; —: Not supported

To use the LCD display data register when the number of time slices is static, two, three, or four, the lower four bits and higher four bits of each address of the LCD display data register become an A-pattern area and a B-pattern area, respectively.

The correspondences between A-pattern area data and COM signals are as follows: bit 0 ↔ COM0, bit 1 ↔ COM1, bit 2 ↔ COM2, and bit 3 ↔ COM3.

The correspondences between B-pattern area data and COM signals are as follows: bit 4 ↔ COM0, bit 5 ↔ COM1, bit 6 ↔ COM2, and bit 7 ↔ COM3.

A-pattern area data will be displayed on the LCD panel when BLON = LCDSEL = 0 has been selected, and B-pattern area data will be displayed on the LCD panel when BLON = 0 and LCDSEL = 1 have been selected.

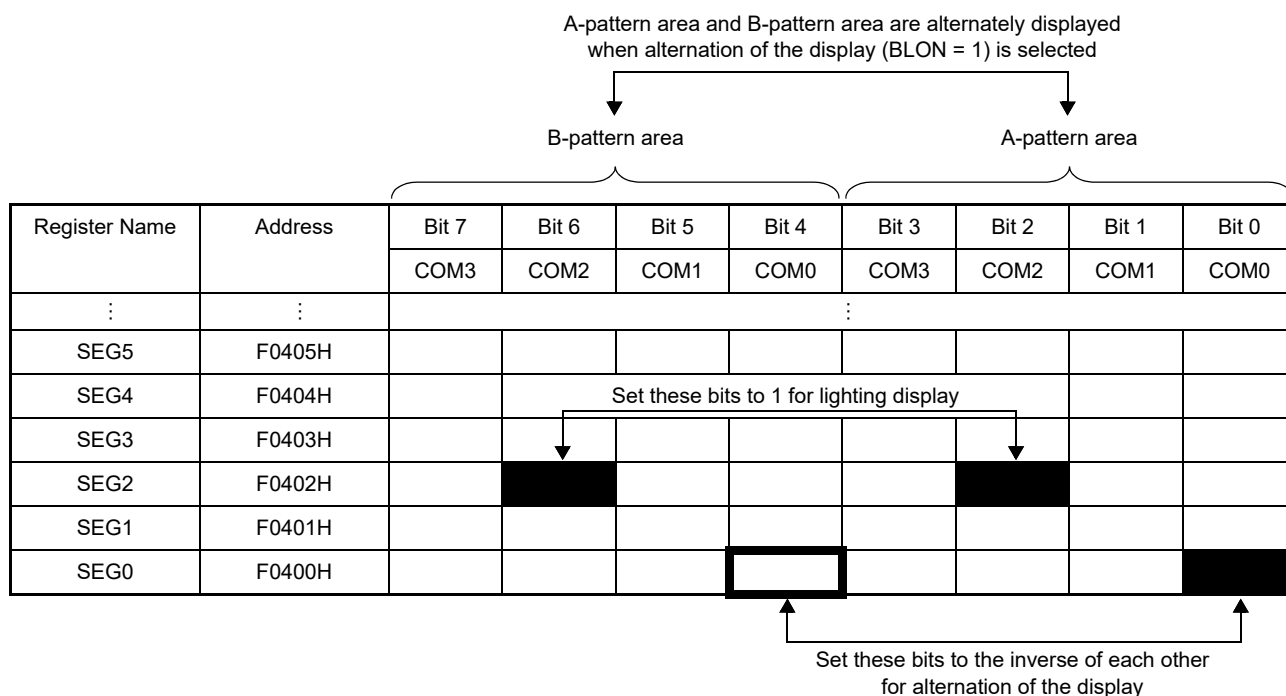
22.5 Selection of LCD Display Register

With RL78/L23, to use the LCD display data registers when the number of time slices is static, two, three, or four, the LCD display data register can be selected from the following three types, according to the BLON and LCDSEL bit settings.

- Displaying an A-pattern area data (lower four bits of LCD display data register)
- Displaying a B-pattern area data (higher four bits of LCD display data register)
- Alternately displaying the data in the A-pattern area and the data in the B-pattern area (alternation of the display according to the timing set by the LCDSEL2 and LCDSEL1 bits of the LCDM2 register)

Caution When the number of time slices is six or eight, LCD display data registers (A-pattern, B-pattern, or alternation) cannot be selected.

Figure 22 - 18 Example of Setting LCD Display Registers When Pattern Is Changed



22.5.1 A-pattern area and B-pattern area data display

When $BLON = LCDSEL = 0$, A-pattern area (lower four bits of the LCD display data register) data will be output as the LCD display register.

When $BLON = 0$, and $LCDSEL = 1$, B-pattern area (higher four bits of the LCD display data register) data will be output as the LCD display register.

See **22.4 LCD Display Data Registers** for details about the display area.

22.5.2 Alternation of the display (Alternately displaying A-pattern and B-pattern area data)

When $BLON = 1$ has been set, the data in the A-pattern area and the data in the B-pattern area will be alternately displayed, according to the timing set by the $LCDSEL2$ and $LCDSEL1$ bits of the $LCDM2$ register. For the timing settings of each interrupt ($INTRTC$, sub500ms, or $ELCITL2$), see **Section 10 Realtime Clock (RTC)** and **Section 12 32-bit Interval Timer (TML32)**. For alternation of the display on the LCD, set inverse values in B-pattern area bits corresponding to A-pattern area bits. For example, write 1 to bit 0 of F0400H, and set bit 4 of F0400H to 0. When alternation of the display on the LCD is not intended, set the same values. For example, write 1 to bit 2 of F0402H, and write 1 to bit 6 of F0402H to keep the display illuminated.

See **22.4 LCD Display Data Registers** for details about the display area.

The times for switching the display are shown below.

Figure 22 - 19 Switching Operation from A-Pattern Display to Alternation of the Display

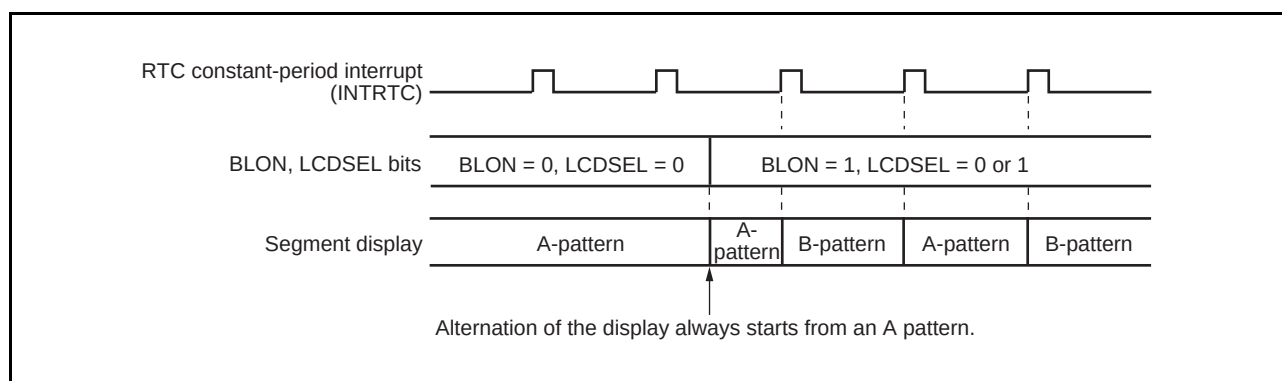
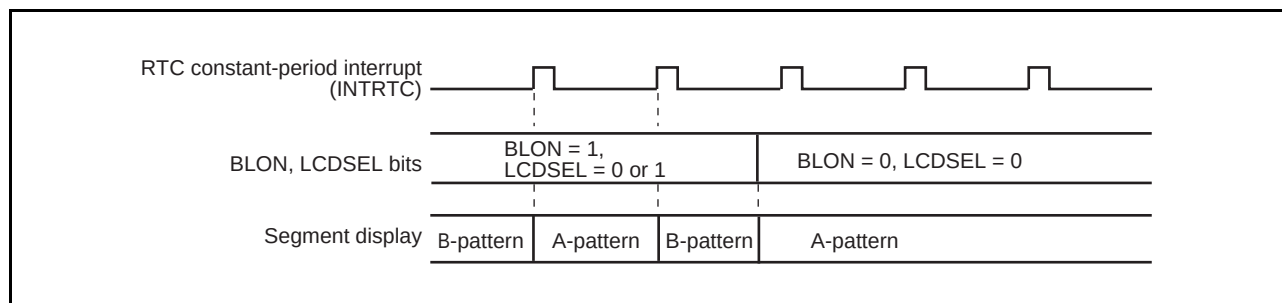


Figure 22 - 20 Switching Operation from Alternation of the Display to A-Pattern Display



22.6 Setting the LCD Controller/Driver

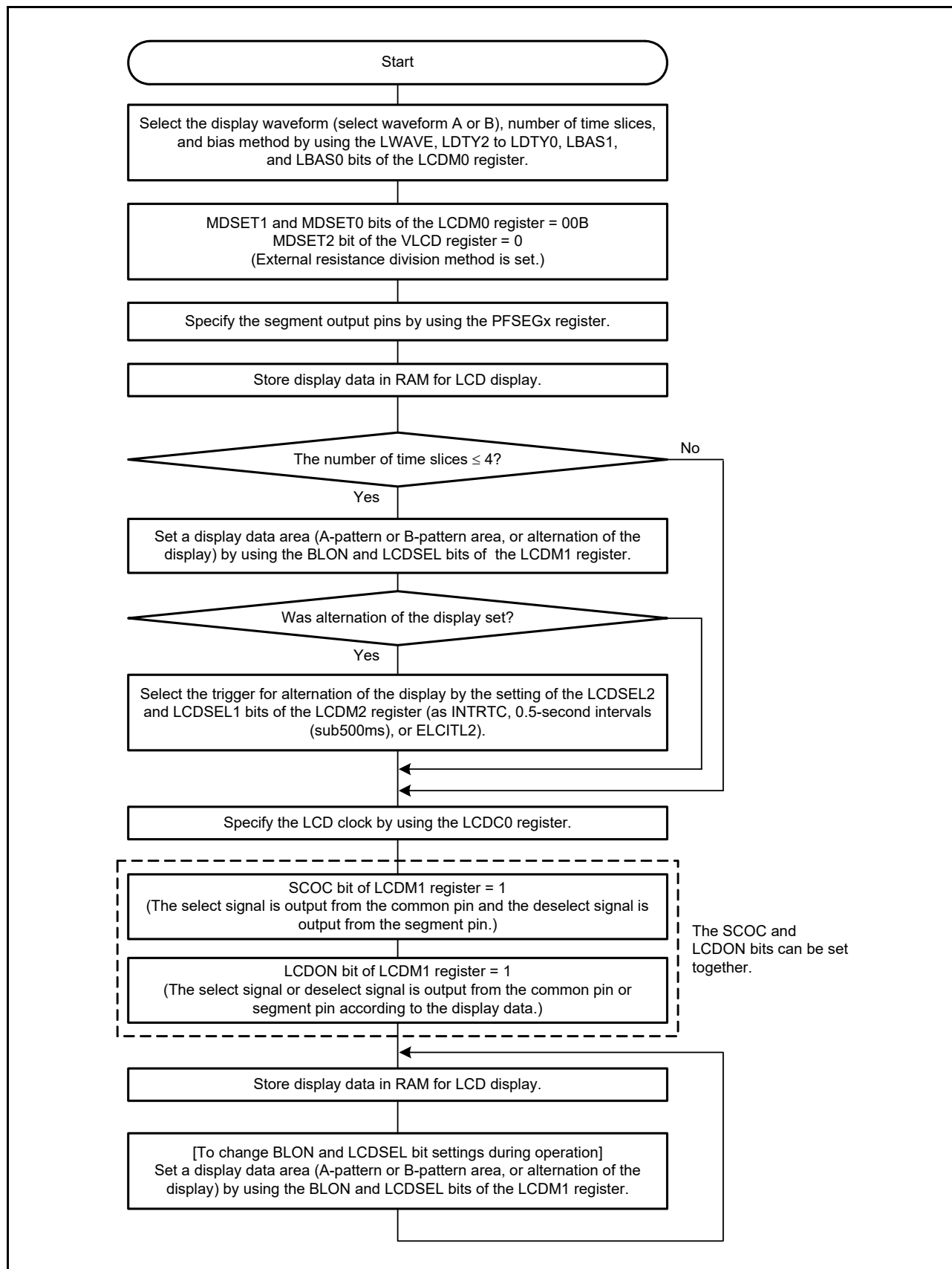
Set the LCD controller/driver using the following procedure.

Caution 1. To operate the LCD controller/driver, be sure to follow procedures shown in Figures 22 - 21 to 22 - 24. Unless these procedures are observed, the operation will not be guaranteed.

Caution 2. The steps shown in the flowcharts in Figures 22 - 21 to 22 - 24 are performed by the CPU.

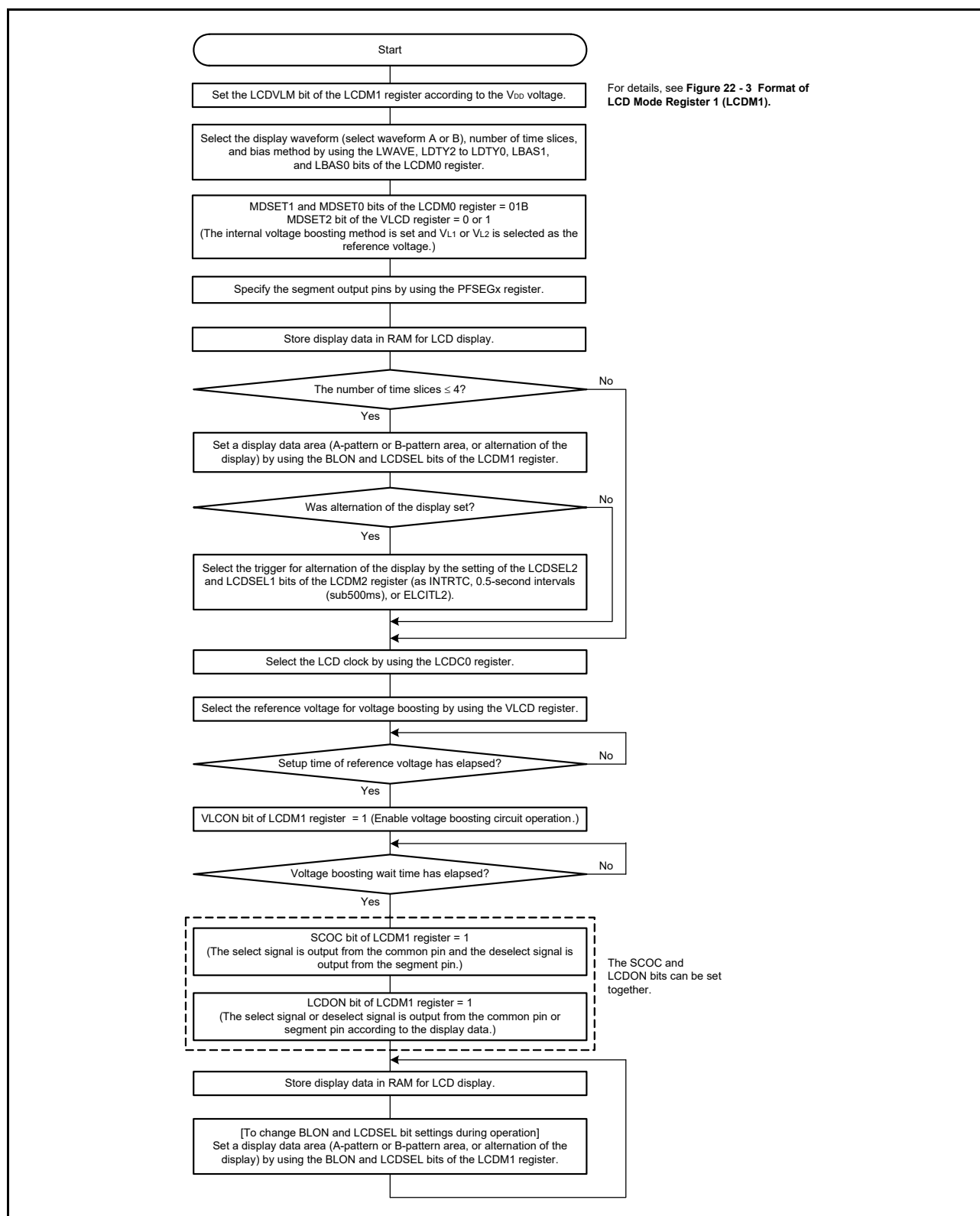
1. External resistance division method

Figure 22 - 21 External Resistance Division Method Setting Procedure



2. Internal voltage boosting method

Figure 22 - 22 Internal Voltage Boosting Method Setting Procedure

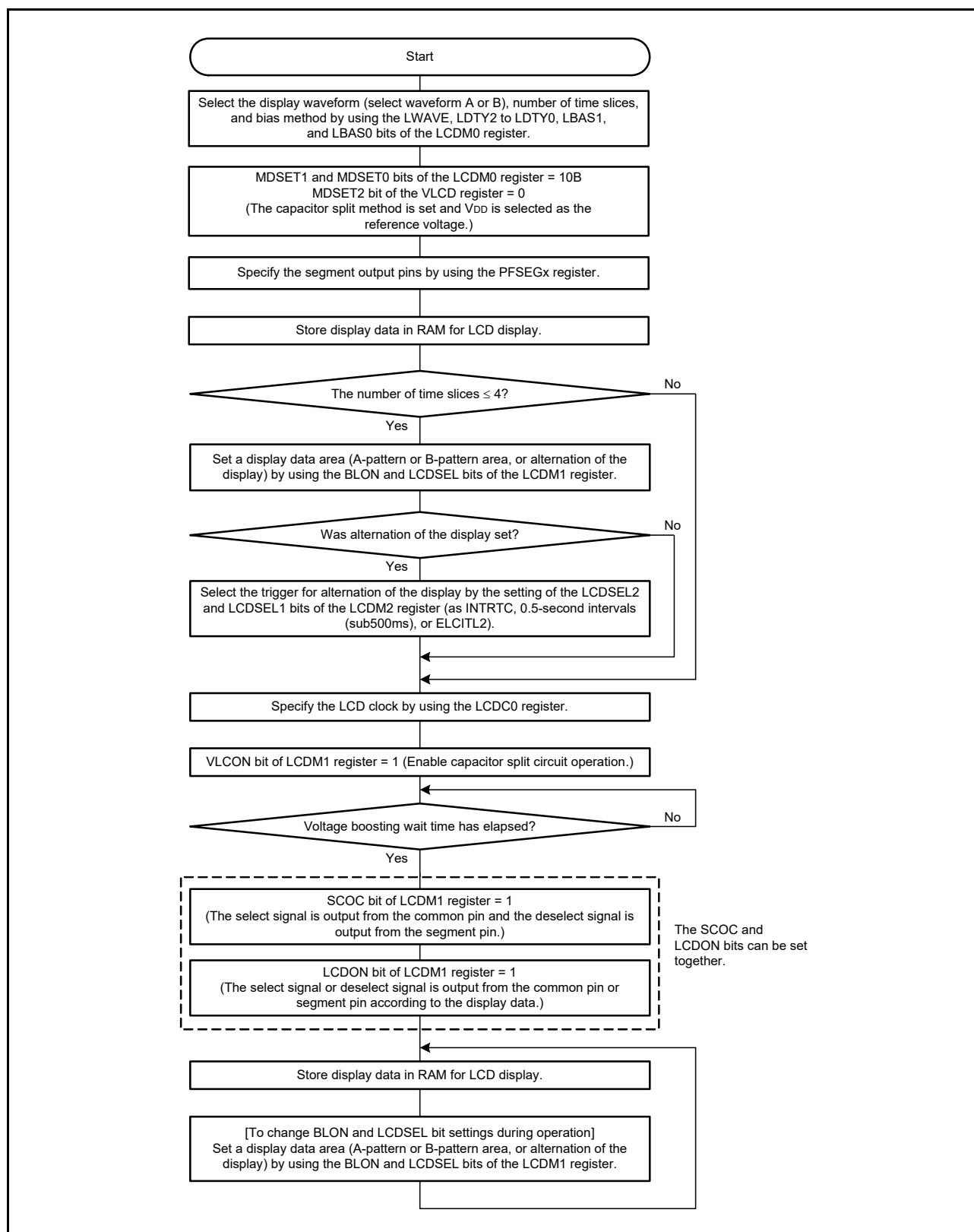


Caution 1. Wait until the setup time has elapsed even if not changing the setting of the VLCD register.

Caution 2. For the specifications of the reference voltage setup time and voltage boosting wait time, see Section 42 Electrical Characteristics.

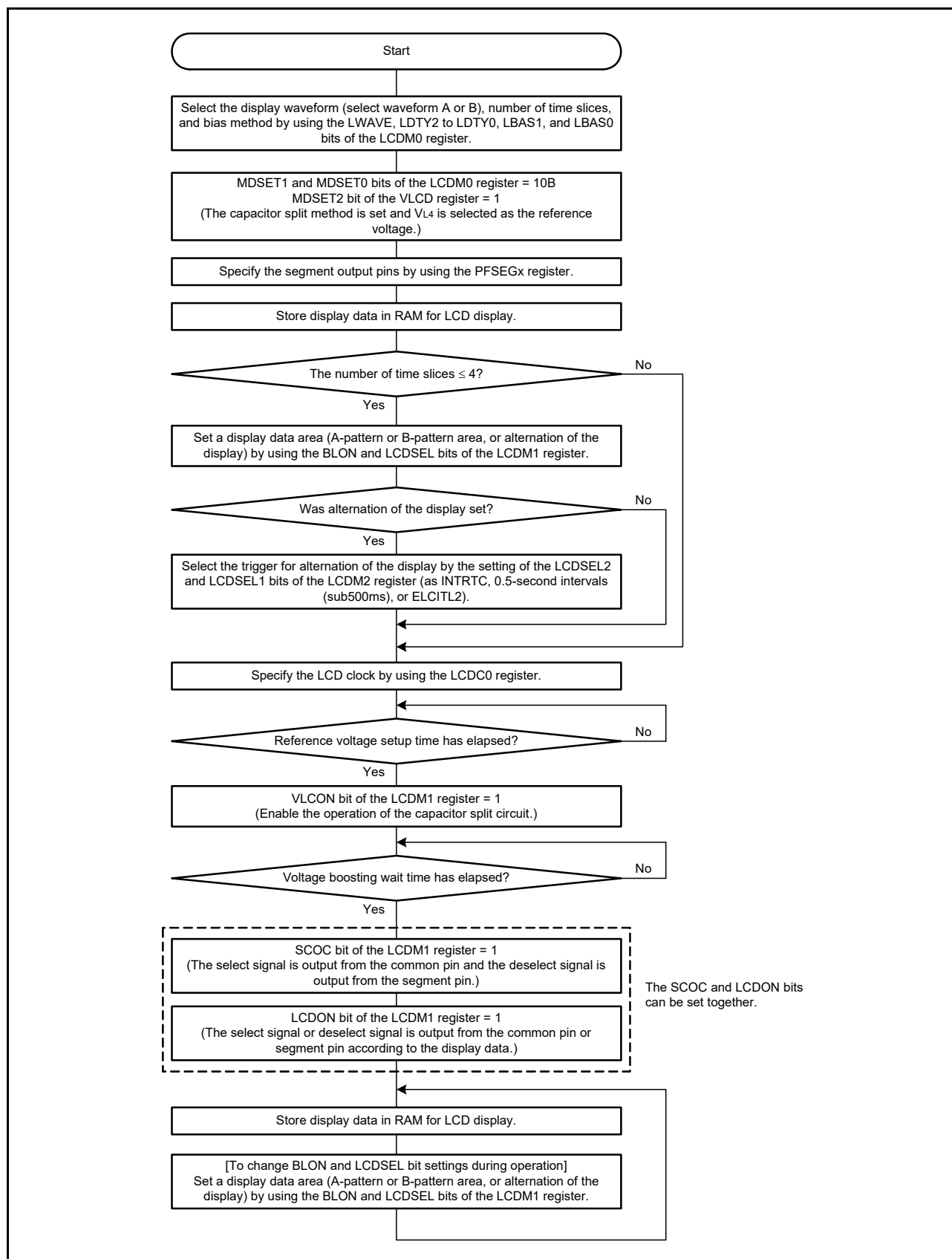
3. Capacitor split method

Figure 22 - 23 Procedure for Setting the Capacitor Split Method on the VDD Reference



Caution For the specifications of the voltage boosting wait time, see Section 42 Electrical Characteristics.

Figure 22 - 24 Procedure for Setting the Capacitor Split Method on the VL4 Reference



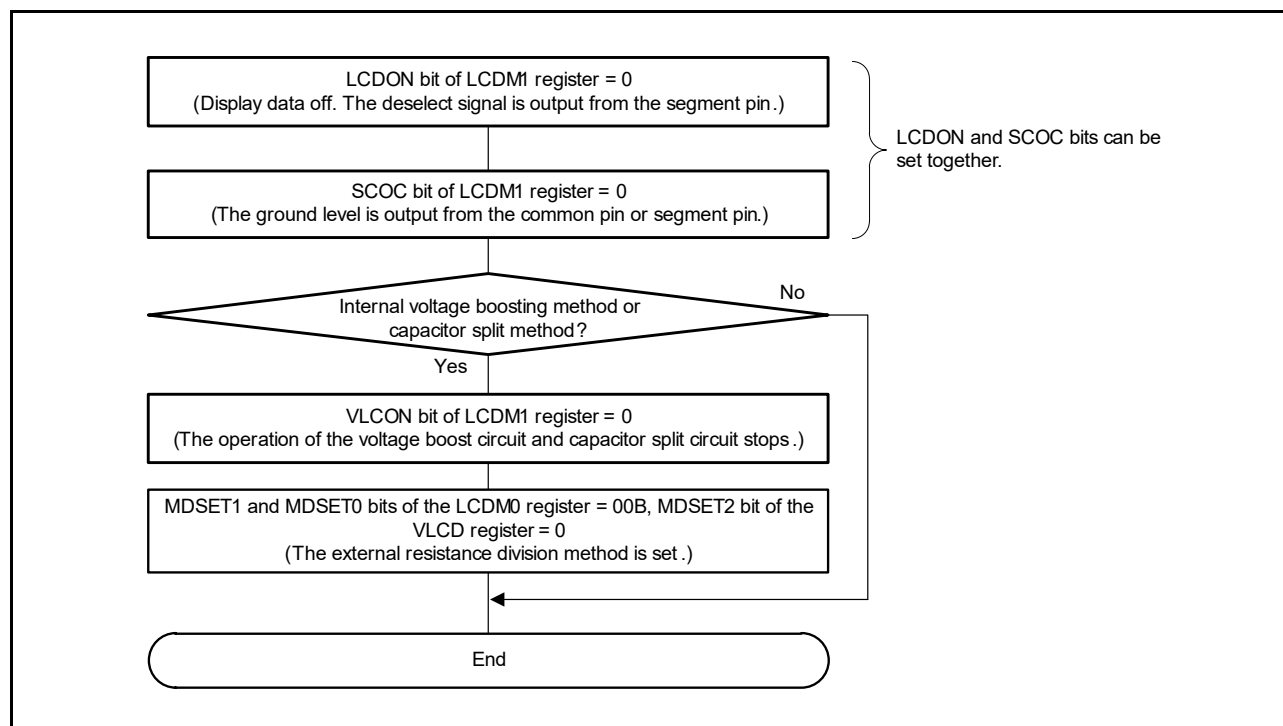
Caution For the specifications of the reference voltage setup time and voltage boosting wait time, see Section 42 Electrical Characteristics.

22.7 Operation Stop Procedure

To stop the operation of the LCD while it is displaying waveforms, follow the steps shown in the flowchart below.

The LCD stops operating when the LCDON bit of LCDM1 register and SCOC bit of the LCDM1 register are set to 0.

Figure 22 - 25 Operation Stop Procedure



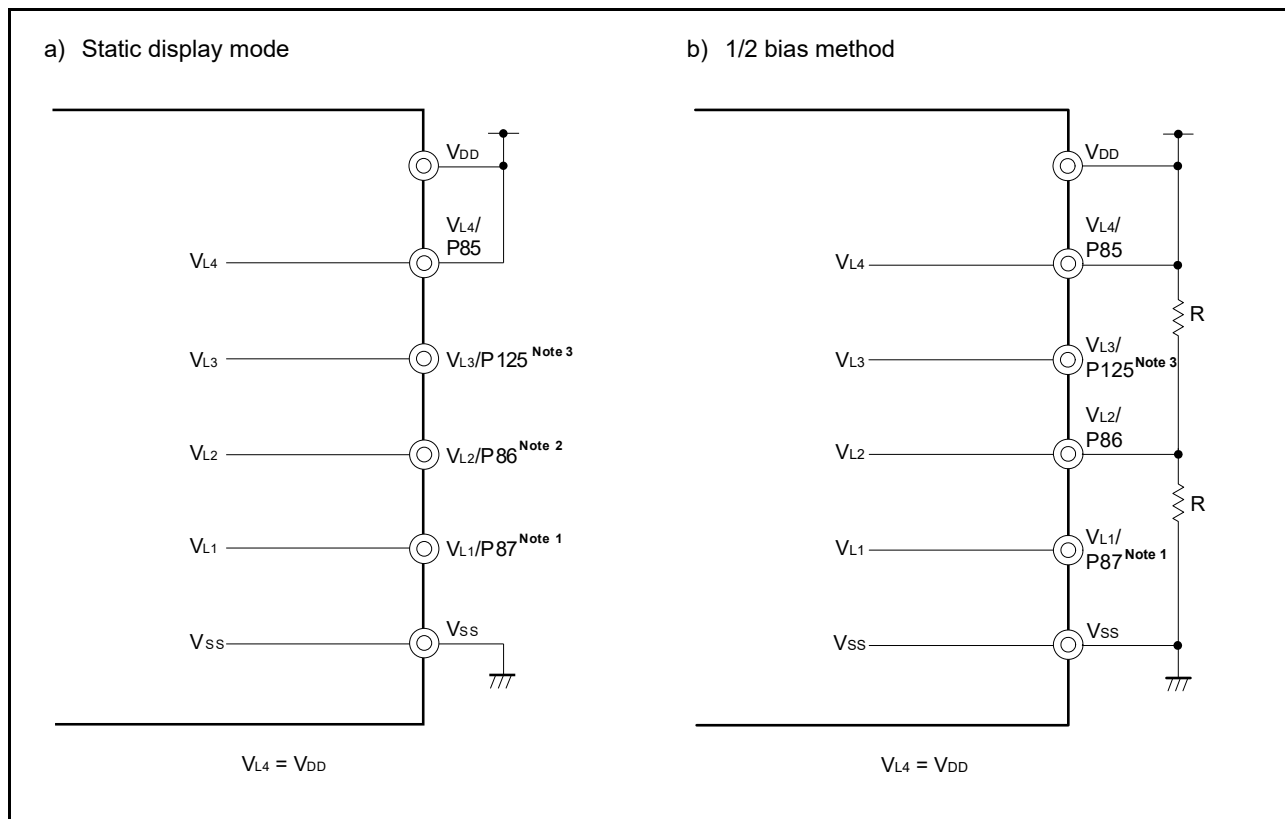
Caution Stopping the voltage boost/capacitor split circuits is prohibited while the display is on (SCOC and LCDON bits of LCDM1 register = 11B). Otherwise, the operation will not be guaranteed. Be sure to turn off display (SCOC and LCDON bits of LCDM1 register = 10B) before stopping the voltage boost/capacitor split circuits (VLCON bit of LCDM1 register = 0).

22.8 Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4

22.8.1 External resistance division method

Figure 22 - 26 shows an example of connecting the LCD drive voltages for each bias method.

Figure 22 - 26 Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)



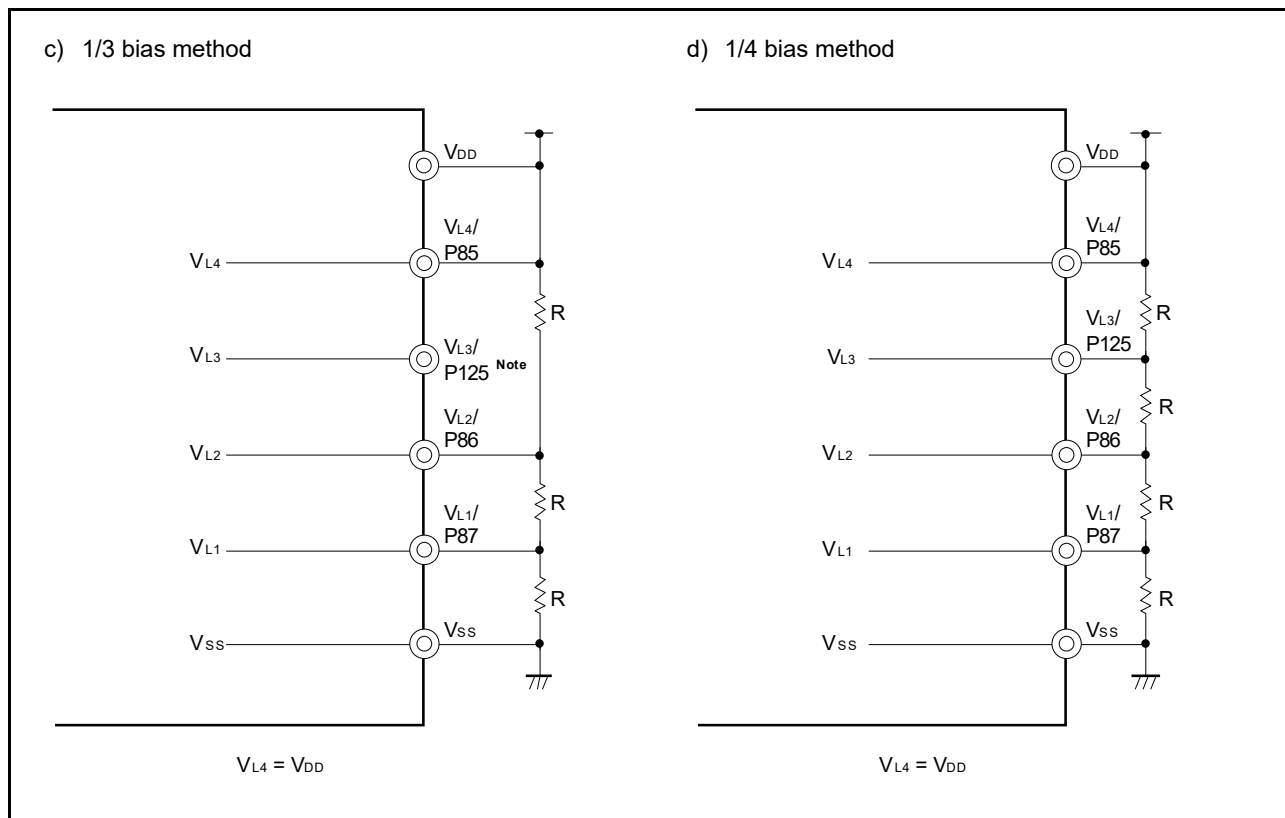
Note 1. VL1 can be used as port-pin P87.

Note 2. VL2 can be used as port-pin P86.

Note 3. VL3 can be used as port-pin P125.

Caution CAPL can be used as port-pin P126 and CAPH can be used as port-pin P127.

Figure 22 - 26 Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)



Note VL3 can be used as port-pin P125.

Caution 1. CAPL can be used as port-pin P126 and CAPH can be used as port-pin P127.

Caution 2. The reference resistance “R” value for external resistance division is 10 kΩ to 1 MΩ. In addition, to stabilize the potential of the VL1 to VL4 pins, connect a capacitor between each of pins VL1 to VL4 and the GND pin as needed. The reference capacitance is about 0.47 μF but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust and determine the capacitance.

22.8.2 Internal voltage boosting method

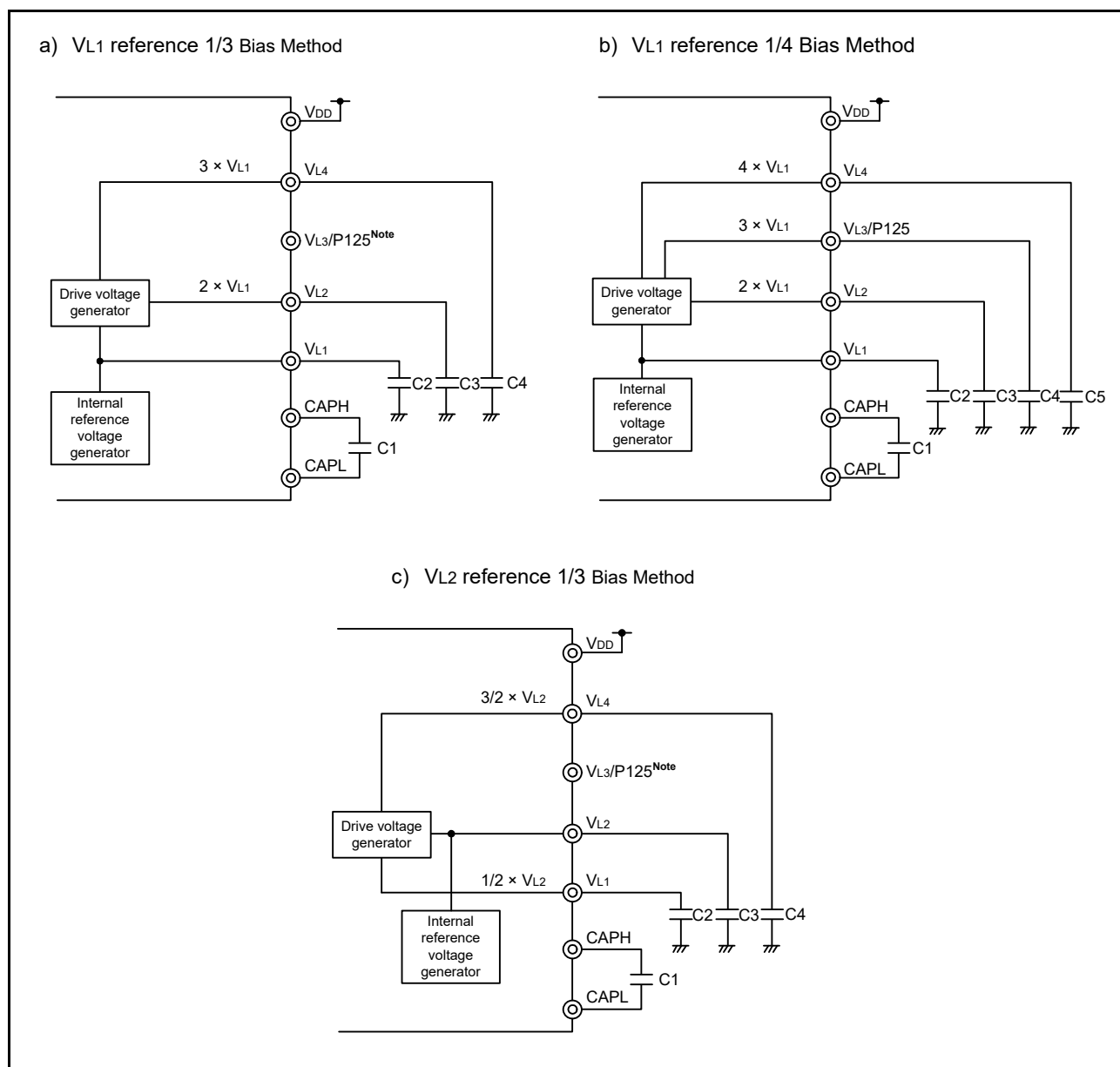
The RL78/L23 contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors ($0.47\ \mu\text{F}\pm 30\%$) are used to generate an LCD drive voltage. For the internal voltage boosting method on the VL1 reference, only 1/3 bias method or 1/4 bias method can be used. For the internal voltage boosting method on the VL2 reference, only 1/3 bias method can be used. The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in VDD, because it is a power supply separate from the main unit.

In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

Table 22 - 15 LCD Drive Voltages (Internal Voltage Boosting Method)

Bias Method LCD Drive Voltage Pin	1/3 Bias Method		1/4 Bias Method
	VL1 reference	VL2 reference	VL1 reference
VL4	$3 \times \text{VL1}$	$3/2 \times \text{VL2}$	$4 \times \text{VL1}$
VL3	—	—	$3 \times \text{VL1}$
VL2	$2 \times \text{VL1}$	LCD reference voltage	$2 \times \text{VL1}$
VL1	LCD reference voltage	$1/2 \times \text{VL2}$	LCD reference voltage

Figure 22 - 27 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)



Note V_{L3} can be used as port-pin P125.

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

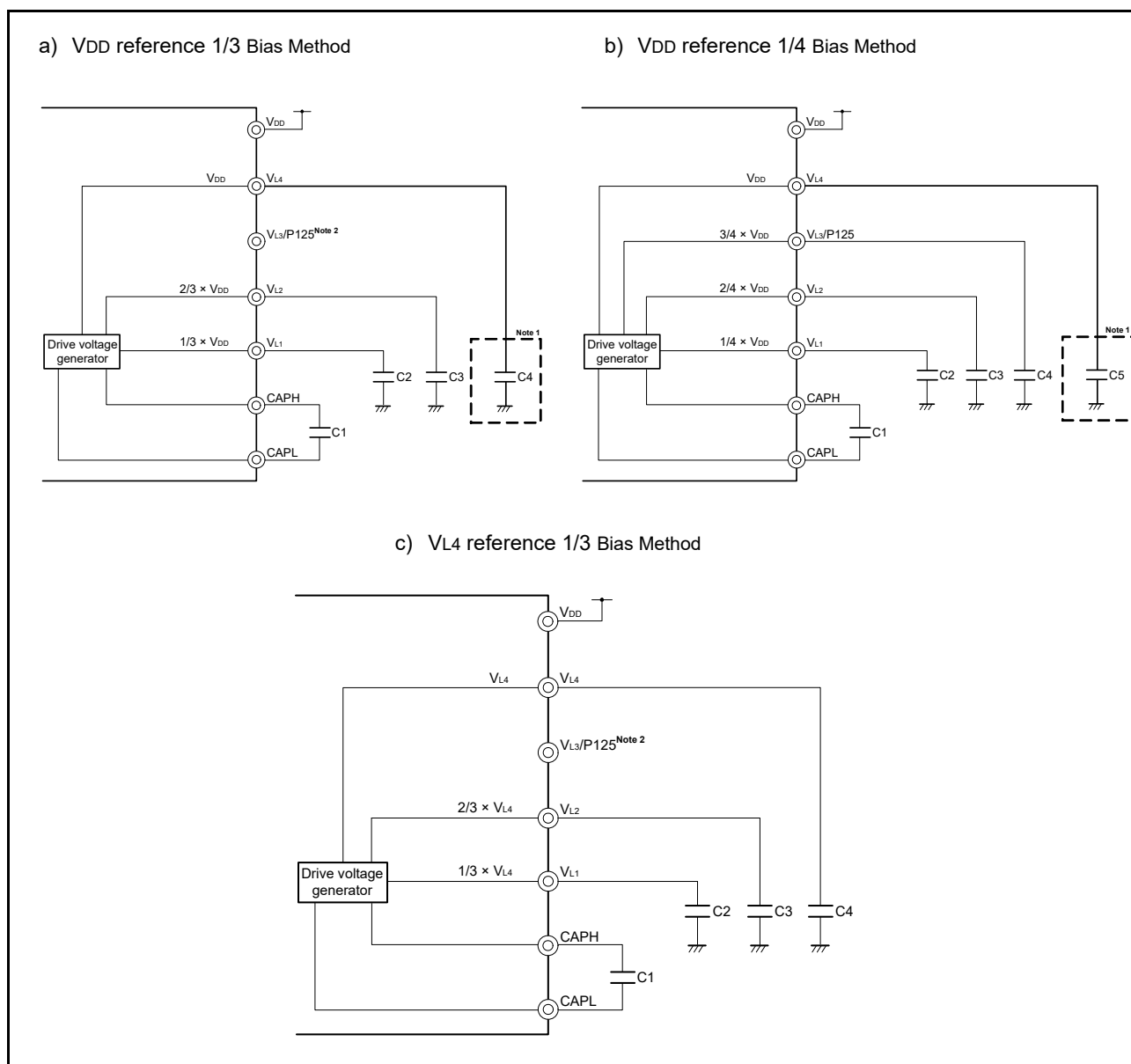
22.8.3 Capacitor split method

The RL78/L23 contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors ($0.47\ \mu\text{F} \pm 30\%$) are used to generate an LCD drive voltage. For the capacitor split method on the VDD reference, only 1/3 bias method or 1/4 bias method can be used. For the capacitor split method on the VL4 reference, only 1/3 bias method can be used. Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

Table 22 - 16 LCD Drive Voltages (Capacitor Split Method)

Bias Method LCD Drive Voltage Pin	1/3 Bias Method		1/4 Bias Method
	VDD reference	VL4 reference	VDD reference
VL4	VDD	VL4	VDD
VL3	—	—	$3/4 \times \text{VL4}$
VL2	$2/3 \times \text{VL4}$	$2/3 \times \text{VL4}$	$2/4 \times \text{VL4}$
VL1	$1/3 \times \text{VL4}$	$1/3 \times \text{VL4}$	$1/4 \times \text{VL4}$

Figure 22 - 28 Examples of LCD Drive Power Connections (Capacitor Split Method)



Note 1. As the VL4 pin is internally connected to the VDD pin, the capacitor is not always essential. Note, however, that selection of the internal voltage boosting method requires connection of this capacitor.

In addition, when only capacitor split method is in use, the capacitor can be connected as a means for stabilizing the voltage supplied to the LCD.

Note 2. VL3 can be used as port-pin P125.

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

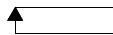
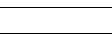
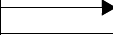
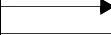
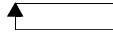
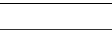
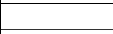
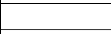
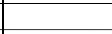
22.9 Common and Segment Signals

Each pixel of the LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, V_{LCD}). The pixels turn off when the potential difference becomes lower than V_{LCD} . Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, this LCD panel is driven by AC voltage.

1. Common signals

Each common signal is selected sequentially according to a specified number of time slices at the timing listed in **Table 22 - 17**. In the static display mode, the same signal is output to COM0 to COM3. In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open. In time slice modes other than six-time-slice and eight-time-slice modes, leave the COM4 to COM7 pins open-circuit or use them as segment pins. In six-time-slice mode, leave the COM6 and COM7 pins open-circuit or use them as segment pins.

Table 22 - 17 COM Signals

COM Signal Number of Time Slices	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					Note	Note	Note	Note
Two-time-slice mode			Open	Open	Note	Note	Note	Note
Three-time-slice mode				Open	Note	Note	Note	Note
Four-time-slice mode					Note	Note	Note	Note
Six-time-slice mode							Note	Note
Eight-time-slice mode								

Note Leave the pins open-circuit or use them as segment pins.

2. Segment signals

Segment signals correspond to LCD display data registers (see **22.4 LCD Display Data Registers**). When the number of time slices is eight, bits 0 to 7 of each display data register are read in synchronization with COM0 to COM7, respectively. When the number of time slices is six, bits 0 to 5 of each display data register are read in synchronization with COM0 to COM5, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG4 to SEG55). When the number of time slices is not six or eight, bits 0 to 3 of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bits 4 to 7 of each byte in B-pattern area are read in synchronization with COM0 to COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins (SEG0 to SEG55).

Check, with the information given above, what combination of front-surface electrodes (corresponding to the segment signals) and rear-surface electrodes (corresponding to the common signals) forms display patterns in the LCD display data register, and write the bit data that corresponds to the desired display pattern on a one-to-one basis.

Remark The mounted segment output pins vary depending on the product.

- 44-pin products: SEG0 to SEG3, SEG11, SEG18, SEG19, SEG29, SEG33 to SEG37, SEG41 to SEG43, SEG48 to SEG50
- 48-pin products: SEG0 to SEG3, SEG8, SEG11, SEG17 to SEG19, SEG29, SEG33 to SEG38, SEG41 to SEG43, SEG48 to SEG50
- 52-pin products: SEG0 to SEG3, SEG7, SEG8, SEG11, SEG16 to SEG19, SEG29, SEG33 to SEG39, SEG41 to SEG43, SEG48 to 50
- 64-pin products: SEG0 to SEG3, SEG6 to SEG8, SEG11, SEG12, SEG16 to SEG21, SEG29, SEG33 to SEG50
- 80-pin products: SEG0 to SEG21, SEG24 to SEG50
- 100-pin products: SEG0 to SEG55

3. Output waveforms of common and segment signals

The voltages listed in **Table 22 - 18** are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 22 - 18 LCD Drive Voltage

a) Static display mode

Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L4}/V_{SS}
Common Signal			
V_{L4}/V_{SS}		$-V_{LCD}/+V_{LCD}$	0 V/0 V

b) 1/2 bias method

Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L4}/V_{SS}
Common Signal			
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
Deselect signal level	V_{L2}	$-\frac{1}{2} V_{LCD} / +\frac{1}{2} V_{LCD}$	$+\frac{1}{2} V_{LCD} / -\frac{1}{2} V_{LCD}$

c) 1/3 bias method (waveform A or B)

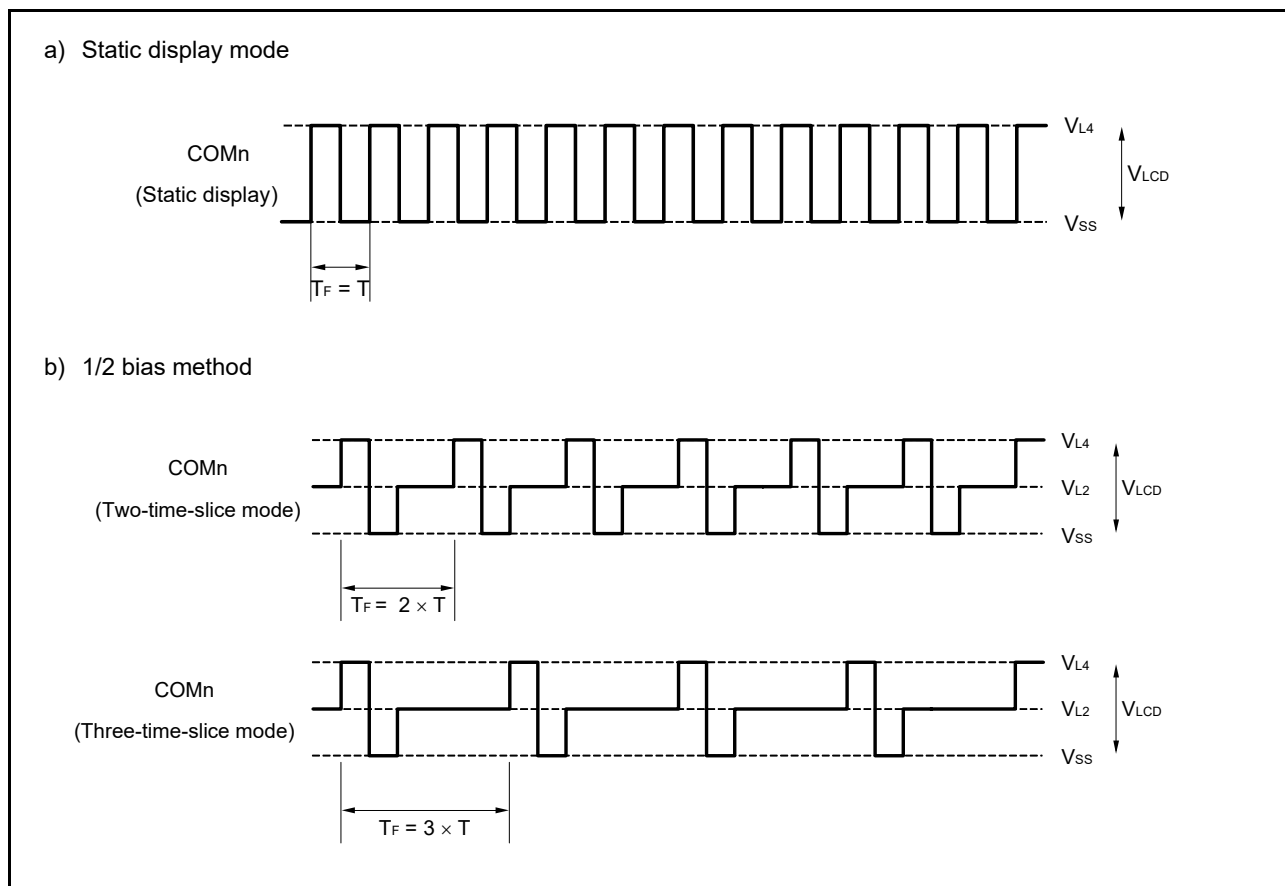
Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L2}/V_{L1}
Common Signal			
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3} V_{LCD} / +\frac{1}{3} V_{LCD}$
Deselect signal level	V_{L1}/V_{L2}	$-\frac{1}{3} V_{LCD} / +\frac{1}{3} V_{LCD}$	$+\frac{1}{3} V_{LCD} / -\frac{1}{3} V_{LCD}$

d) 1/4 bias method (waveform A or B)

Segment Signal		Select Signal Level	Deselect Signal Level
		V_{SS}/V_{L4}	V_{L2}
Common Signal			
Select signal level	V_{L4}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{2} V_{LCD} / +\frac{1}{2} V_{LCD}$
Deselect signal level	V_{L1}/V_{L3}	$-\frac{1}{4} V_{LCD} / +\frac{1}{4} V_{LCD}$	$+\frac{1}{4} V_{LCD} / -\frac{1}{4} V_{LCD}$

Figure 22 - 29 shows the common signal waveforms, and **Figure 22 - 30** shows the voltages and phases of the common and segment signals.

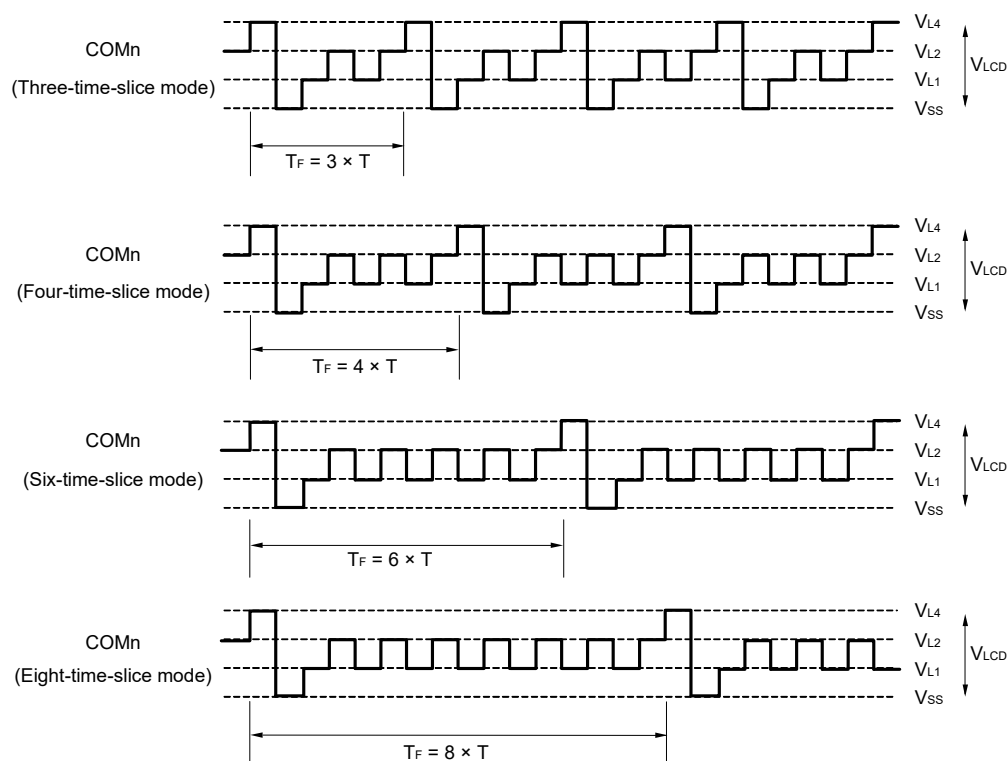
Figure 22 - 29 Common Signal Waveforms (1/2)



Remark T: One LCD clock period; T_F : Frame frequency

Figure 22 - 29 Common Signal Waveforms (2/2)

c) 1/3 bias method

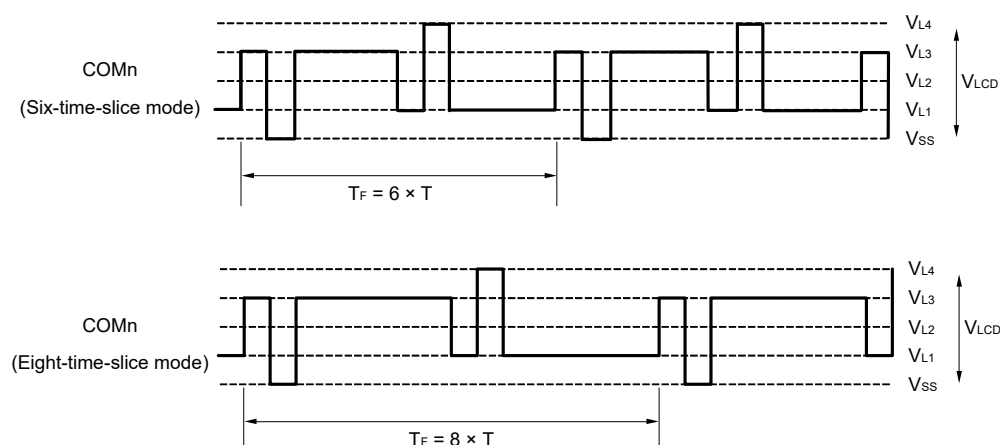


< Example of calculation of LCD frame frequency (When four-time slot mode is used) >

LCD clock: $32768/2^7 = 256$ Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 64 Hz

d) 1/4 bias method



< Example of calculation of LCD frame frequency (When eight-time slot mode is used) >

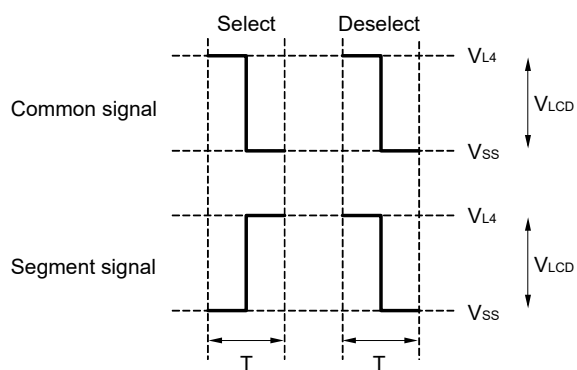
LCD clock: $32768/2^7 = 256$ Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 32 Hz

Remark T: One LCD clock period; TF: Frame frequency

Figure 22 - 30 Voltages and Phases of Common and Segment Signals (1/3)

a) Static display mode (waveform A)



b) 1/2 bias method (waveform A)

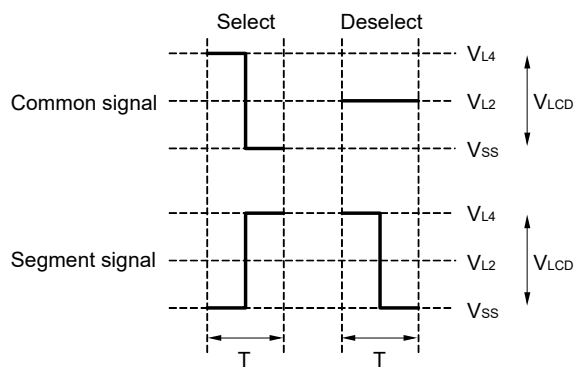
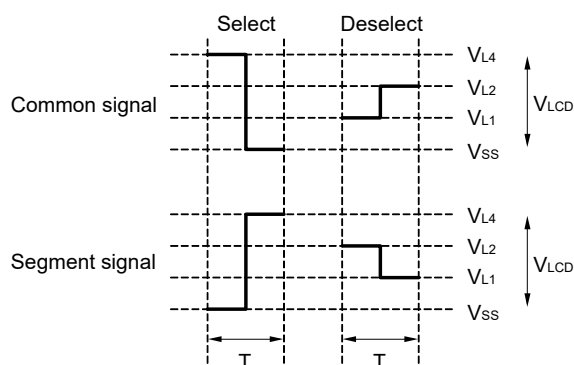
**Remark** T: One LCD clock period

Figure 22 - 30 Voltages and Phases of Common and Segment Signals (2/3)

c) 1/3 bias method (waveform A)



d) 1/3 bias method (waveform B)

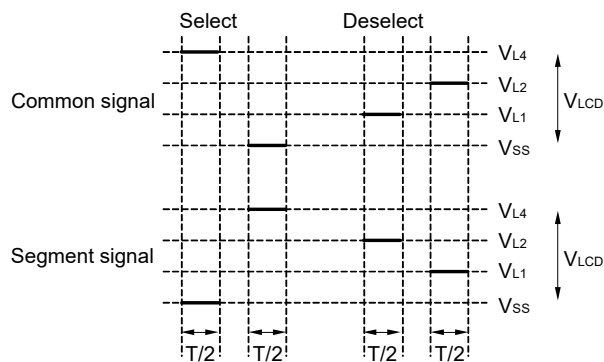
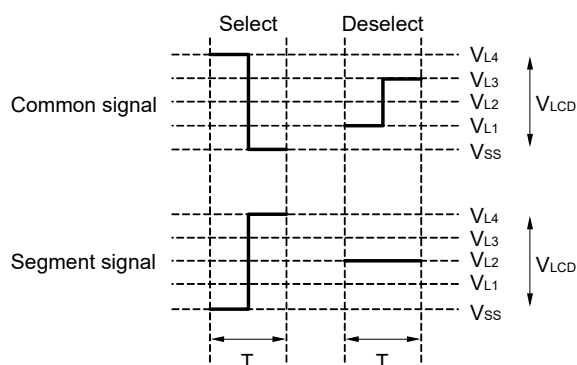
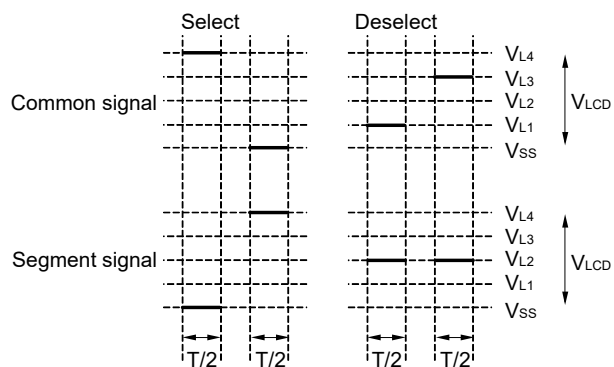
**Remark** T: One LCD clock period

Figure 22 - 30 Voltages and Phases of Common and Segment Signals (3/3)

e) 1/4 bias method (waveform A)



f) 1/4 bias method (waveform B)

**Remark** T: One LCD clock period

22.10 Display Modes

22.10.1 Static display example

Figure 22 - 32 shows how the three-digit LCD panel having the display pattern shown in **Figure 22 - 31** is connected to the segment signals (SEG0 to SEG23) and the common signal (COM0). This example displays data “12.3” in the LCD panel. The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “2.” (2.) displayed in the second digit. To display “2.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG8 to SEG15 pins according to **Table 22 - 19** at the timing of the common signal COM0; see **Figure 22 - 31** for the relationship between the segment signals and LCD segments.

Table 22 - 19 Select and Deselect Voltages (COM0)

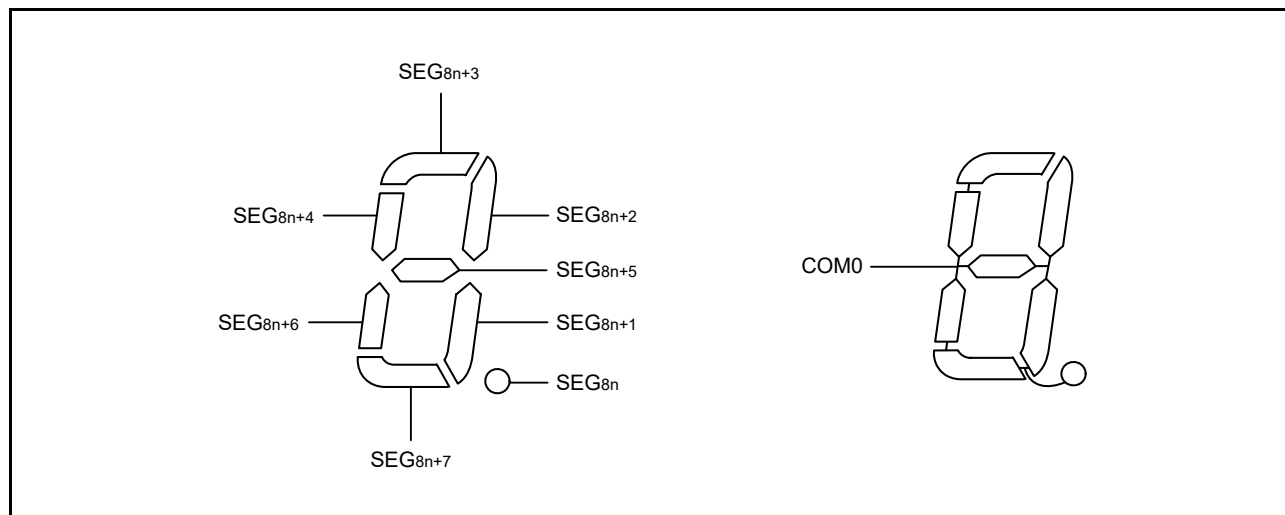
Segment	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to the above, it is determined that the bit-0 pattern of the display data register locations (F0408H to F040FH) must be 10110111.

Figure 22 - 33 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternate rectangle waveform, +VLCD/-VLCD, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the drive capability.

Figure 22 - 31 Static LCD Display Pattern and Electrode Connections



Remark 100-pin products: n = 0 to 6

Figure 22 - 32 Example of Connecting Static LCD Panel

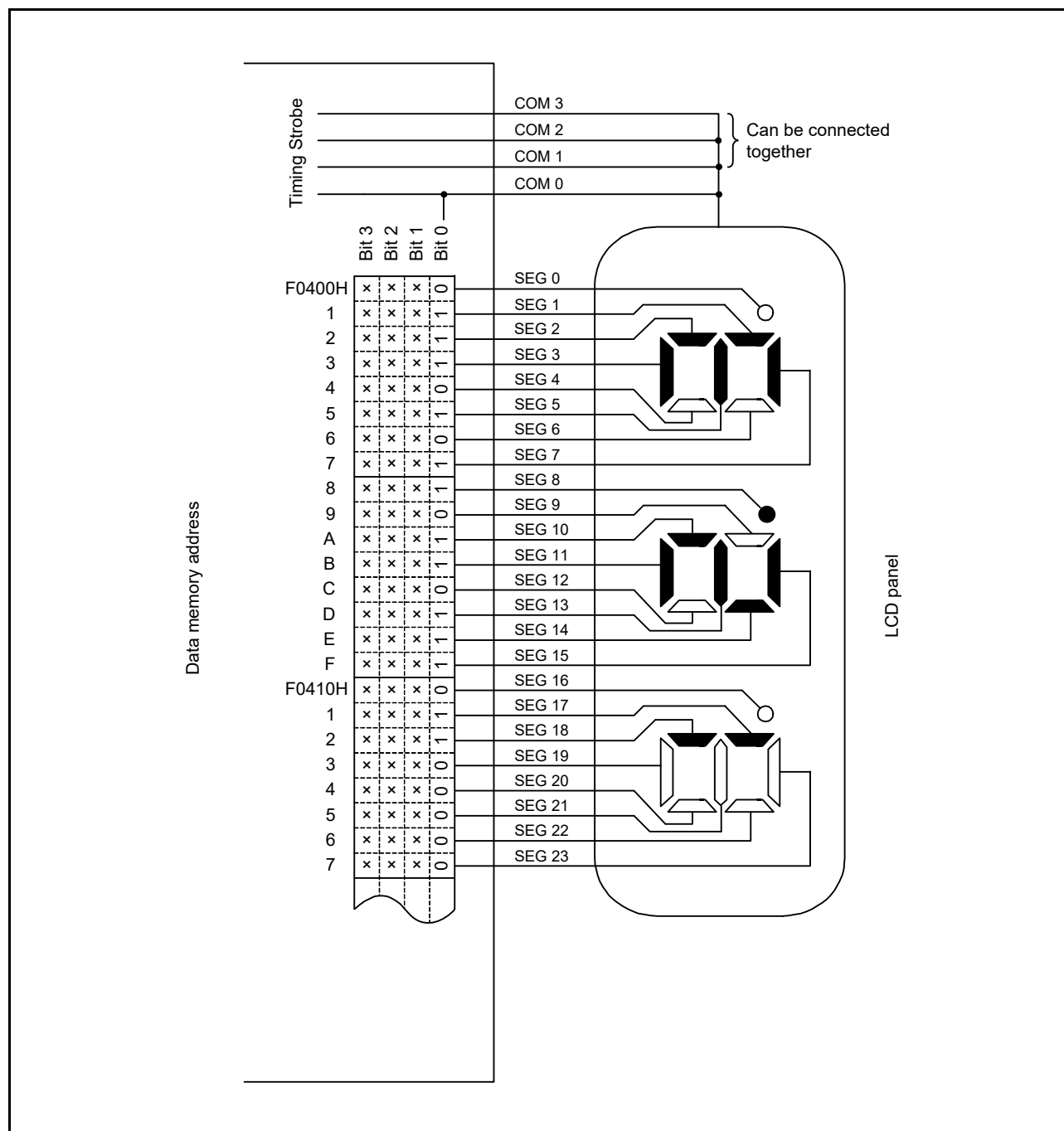
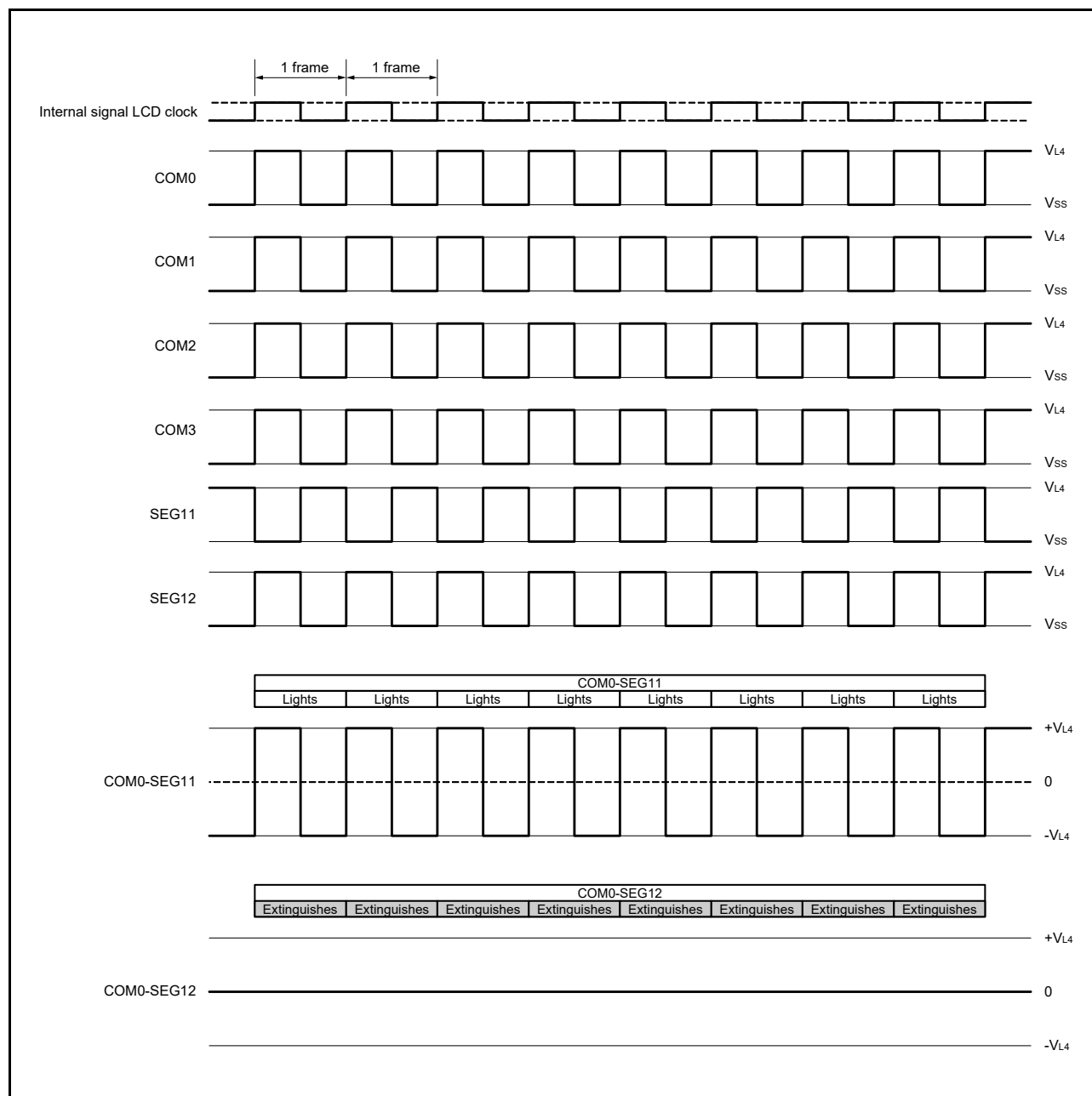


Figure 22 - 33 Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0



22.10.2 Two-time-slice display example

Figure 22 - 35 shows how the 6-digit LCD panel having the display pattern shown in **Figure 22 - 34** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 and COM1). This example displays data “12345.6” in the LCD panel. The contents of the display data register (F0400H to F0417H) correspond to this display.

The following description focuses on numeral “3” () displayed in the fourth digit. To display “3” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 to SEG15 pins according to **Table 22 - 20** at the timing of the common signals COM0 and COM1; see **Figure 22 - 34** for the relationship between the segment signals and LCD segments.

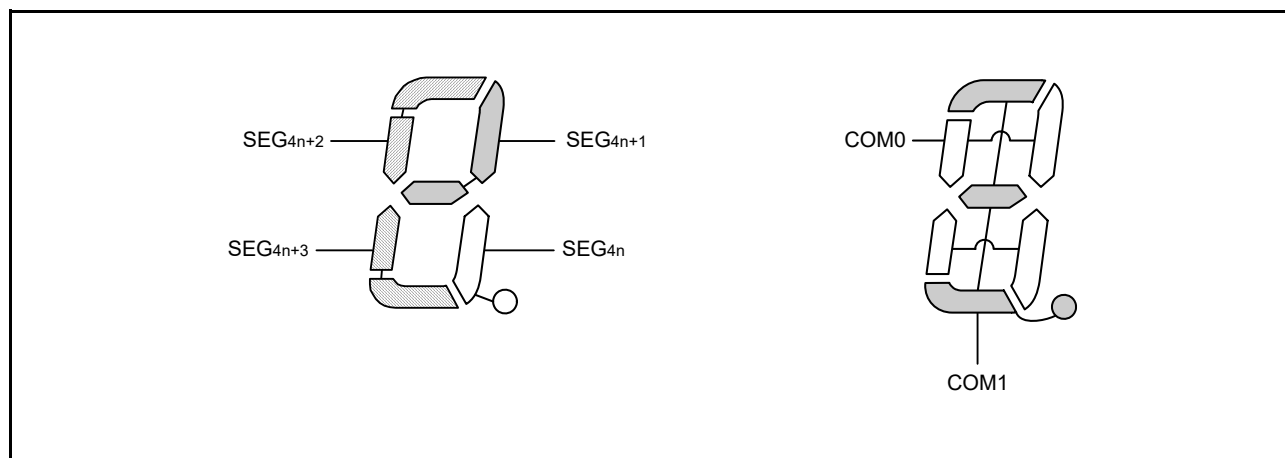
Table 22 - 20 Select and Deselect Voltages (COM0 and COM1)

Common \ Segment	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to the above, it is determined that the display data register location (F040FH) that corresponds to SEG15 must contain xx10.

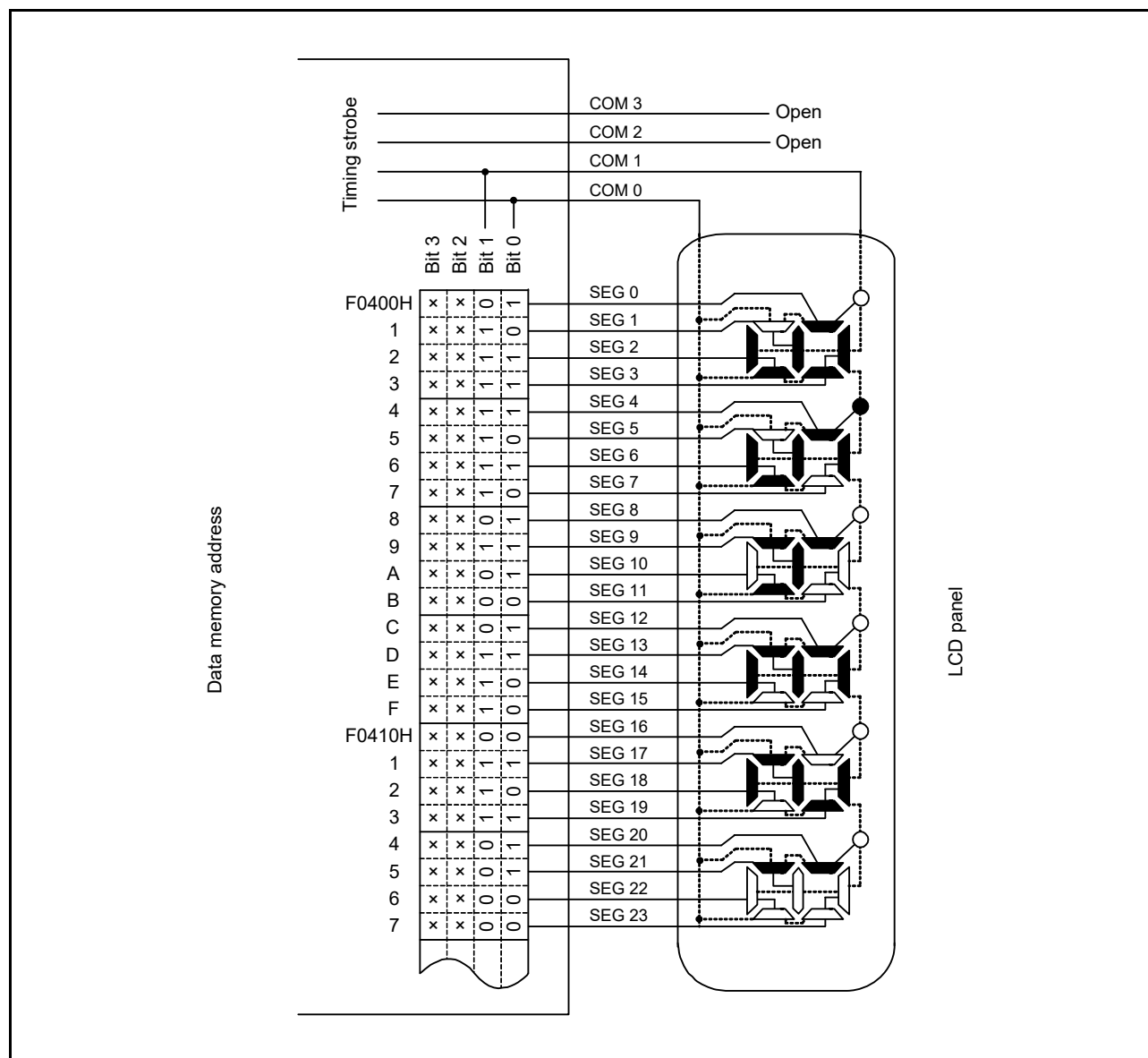
Figure 22 - 36 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +VLCD/–VLCD, is generated to turn on the corresponding LCD segment.

Figure 22 - 34 Two-Time-Slice LCD Display Pattern and Electrode Connections



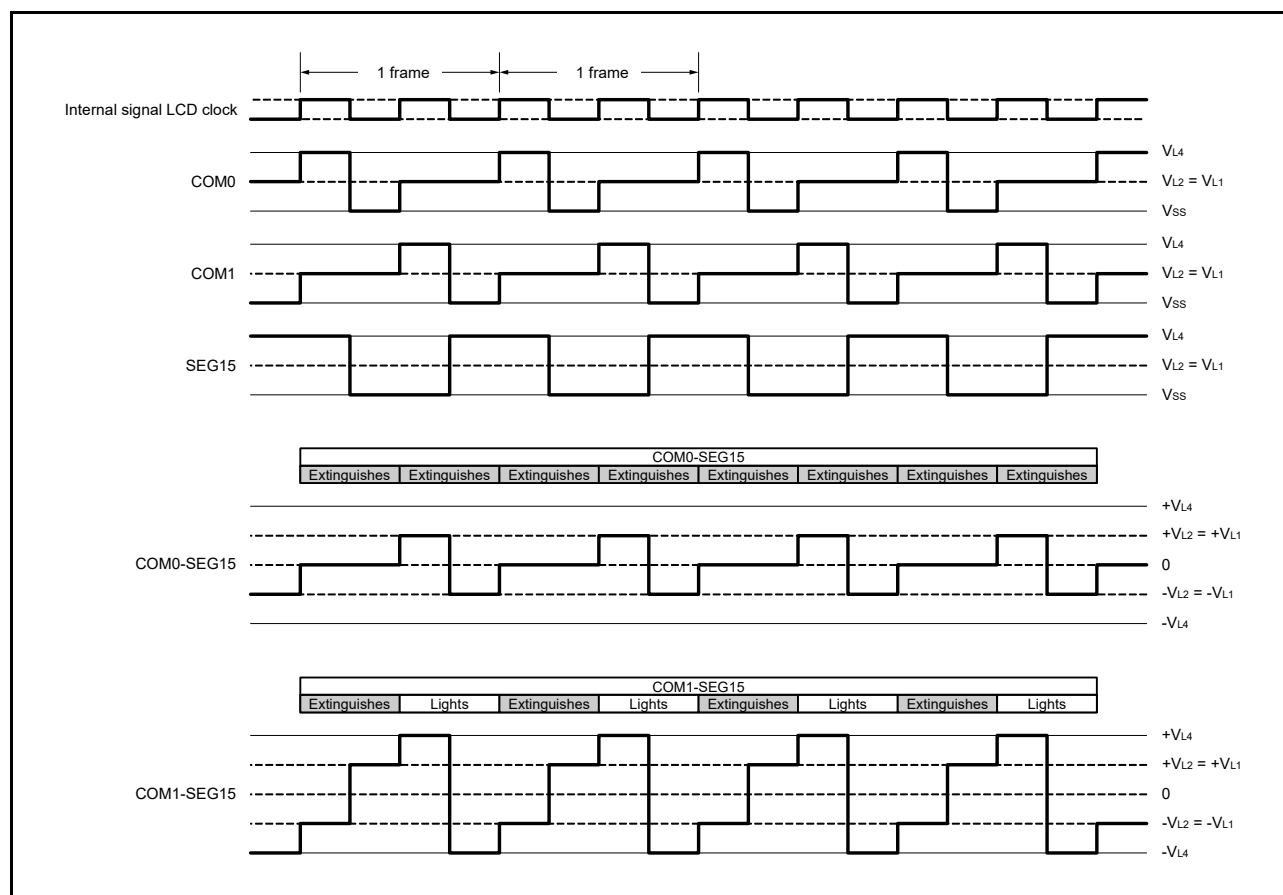
Remark 100-pin products: n = 0 to 13

Figure 22 - 35 Example of Connecting Two-Time-Slice LCD Panel



Remark ×: Can always be used to store any data because the two-time-slice mode is being used.

Figure 22 - 36 Two-Time-Slice LCD Drive Waveform Examples between SEG15 and Each Common Signals
(1/2 Bias Method)



22.10.3 Three-time-slice display example

Figure 22 - 38 shows how the 8-digit LCD panel having the display pattern shown in **Figure 22 - 37** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM2). This example displays data “123456.78” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display. The following description focuses on numeral “6.” (6.) displayed in the third digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG6 to SEG8 pins according to **Table 22 - 21** at the timing of the common signals COM0 to COM2; see **Figure 22 - 37** for the relationship between the segment signals and LCD segments.

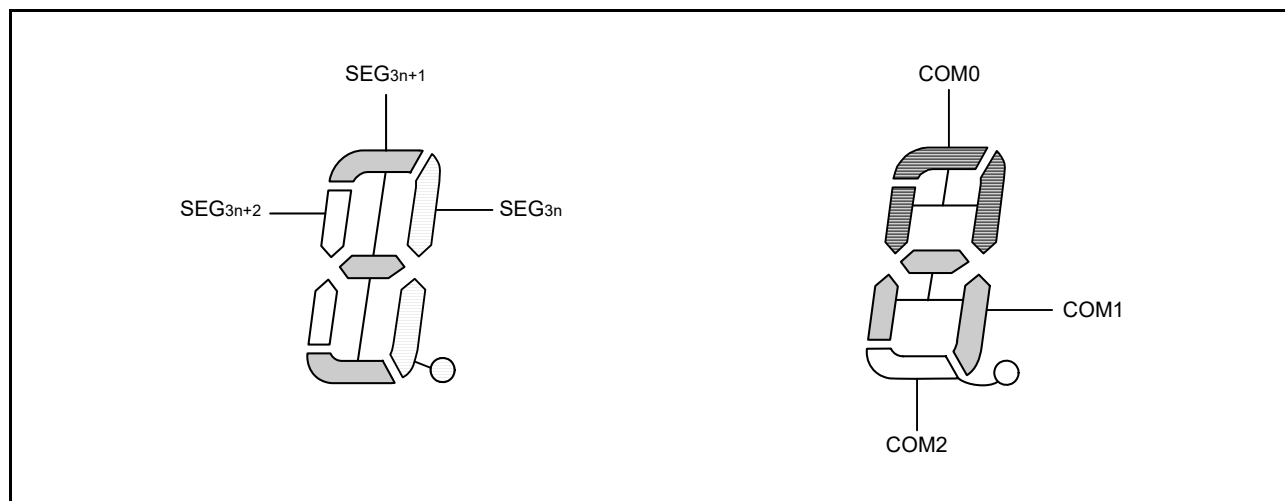
Table 22 - 21 Select and Deselect Voltages (COM0 to COM2)

Common \ Segment	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	—

According to the above, it is determined that the display data register location (F0406H) that corresponds to SEG6 must contain x110.

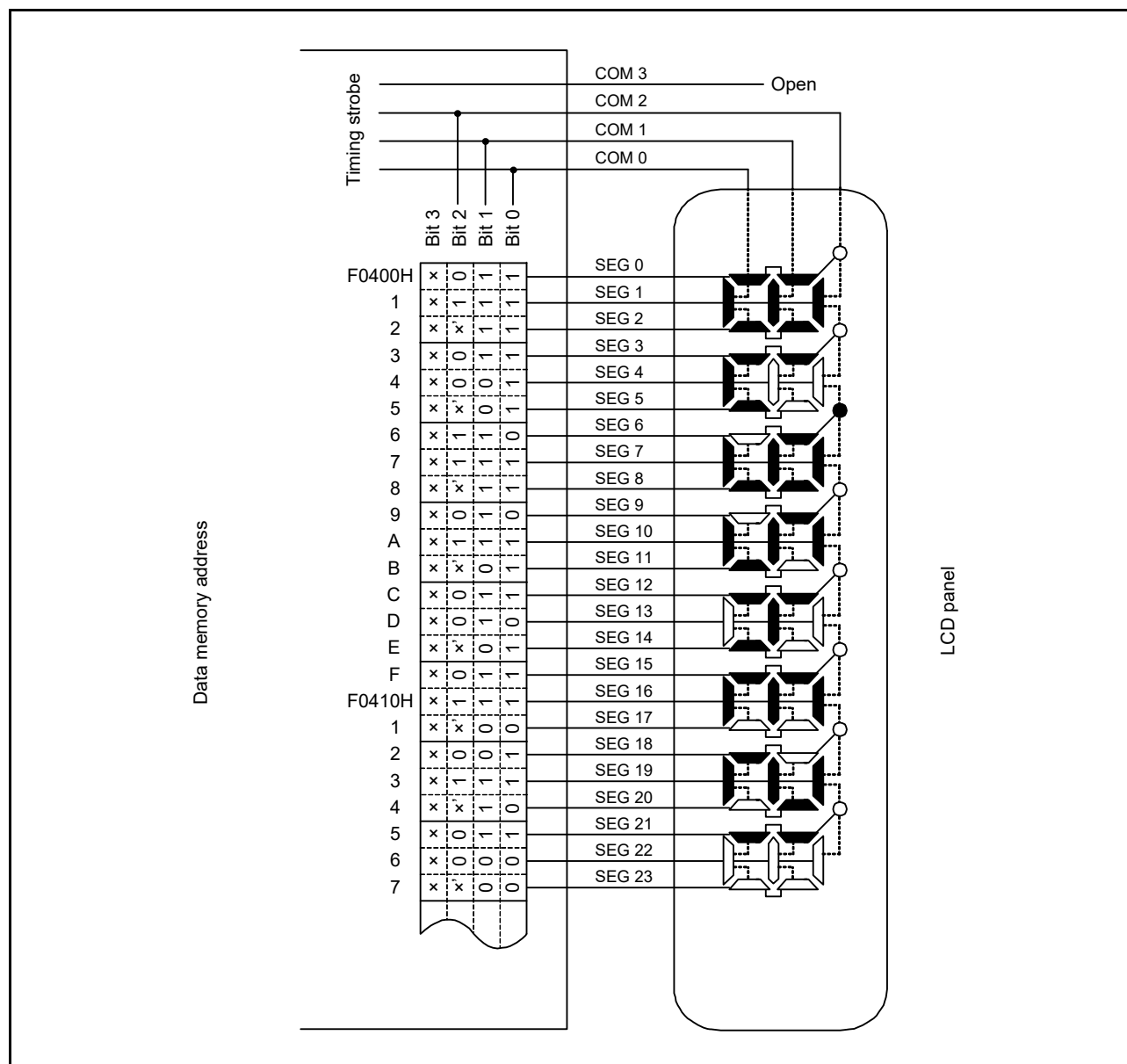
Figure 22 - 39 and **Figure 22 - 40** show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +VLCD/–VLCD, is generated to turn on the corresponding LCD segment.

Figure 22 - 37 Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin products: n = 0 to 17

Figure 22 - 38 Example of Connecting Three-Time-Slice LCD Panel



Remark *: Can be used to store any data because there is no corresponding segment in the LCD panel.
 *: Can always be used to store any data because the three-time-slice mode is being used.

Figure 22 - 39 Three-Time-Slice LCD Drive Waveform Examples between SEG6 and Each Common Signals
(1/2 Bias Method)

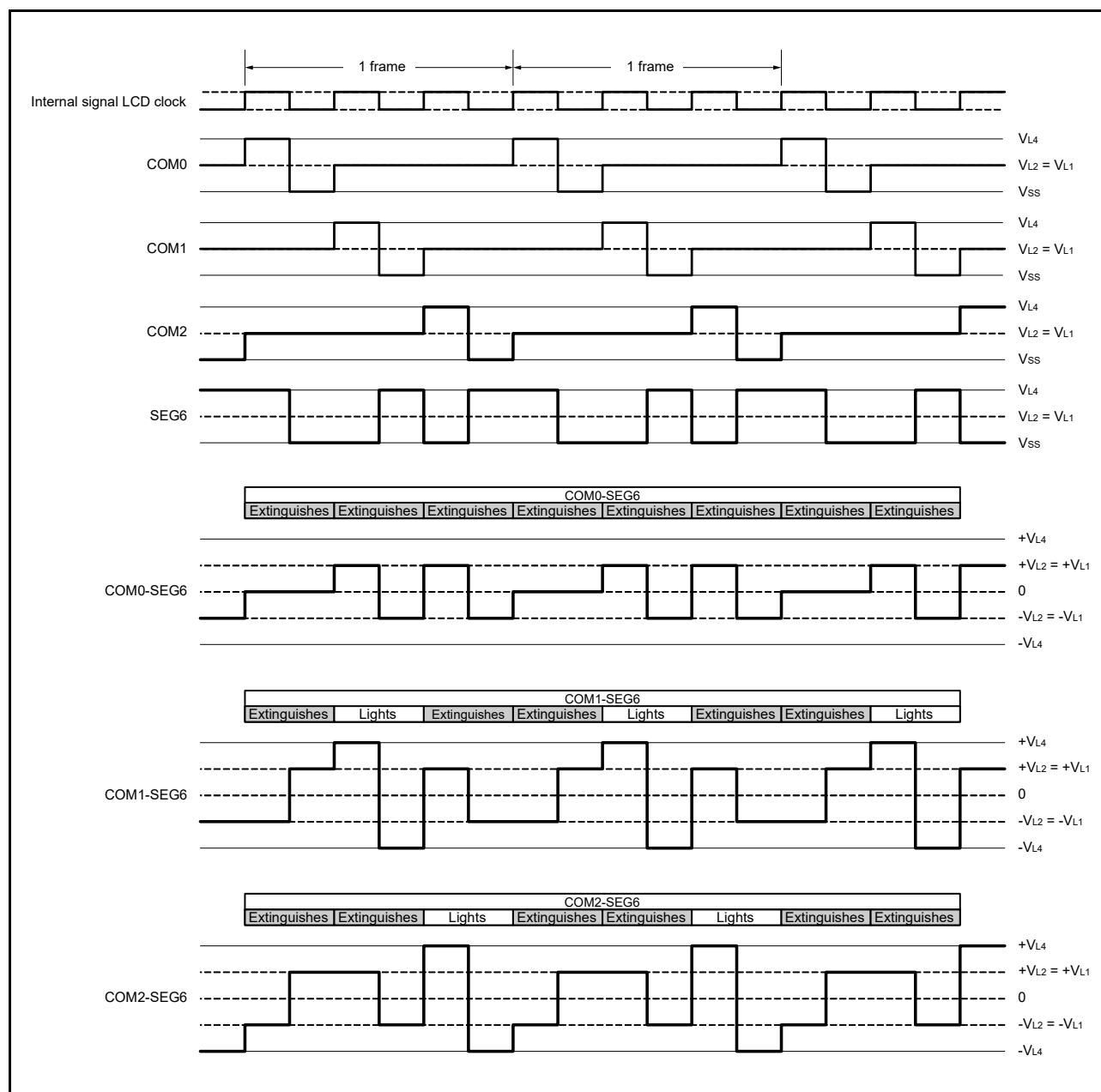
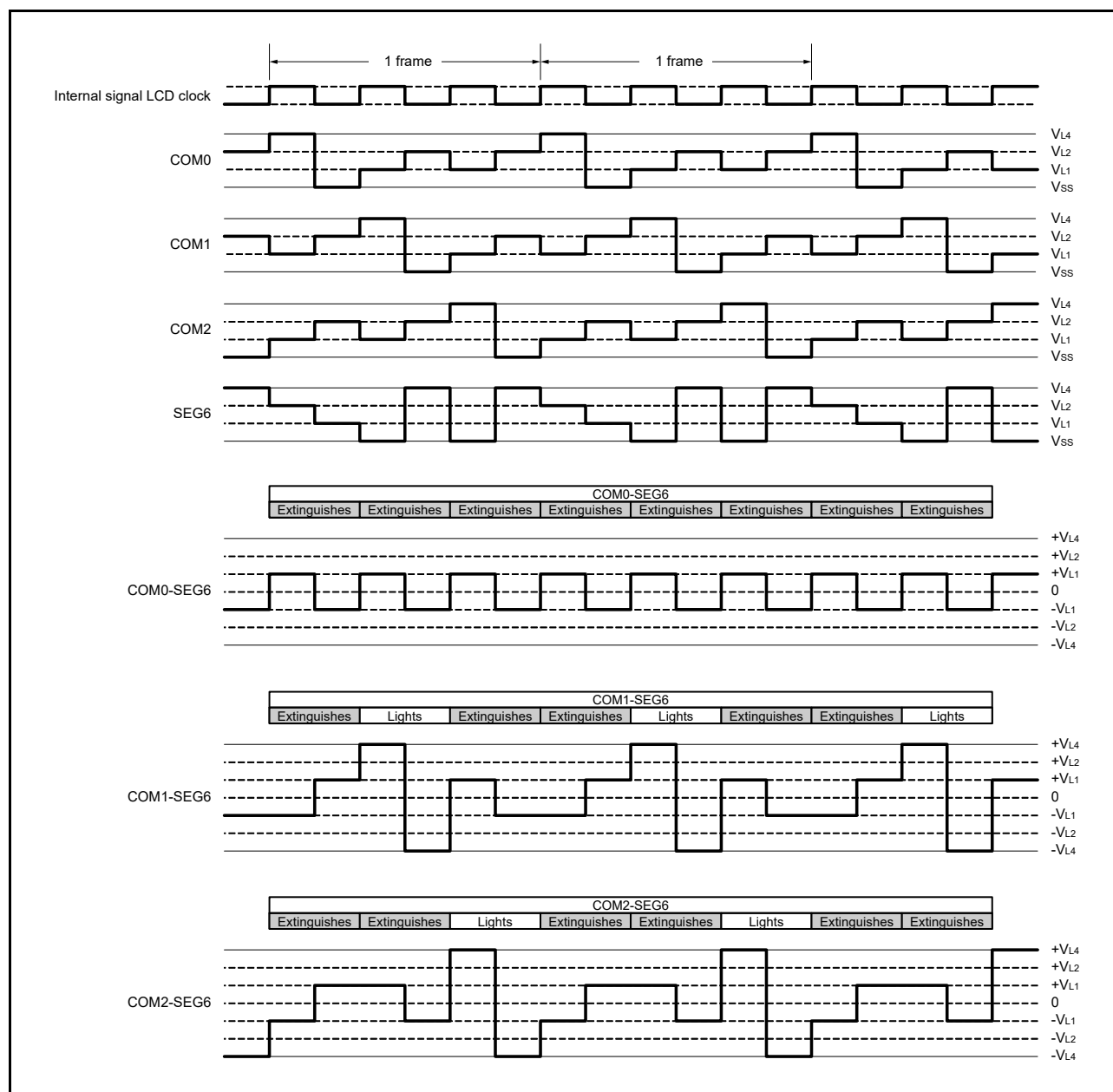


Figure 22 - 40 Three-Time-Slice LCD Drive Waveform Examples between SEG6 and Each Common Signals (1/3 Bias Method)



22.10.4 Four-time-slice display example

Figure 22 - 42 shows how the 12-digit LCD panel having the display pattern shown in **Figure 22 - 41** is connected to the segment signals (SEG0 to SEG23) and the common signals (COM0 to COM3). This example displays data “123456.789012” in the LCD panel. The contents of the display data register (addresses F0400H to F0417H) correspond to this display.

The following description focuses on numeral “6.” (6.) displayed in the seventh digit. To display “6.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG12 and SEG13 pins according to **Table 22 - 22** at the timing of the common signals COM0 to COM3; see **Figure 22 - 41** for the relationship between the segment signals and LCD segments.

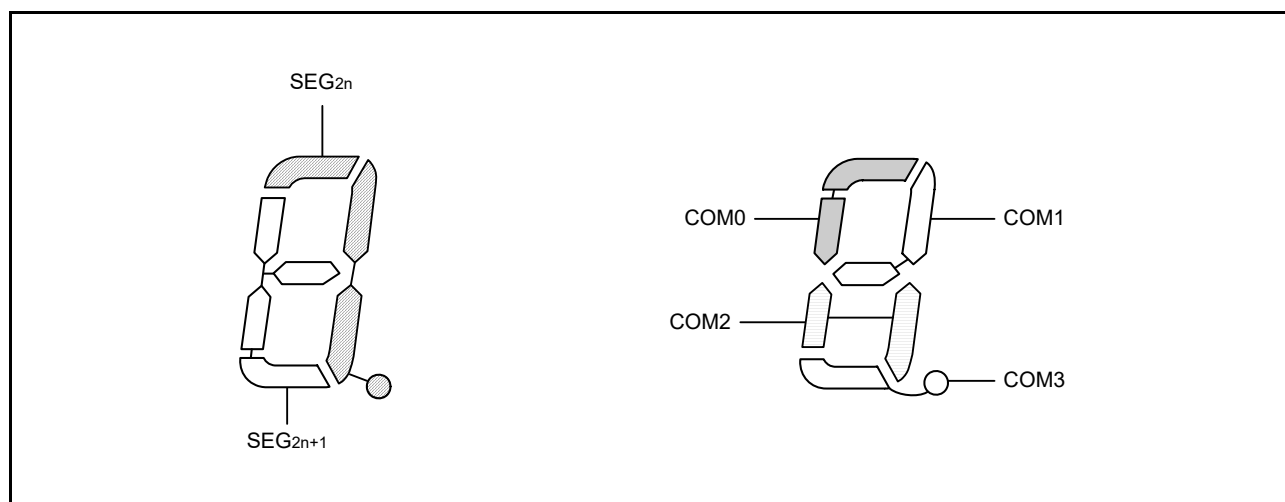
Table 22 - 22 Select and Deselect Voltages (COM0 to COM3)

Common \ Segment	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to the above, it is determined that the display data register location (F040CH) that corresponds to SEG12 must contain 1101.

Figure 22 - 43 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, +VLCD/–VLCD, is generated to turn on the corresponding LCD segment.

Figure 22 - 41 Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin products: n = 0 to 27

Figure 22 - 42 Example of Connecting Four-Time-Slice LCD Panel

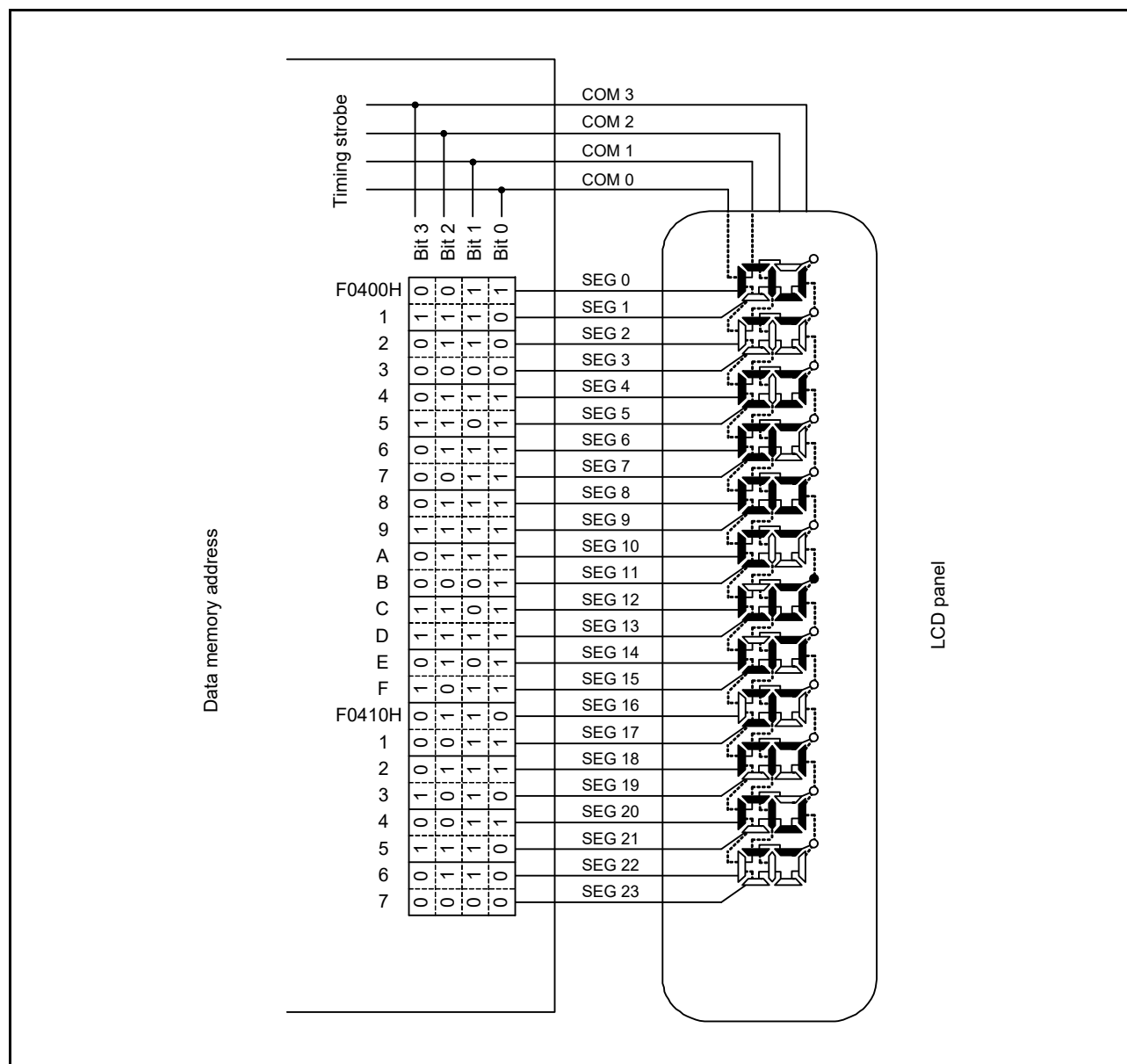


Figure 22 - 43 Four-Time-Slice LCD Drive Waveform Examples between SEG12 and Each Common Signals (1/3 Bias Method) (1/2)

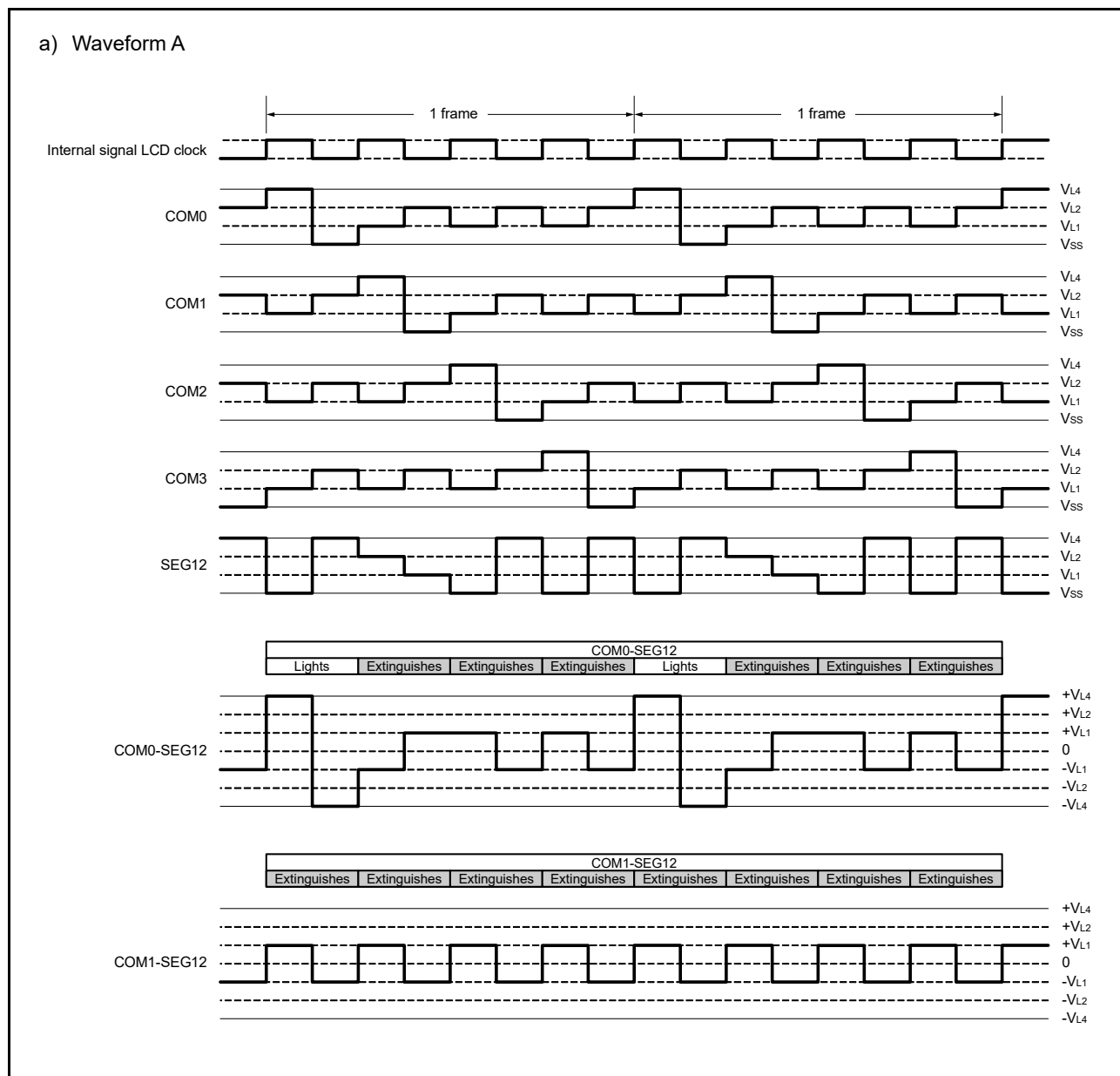
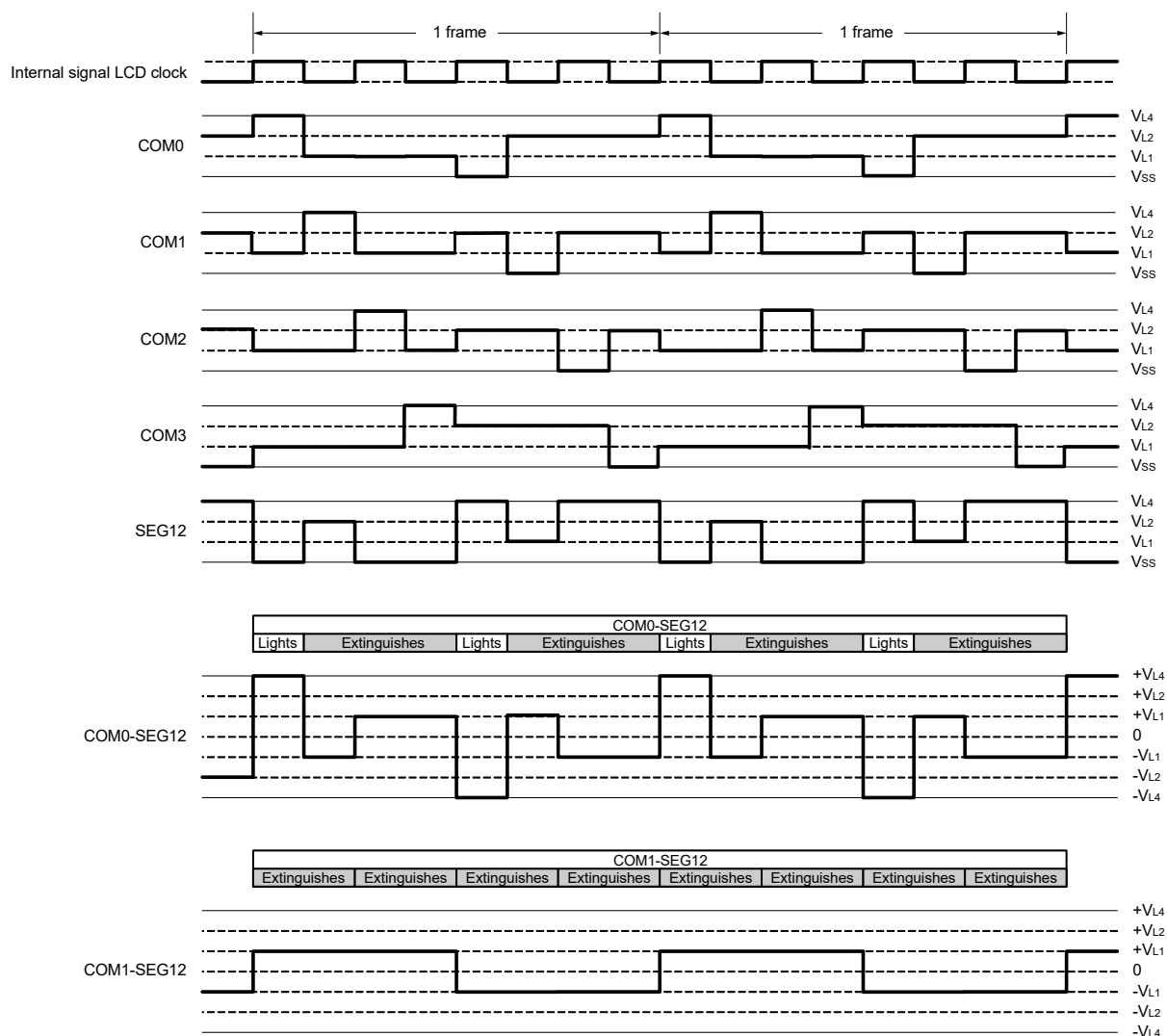


Figure 22 - 43 Four-Time-Slice LCD Drive Waveform Examples between SEG12 and Each Common Signals
(1/3 Bias Method) (2/2)

b) Waveform B



22.10.5 Six-time-slice display example

Figure 22 - 45 shows how the 15x6 dot LCD panel having the display pattern shown in **Figure 22 - 44** is connected to the segment signals (SEG2 to SEG16) and the common signals (COM0 to COM5). This example displays data “123” in the LCD panel. The contents of the display data register (addresses F0402H to F0410H) correspond to this display. The following description focuses on numeral “3” (3) displayed in the first digit. To display “3.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG2 to SEG6 pins according to **Table 22 - 23** at the timing of the common signals COM0 to COM5; see **Figure 22 - 44** for the relationship between the segment signals and LCD segments.

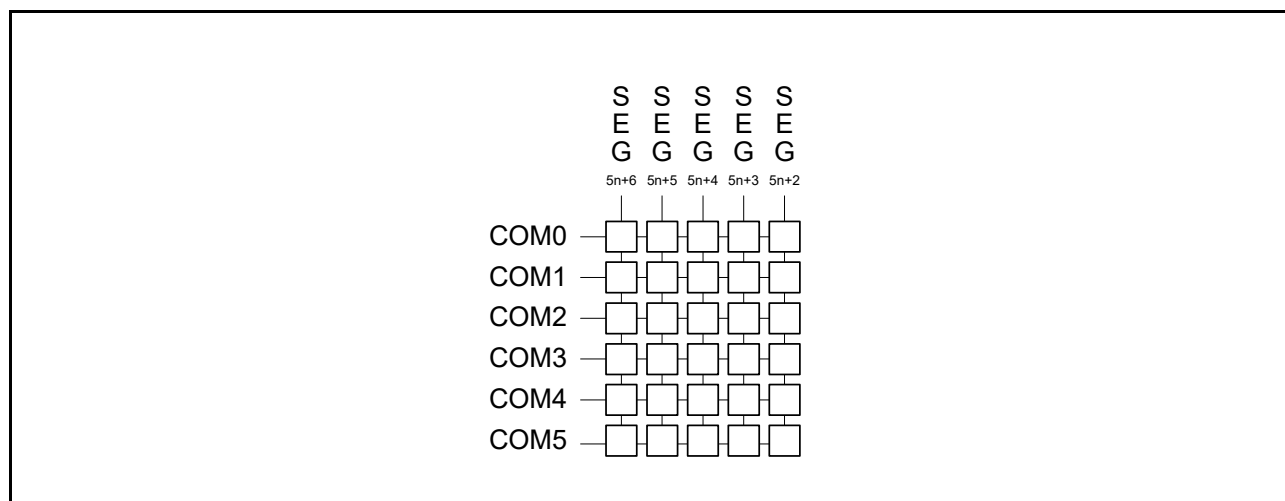
Table 22 - 23 Select and Deselect Voltages (COM0 to COM5)

Common \ Segment	SEG2	SEG3	SEG4	SEG5	SEG6
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Select
COM5	Deselect	Select	Select	Select	Deselect

According to the above, it is determined that the display data register location (F0402H) that corresponds to SEG2 must contain 010001.

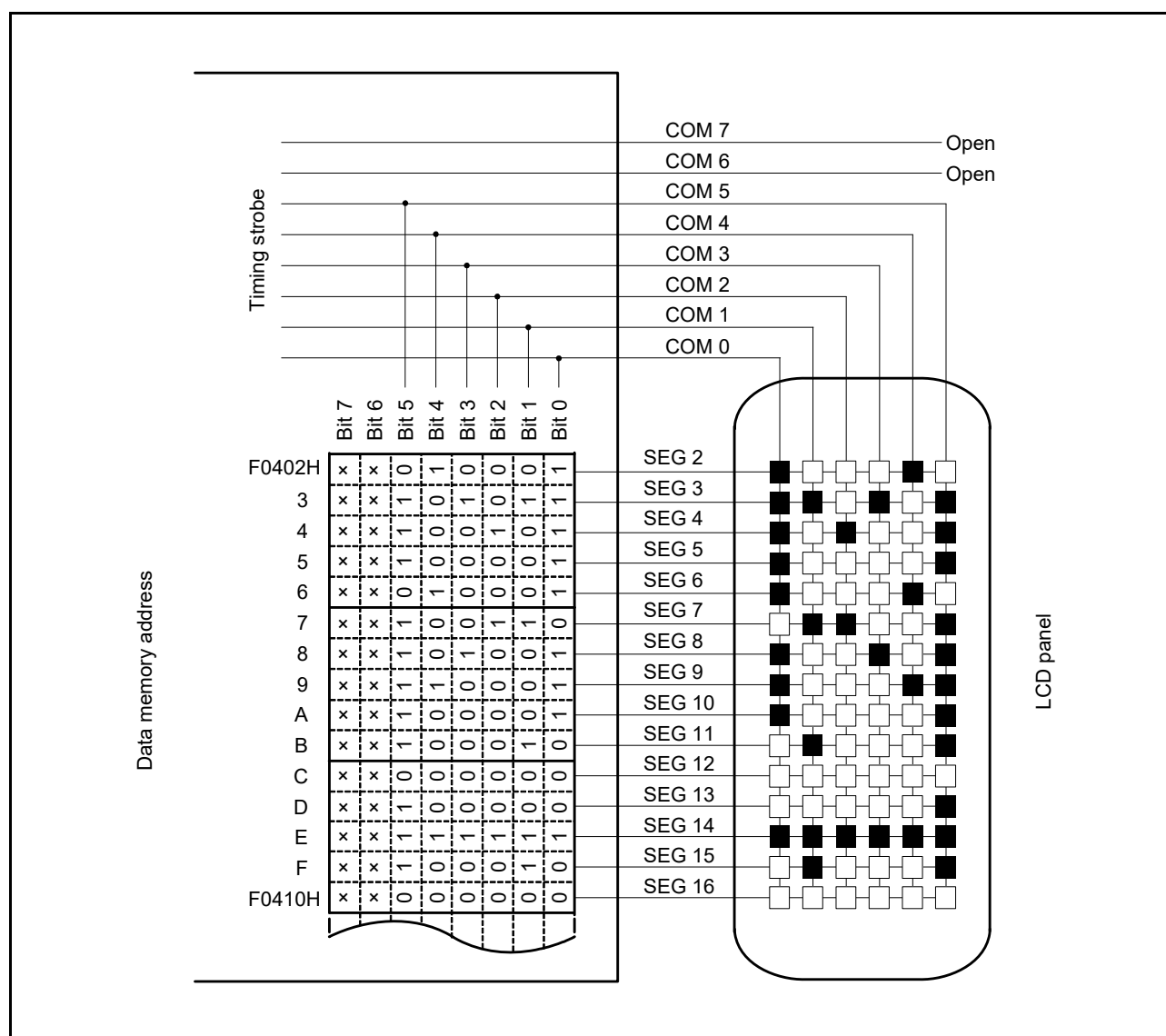
Figure 22 - 46 and **Figure 22 - 47** show examples of LCD drive waveforms between the SEG2 signal and each common signal, respectively. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 22 - 44 Six-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin products: n = 0 to 9

Figure 22 - 45 Example of Connecting Six-Time-Slice LCD Panel



a) Waveform A

The diagram illustrates the timing of various signals for Waveform A over one frame. The signals are:

- Internal signal LCD clock**: A periodic clock signal.
- COM0**: A signal that transitions between V_{L4} , V_{L3} , V_{L2} , V_{L1} , and V_{SS} .
- COM1**: A signal that transitions between V_{L4} , V_{L3} , V_{L2} , V_{L1} , and V_{SS} .
- COM2**: A signal that transitions between V_{L4} , V_{L3} , V_{L2} , V_{L1} , and V_{SS} .
- COM5**: A signal that transitions between V_{L4} , V_{L3} , V_{L2} , V_{L1} , and V_{SS} .
- SEG2**: A signal that transitions between V_{L4} , V_{L3} , V_{L2} , V_{L1} , and V_{SS} .
- COM0-SEG2**: A signal that transitions between $+V_{L4}$, $+V_{L3}$, $+V_{L2}$, $+V_{L1}$, 0 , $-V_{L1}$, $-V_{L2}$, $-V_{L3}$, and $-V_{L4}$.
- COM1-SEG2**: A signal that transitions between $+V_{L4}$, $+V_{L3}$, $+V_{L2}$, $+V_{L1}$, 0 , $-V_{L1}$, $-V_{L2}$, $-V_{L3}$, and $-V_{L4}$.

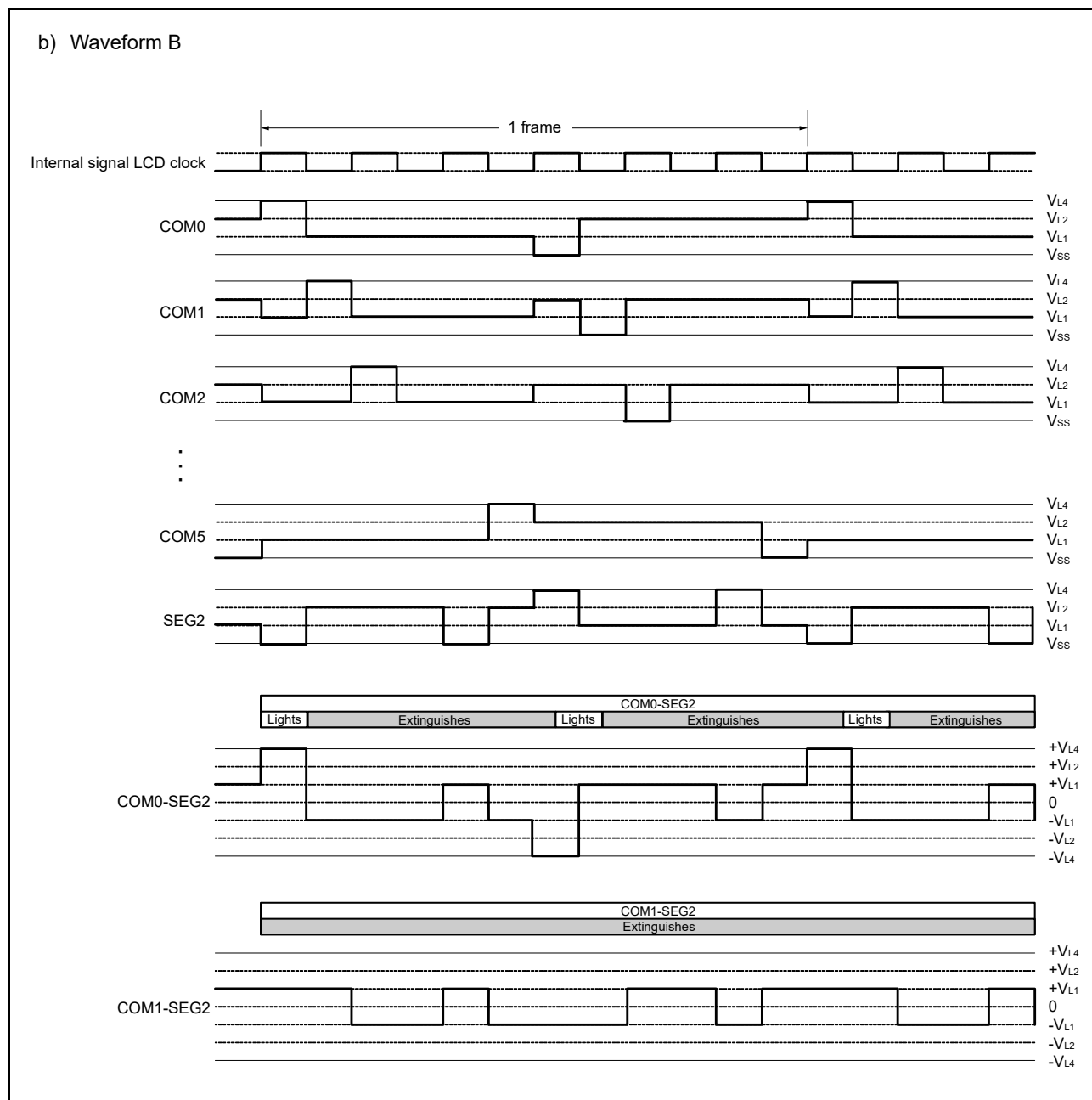
The diagram also includes a legend for the signal levels:

- V_{L4}
- V_{L3}
- V_{L2}
- V_{L1}
- V_{SS}

For the COM0-SEG2 and COM1-SEG2 signals, the legend includes:

- $+V_{L4}$
- $+V_{L3}$
- $+V_{L2}$
- $+V_{L1}$
- 0
- $-V_{L1}$
- $-V_{L2}$
- $-V_{L3}$
- $-V_{L4}$

Figure 22 - 47 Six-Time-Slice LCD Drive Waveform Examples between SEG2 and Each Common Signals
(1/3 Bias Method)



22.10.6 Eight-time-slice display example

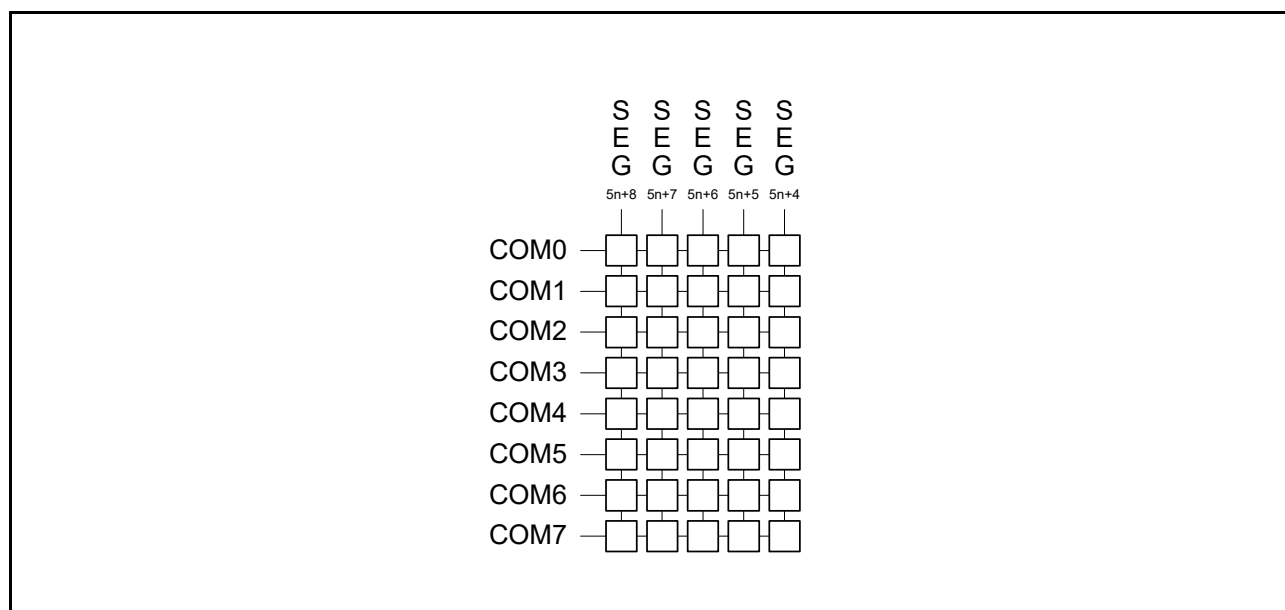
Figure 22 - 49 shows how the 15x8 dot LCD panel having the display pattern shown in **Figure 22 - 48** is connected to the segment signals (SEG4 to SEG18) and the common signals (COM0 to COM7). This example displays data “123” in the LCD panel. The contents of the display data register (addresses F0404H to F0412H) correspond to this display. The following description focuses on numeral “3.” (**3**) displayed in the first digit. To display “3.” in the LCD panel, it is necessary to apply the select or deselect voltage to the SEG4 to SEG8 pins according to **Table 22 - 24** at the timing of the common signals COM0 to COM7; see **Figure 22 - 48** for the relationship between the segment signals and LCD segments.

Table 22 - 24 Select and Deselect Voltages (COM0 to COM7)

Common \ Segment	SEG4	SEG5	SEG6	SEG7	SEG8
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Deselect
COM5	Select	Deselect	Deselect	Deselect	Select
COM6	Deselect	Select	Select	Select	Deselect
COM7	Deselect	Deselect	Deselect	Deselect	Deselect

According to the above, it is determined that the display data register location (F0404H) that corresponds to SEG4 must contain 00110001. **Figure 22 - 50** shows examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 22 - 48 Eight-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin products: n = 0 to 9

Figure 22 - 49 Example of Connecting Eight-Time-Slice LCD Panel

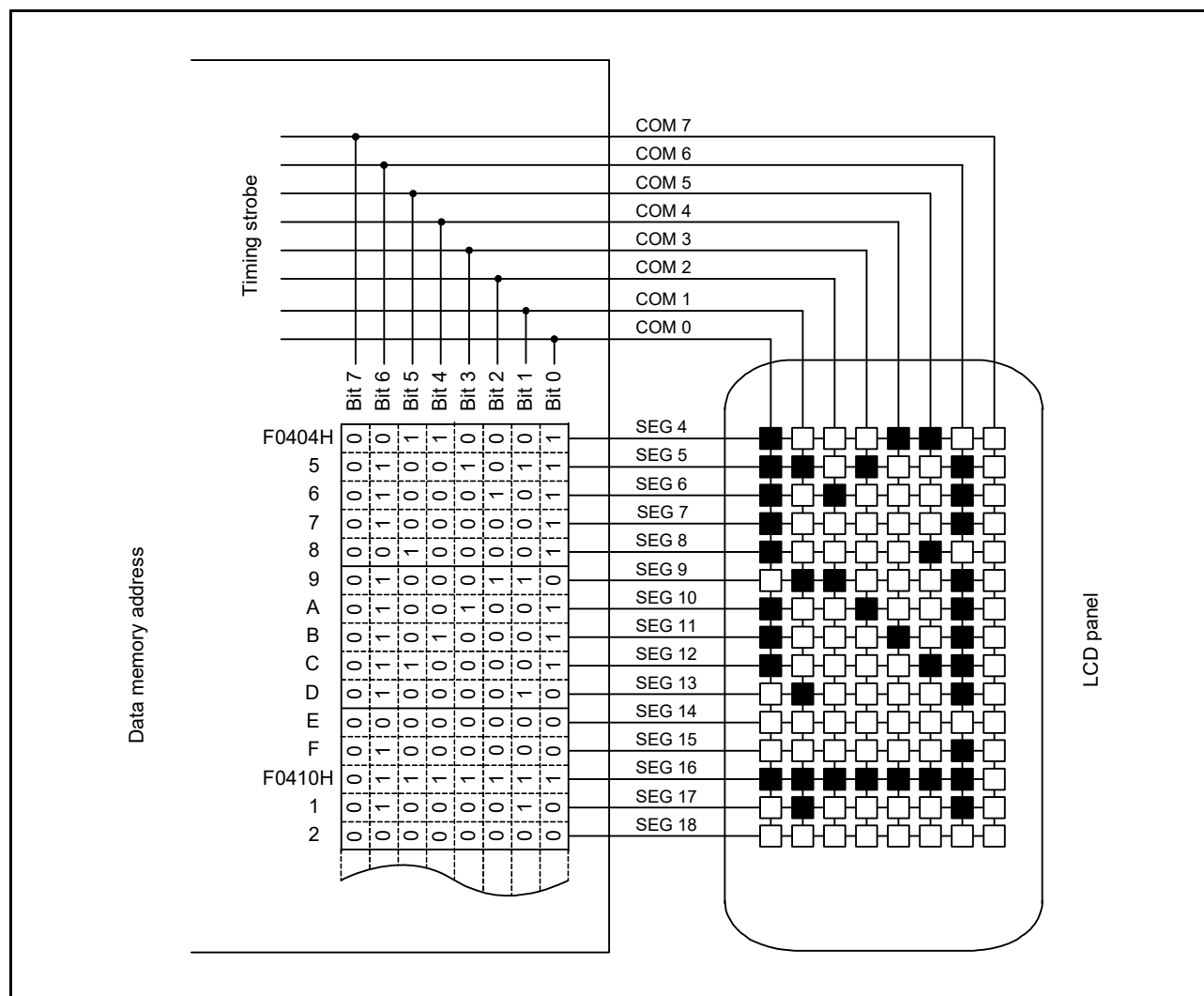


Figure 22 - 50 Eight-Time-Slice LCD Drive Waveform Examples between SEG4 and Each Common Signals
(1/4 Bias Method) (1/2)

a) Waveform A

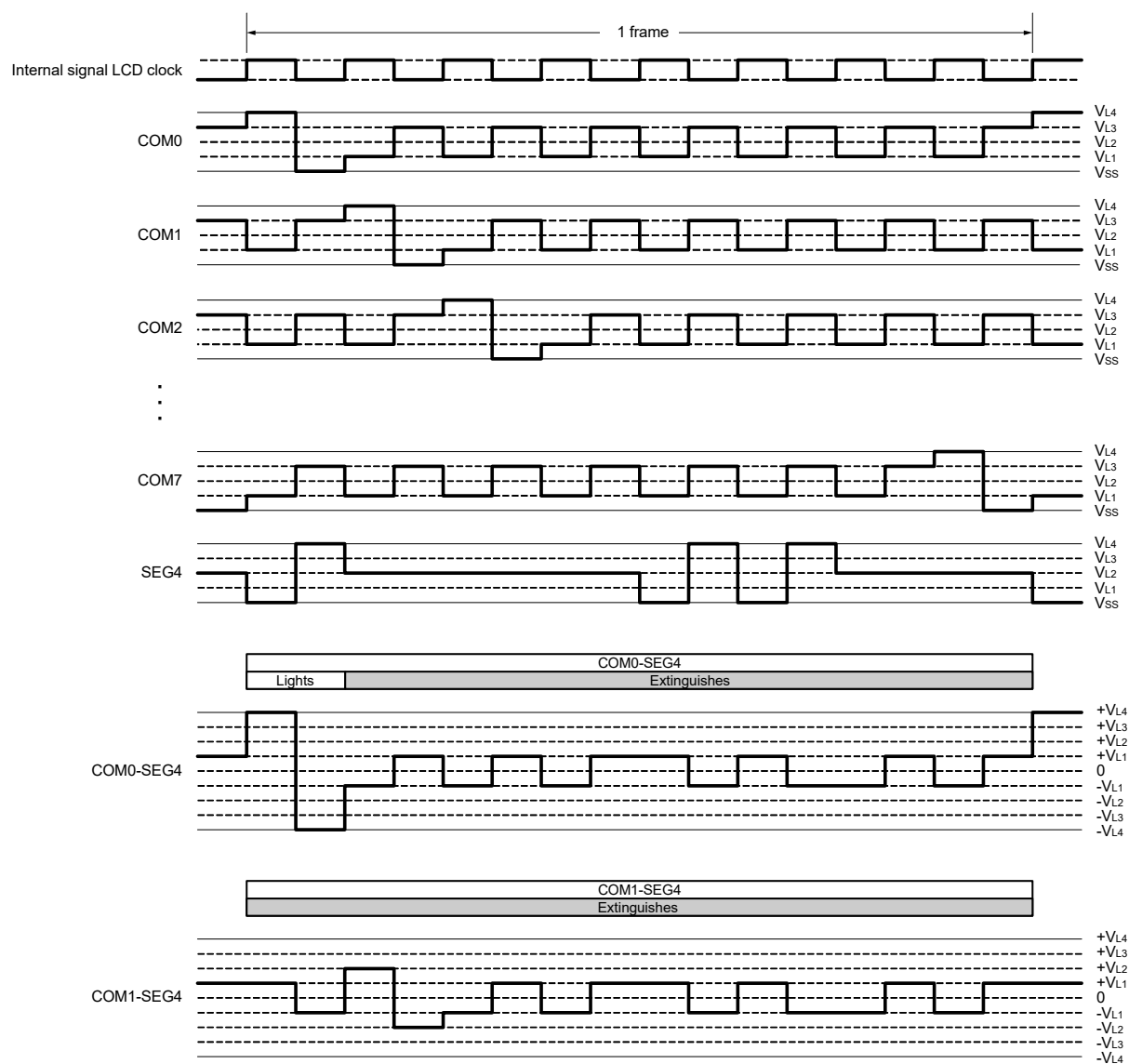
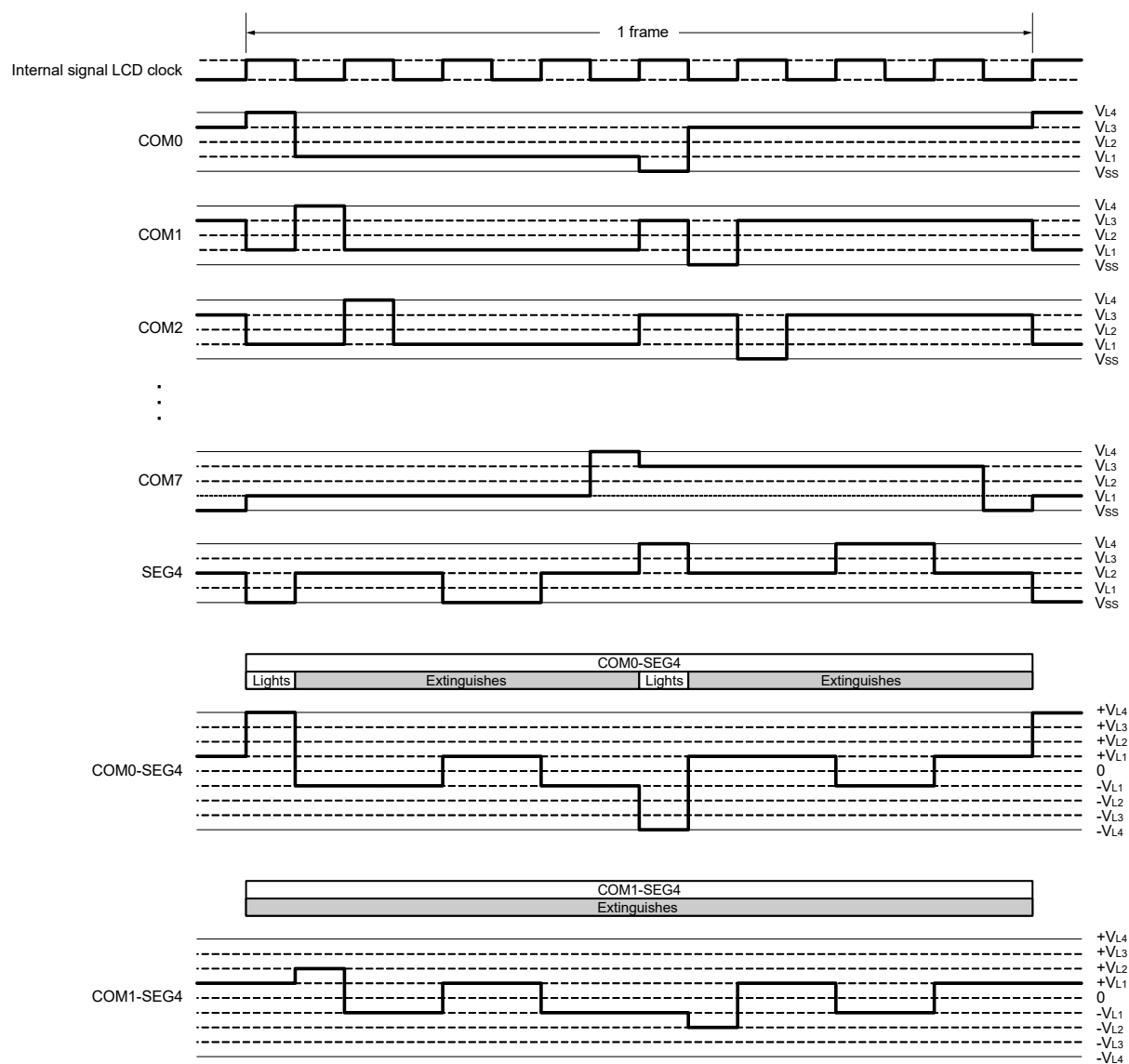


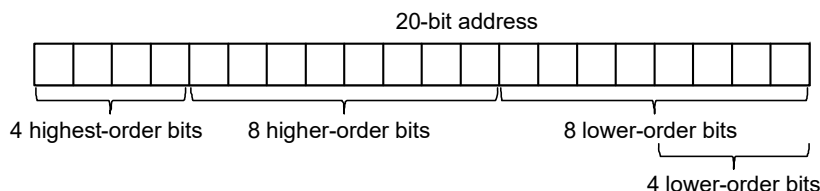
Figure 22 - 50 Eight-Time-Slice LCD Drive Waveform Examples between SEG4 and Each Common Signals
(1/4 Bias Method) (2/2)

b) Waveform B



Section 23 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).

23.1 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 23 - 1** lists the DTC specifications.

Table 23 - 1 DTC Specifications (1/2)

Item		Specification
Activation sources		48 sources (44- and 48-pin products with 64 to 128 Kbytes of flash memory) 49 sources (52- and 64-pin products with 64 to 128 Kbytes of flash memory) 51 sources (44-pin products with 256 to 512 Kbytes of flash memory) 53 sources (48-pin products with 256 to 512 Kbytes of flash memory) 54 sources (52- and 64-pin products with 256 to 512 Kbytes of flash memory) 56 sources (80- and 100-pin products with 128 to 512 Kbytes of flash memory)
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) Note 1 , mirror area Note 2 , data flash memory area Note 2 , 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 DTRLDj 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 23 - 3 DTC Activation Sources and Vector Addresses .

Table 23 - 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 DTCENi7 to DTCENi0 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 DTCENi7 to DTCENi0 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 DTCENi7 to DTCENi0 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 1. If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.

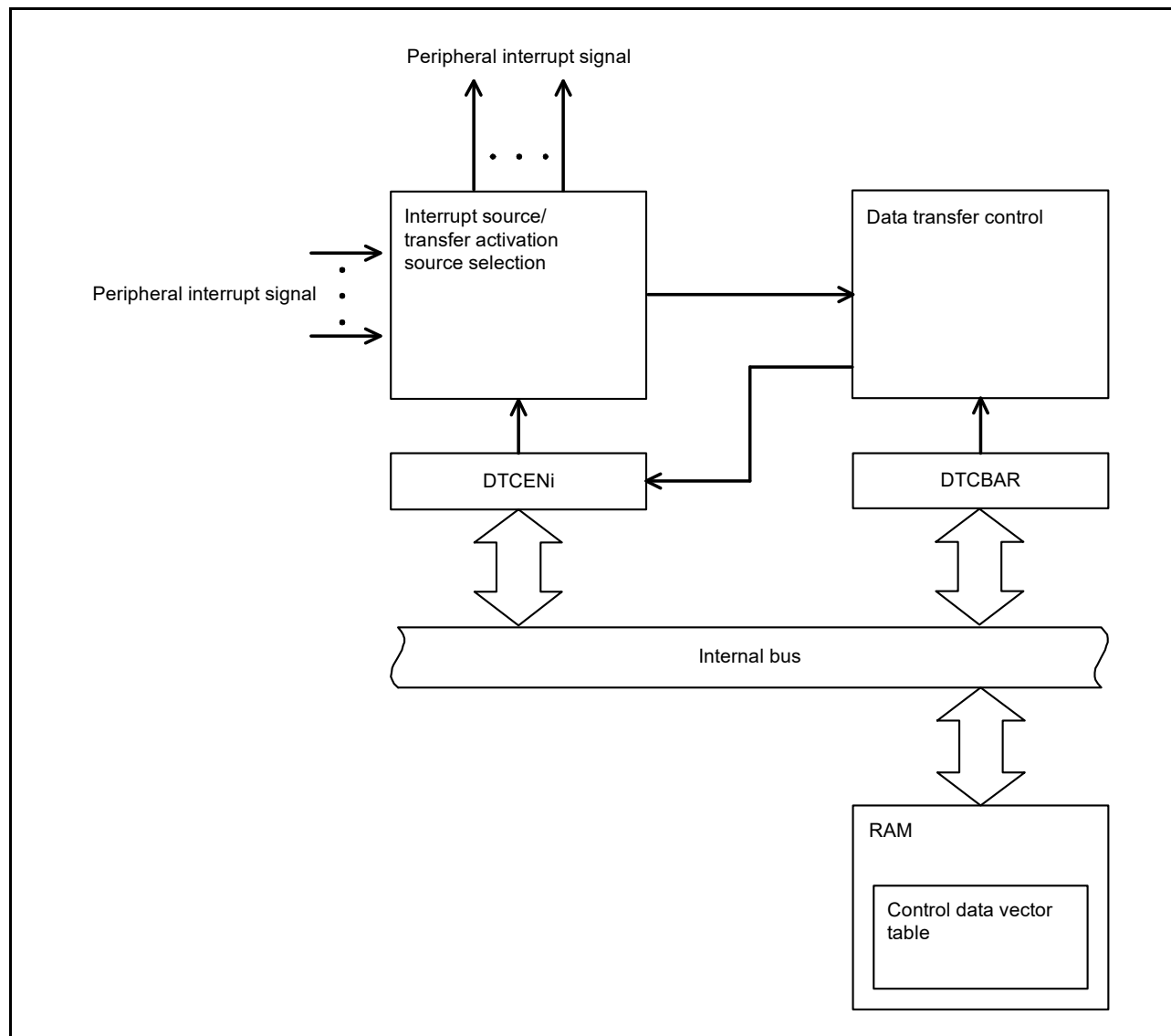
Note 2. In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped. If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.

Remark i = 0 to 6, j = 0 to 23

23.2 Configuration of DTC

Figure 23 - 1 shows the DTC block diagram.

Figure 23 - 1 DTC Block Diagram



Remark $i = 0 \text{ to } 6$

23.3 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 6)
- DTC base address register (DTCBAR)

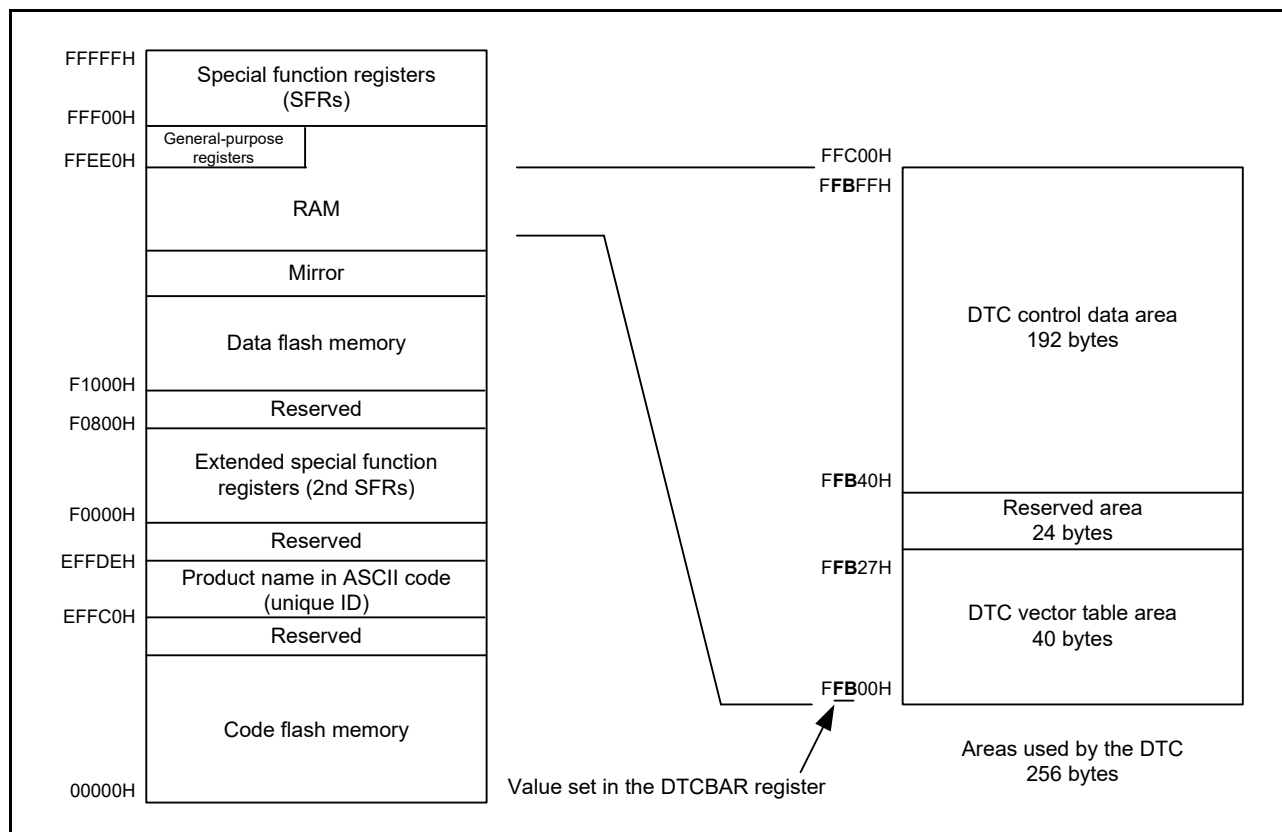
The DTC control data are listed below. DTC control data is allocated in the DTC control data area in RAM. The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

- 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)

23.3.1 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 23 - 2** shows the example of the memory map 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 23 - 2 Example of the Memory Map When the DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the product.

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.

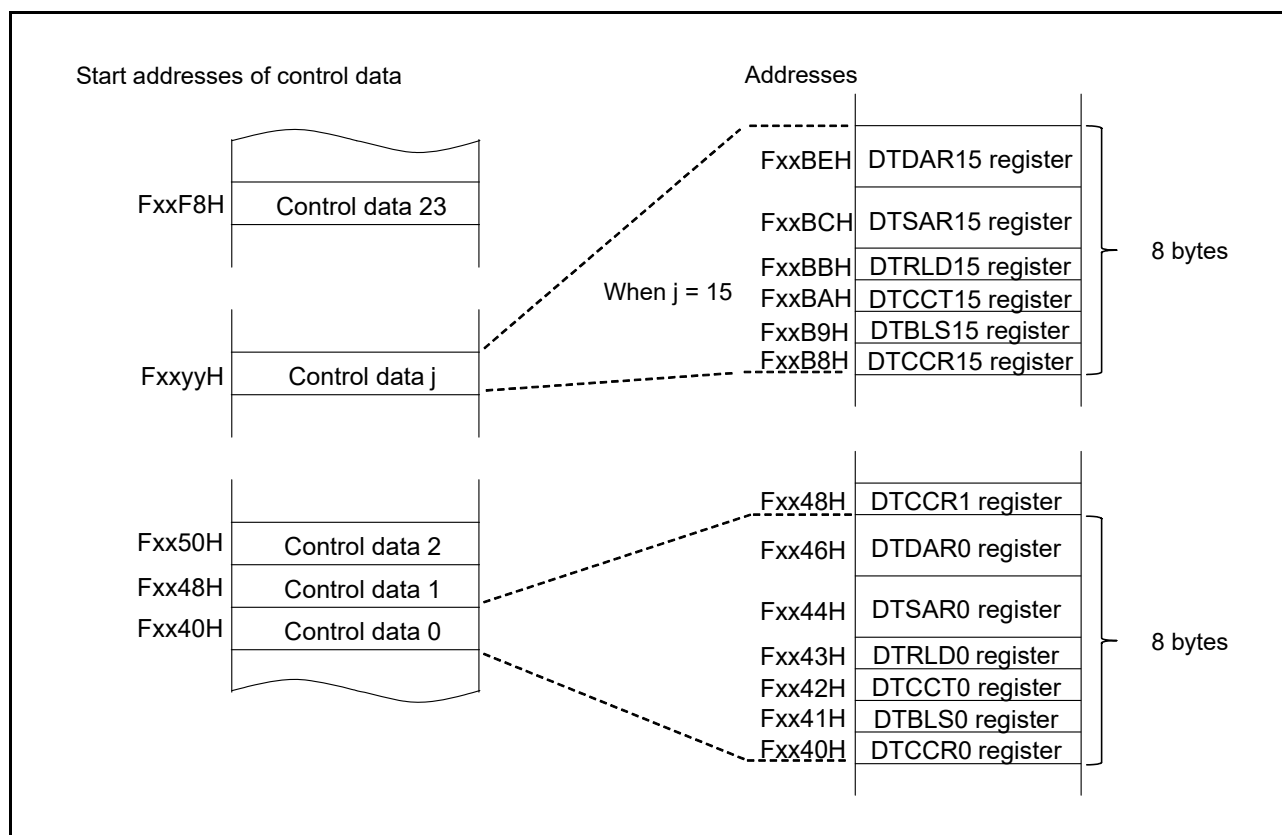
23.3.2 Control data allocation

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 23 - 3** shows the control data allocation.

Caution 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi7 to DTCENi0 (i = 0 to 6) 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 23 - 3 Control Data Allocation



Remark xx: Value set in the DTCBAR register

Table 23 - 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

23.3.3 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 23 - 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 DTCENi7 to DTCENi0 (i = 0 to 6) in the DTCENi register is set to 0 (activation disabled).

Figure 23 - 4 Start Addresses of Control Data and Vector Table

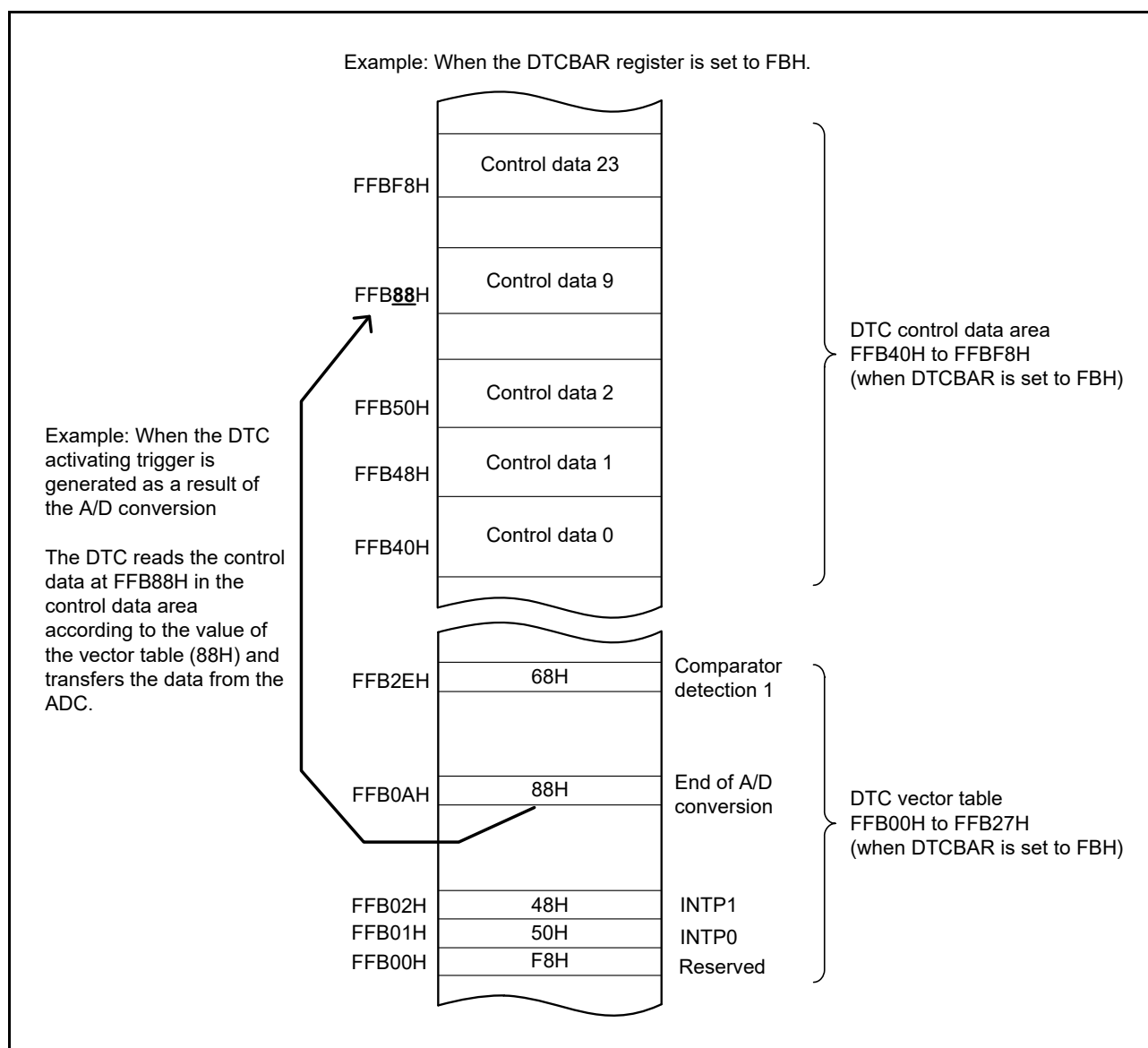
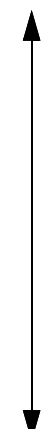


Table 23 - 3 DTC Activation Sources and Vector Addresses (1/2)

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in the DTCBAR register + 00H	 Highest
INTP0	1	Address set in the DTCBAR register + 01H	
INTP1	2	Address set in the DTCBAR register + 02H	
INTP2	3	Address set in the DTCBAR register + 03H	
INTP3	4	Address set in the DTCBAR register + 04H	
INTP4	5	Address set in the DTCBAR register + 05H	
INTP5	6	Address set in the DTCBAR register + 06H	
INTP6	7	Address set in the DTCBAR register + 07H	
INTP7	8	Address set in the DTCBAR register + 08H	
Key input	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/CSI10 transfer end or buffer empty/IIC10 transfer end	14	Address set in the DTCBAR register + 0EH	
UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end	15	Address set in the DTCBAR register + 0FH	
UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	16	Address set in the DTCBAR register + 10H	
UARTA0 reception transfer end	17	Address set in the DTCBAR register + 11H	
UARTA0 transmission transfer end/buffer empty	18	Address set in the DTCBAR register + 12H	
UARTA1 reception transfer end	19	Address set in the DTCBAR register + 13H	
UARTA1 transmission transfer end/buffer empty	20	Address set in the DTCBAR register + 14H	
UARTA2 reception transfer end ^{Note 1}	21	Address set in the DTCBAR register + 15H	
UARTA2 transmission transfer end/buffer empty ^{Note 1}	22	Address set in the DTCBAR register + 16H	
UARTA3 reception transfer end ^{Note 2}	23	Address set in the DTCBAR register + 17H	
UARTA3 transmission transfer end/buffer empty ^{Note 2}	24	Address set in the DTCBAR register + 18H	
End of channel 0 of timer array unit 0 count or capture	25	Address set in the DTCBAR register + 19H	
End of channel 1 of timer array unit 0 count or capture	26	Address set in the DTCBAR register + 1AH	
End of channel 2 of timer array unit 0 count or capture	27	Address set in the DTCBAR register + 1BH	
End of channel 3 of timer array unit 0 count or capture	28	Address set in the DTCBAR register + 1CH	
End of channel 4 of timer array unit 0 count or capture	29	Address set in the DTCBAR register + 1DH	
End of channel 5 of timer array unit 0 count or capture	30	Address set in the DTCBAR register + 1EH	
End of channel 6 of timer array unit 0 count or capture	31	Address set in the DTCBAR register + 1FH	
End of channel 7 of timer array unit 0 count or capture	32	Address set in the DTCBAR register + 20H	
Channel 00 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	33	Address set in the DTCBAR register + 21H	
Channel 01 8-bit interval timer compare match	34	Address set in the DTCBAR register + 22H	
Channel 10 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	35	Address set in the DTCBAR register + 23H	
Channel 11 8-bit interval timer compare match	36	Address set in the DTCBAR register + 24H	
Channel 20 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	37	Address set in the DTCBAR register + 25H	
Channel 21 8-bit interval timer compare match	38	Address set in the DTCBAR register + 26H	
Channel 30 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	39	Address set in the DTCBAR register + 27H	
Channel 31 8-bit interval timer compare match	40	Address set in the DTCBAR register + 28H	
Fixed-cycle signal of realtime clock/alarm match detection	41	Address set in the DTCBAR register + 29H	
Interval signal detection of 32-bit interval timer	42	Address set in the DTCBAR register + 2AH	Lowest

Table 23 - 3 DTC Activation Sources and Vector Addresses (2/2)

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
Request to write to a configuration register of an individual channel of the capacitive sensing unit	43	Address set in the DTCBAR register + 2BH	 Highest
Request to transfer data measured by the capacitive sensing unit	44	Address set in the DTCBAR register + 2CH	
Comparator detection 0	45	Address set in the DTCBAR register + 2DH	
Comparator detection 1 ^{Note 3}	46	Address set in the DTCBAR register + 2EH	
Event output from the logic and event link controller	47	Address set in the DTCBAR register + 2FH	
Event output from the SNOOZE mode sequencer	48	Address set in the DTCBAR register + 30H	
RJ0 timer underflow	49	Address set in the DTCBAR register + 31H	
RJ1 timer underflow ^{Note 4}	50	Address set in the DTCBAR register + 32H	
End of TMKB40 count	51	Address set in the DTCBAR register + 33H	
End of TMKB41 count ^{Note 4}	52	Address set in the DTCBAR register + 34H	
End of TMKB42 count ^{Note 4}	53	Address set in the DTCBAR register + 35H	
External signal sampler edge detection	54	Address set in the DTCBAR register + 36H	
Voltage detection ^{Note 5}	55	Address set in the DTCBAR register + 37H	
			Lowest

Note 1. Only for 48- to 64-pin products with 256 or 512 Kbytes of flash memory and 80- and 100-pin products

Note 2. Only for 80- and 100-pin products

Note 3. Only for 52- to 100-pin products

Note 4. Only for 44- to 64-pin products with 256 or 512 Kbytes of flash memory and 80- and 100-pin products

Note 5. When bit 6 (LVD0SEL) of the option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

23.3.4 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 23 - 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	DACEN	SMSSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUN
	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.						

Caution Be sure to clear bit 1 to 0.

23.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 23 - 6 Format of DTC Control Register j (DTCCRj) (1/2)

Address: Refer to **23.3.2 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).								

Figure 23 - 6 Format of DTC Control Register j (DTCCRj) (2/2)

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

23.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

Figure 23 - 7 Format of DTC Block Size Register j (DTBLSj)

Address: Refer to **23.3.2 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.

23.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

This 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 23 - 8 Format of DTC Transfer Count Register j (DTCCTj)

Address: Refer to **23.3.2 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.

23.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

This 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 23 - 9 Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: Refer to **23.3.2 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.

23.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

This 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 23 - 10 Format of DTC Source Address Register j (DTSARj)

Address: Refer to **23.3.2 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.

23.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

This 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 23 - 11 Format of DTC Destination Address Register j (DTDARj)

Address: Refer to **23.3.2 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.

23.3.11 DTC activation enable register i (DTCENi) (i = 0 to 6)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. **Table 23 - 4** lists the correspondences between interrupt sources and bits DTCENi7 to DTCENi0. The DTCENi register can be set by a 1-bit or 8-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi7 to DTCENi0 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 23 - 12 Format of DTC Activation Enable Register i (DTCENi) (1/2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4), F02EDH (DTCEN5), F02EEH (DTCEN6)

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 23 - 12 Format of DTC Activation Enable Register i (DTCENi) (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 23 - 4 Correspondences between Interrupt Sources and Bits DTCENi7 to DTCENi0

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	INTP7	Key input	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/ CSI10 transfer end or buffer empty/ IIC10 transfer end	UART2 reception transfer end/ CSI21 transfer end or buffer empty/ IIC21 transfer end
DTCEN2	UART2 transmission transfer end/ CSI20 transfer end or buffer empty/ IIC20 transfer end	UARTA0 reception transfer end	UARTA0 transmission transfer end/ buffer empty	UARTA1 reception transfer end	UARTA1 transmission transfer end/ buffer empty	UARTA2 reception transfer end Note 1	UARTA2 transmission transfer end/ buffer empty Note 1	UARTA3 reception transfer end Note 2
DTCEN3	UARTA3 transmission transfer end/ buffer empty Note 2	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture	End of channel 4 of timer array unit 0 count or capture	End of channel 5 of timer array unit 0 count or capture	End of channel 6 of timer array unit 0 count or capture
DTCEN4	End of channel 7 of timer array unit 0 count or capture	Channel 00 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	Channel 01 8-bit interval timer compare match	Channel 10 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	Channel 11 8-bit interval timer compare match	Channel 20 8-bit interval timer compare match (in 8-bit/16-bit timer operation)	Channel 21 8-bit interval timer compare match	Channel 30 8-bit interval timer compare match (in 8-bit/16-bit timer operation)
DTCEN5	Channel 31 8-bit interval timer compare match	Fixed-cycle signal of realtime clock/ alarm match detection	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	Comparator detection 0	Comparator detection 1 Note 3	Event output from the logic and event link controller
DTCEN6	Event output from the SNOOZE mode sequencer	RJ0 timer underflow	RJ1 timer underflow Note 4	End of TMKB40 count	End of TMKB41 count Note 4	End of TMKB42 count Note 4	External signal sampler edge detection	Voltage detection Note 5

Note 1. Only for 48- to 64-pin products with 256 or 512 Kbytes of flash memory and 80- and 100-pin products

Note 2. Only for 80- and 100-pin products

Note 3. Only for 52- to 100-pin products

Note 4. Only for 44- to 64-pin products with 256 or 512 Kbytes of flash memory and 80- and 100-pin products

Note 5. When bit 6 (LVD0SEL) of the option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Caution For the bits to which no function is assigned, be sure to set their values to 0.

Remark i = 0 to 6

23.3.12 DTC base address register (DTCBAR)

This 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 23.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 23 - 13 Format of 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

23.4 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.

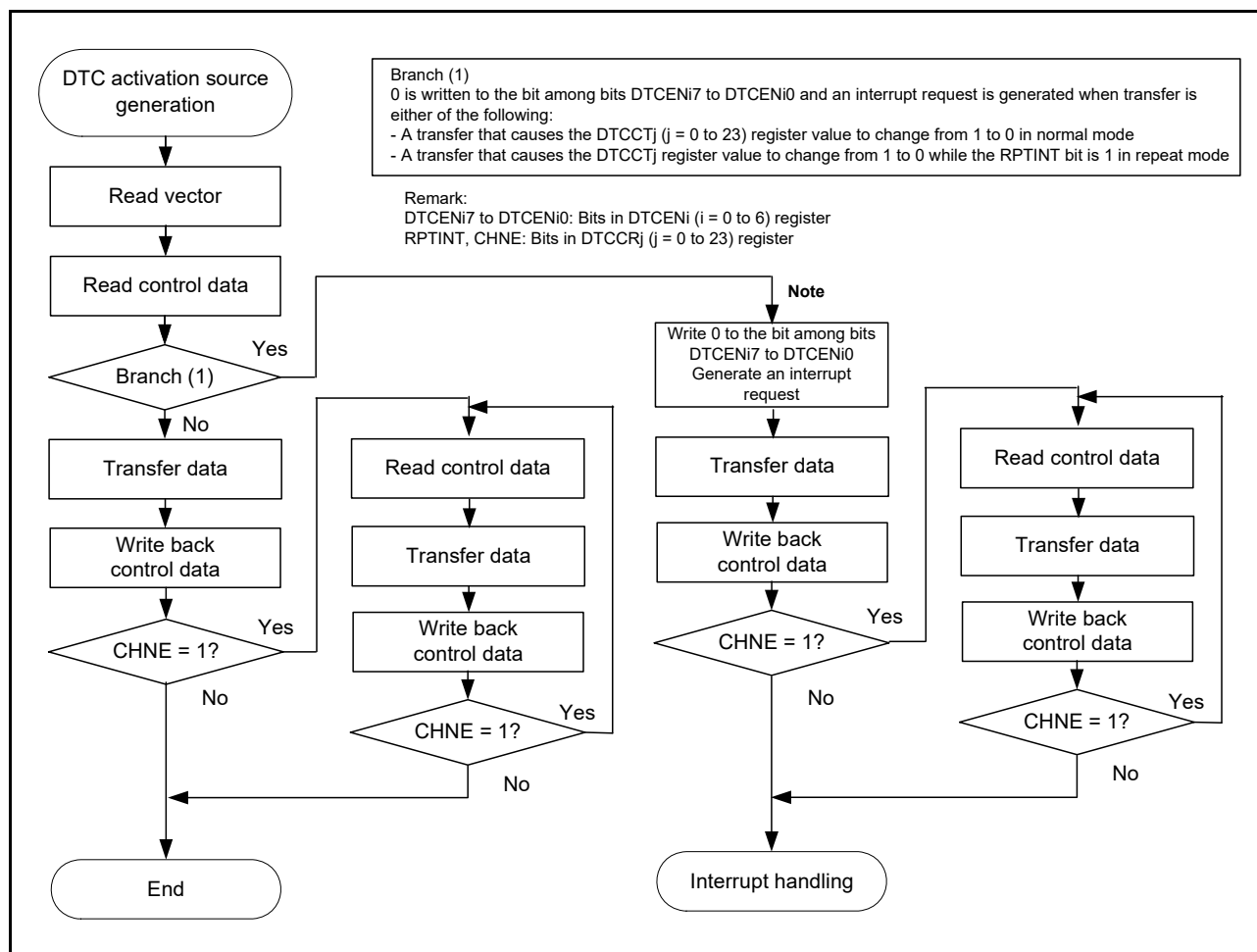
23.4.1 Activation sources

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 6) register. The DTC sets the corresponding bit among bits DTCENi7 to DTCENi0 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 23 - 14 shows the DTC internal operation flowchart.

Figure 23 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi7 to DTCENi0 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

23.4.2 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 DTCENi7 to DTCENi0 (i = 0 to 6) in the DTCENi register to 0 (activation disabled). **Table 23 - 5** shows the register functions in normal mode. **Figure 23 - 15** shows the data transfers in normal mode.

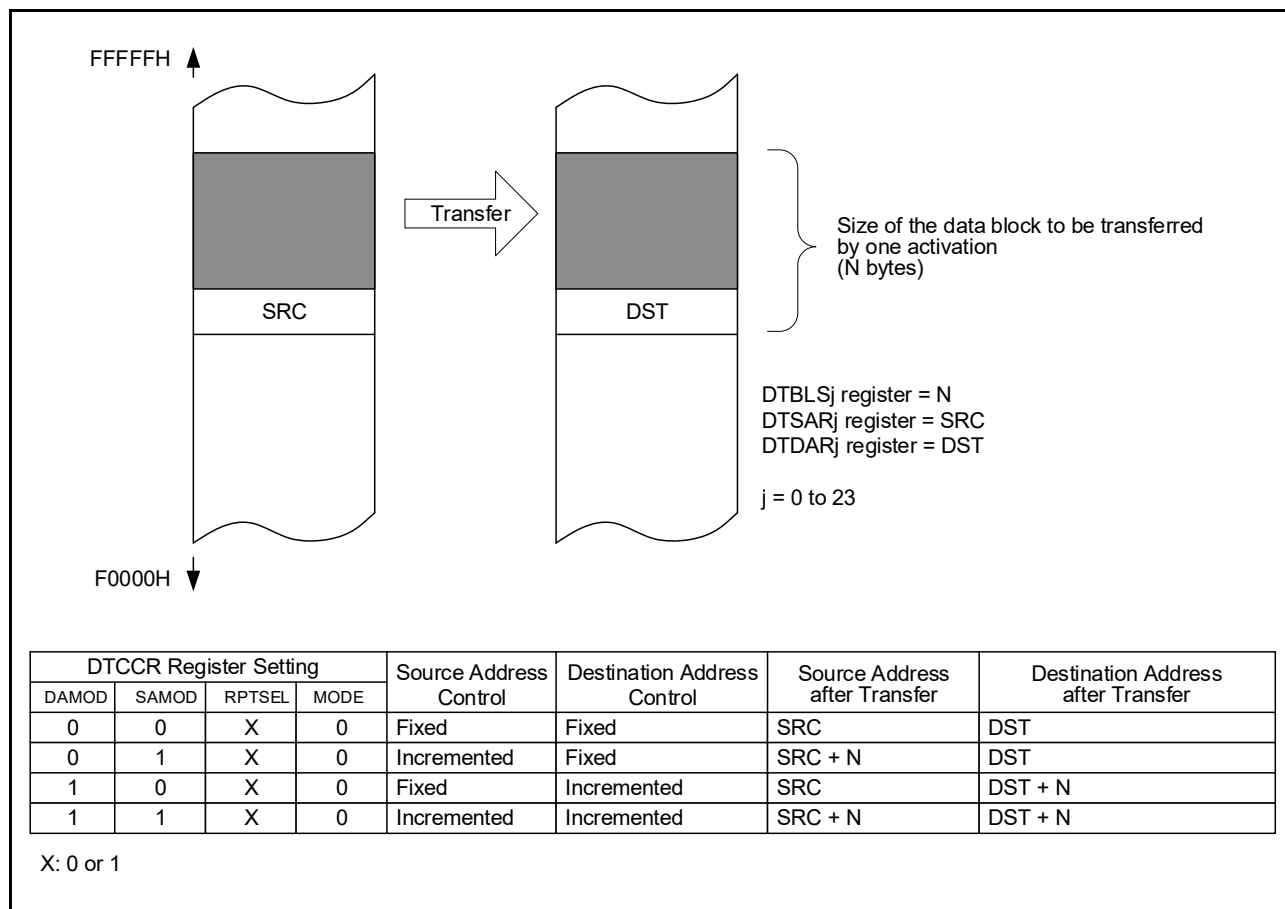
Table 23 - 5 Register Functions in Normal Mode

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 23 - 15 Data Transfers in Normal Mode

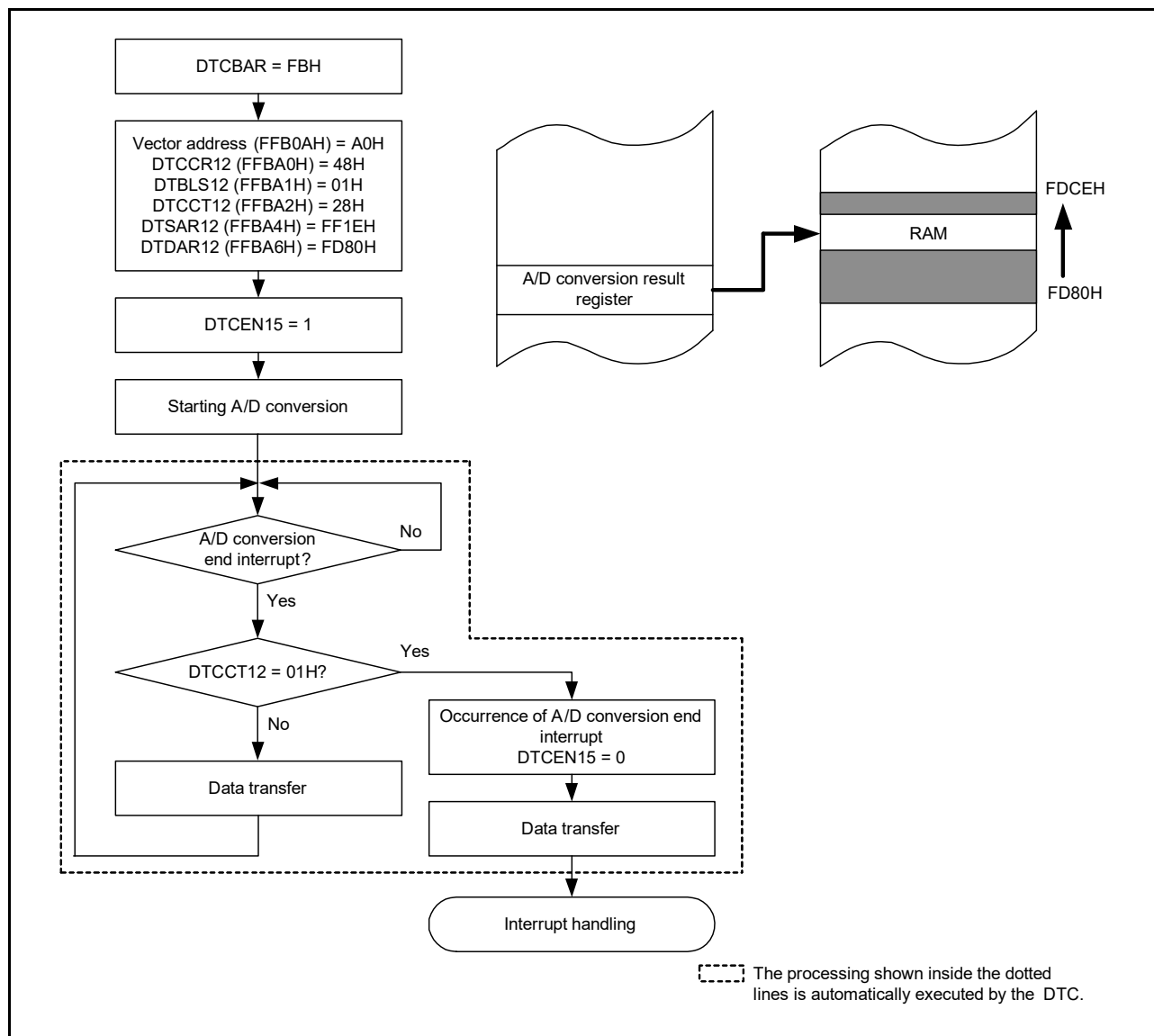


1. Example 1 of using normal mode: Consecutively capturing A/D conversion results

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 FFB0AH 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 FFDCEH of RAM 40 times.

Figure 23 - 16 Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results



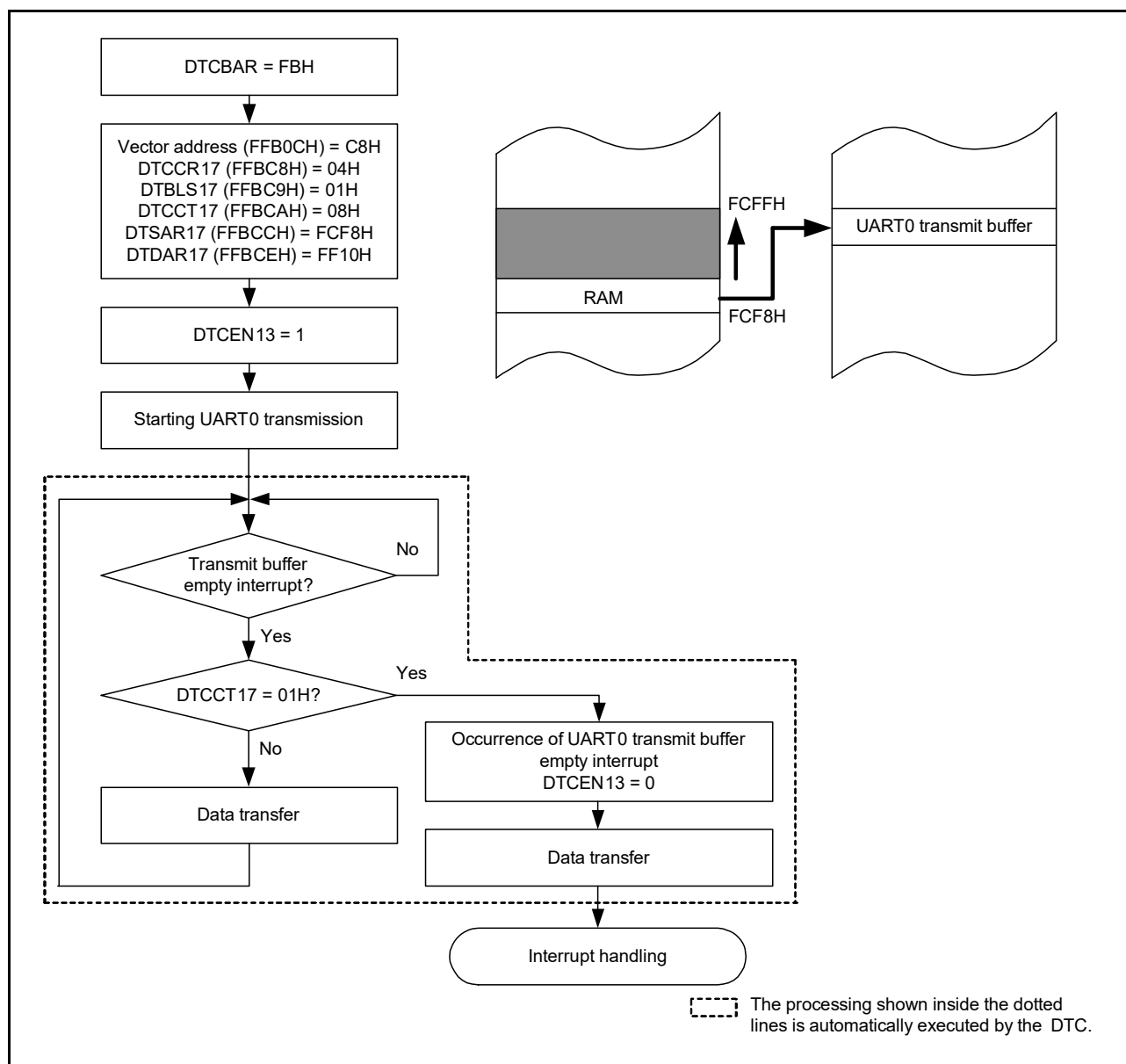
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

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 23 - 17 Example 2 of Using Normal Mode: UART0 Consecutive Transmission



The value of the DTRL17 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.

23.4.3 Repeat mode

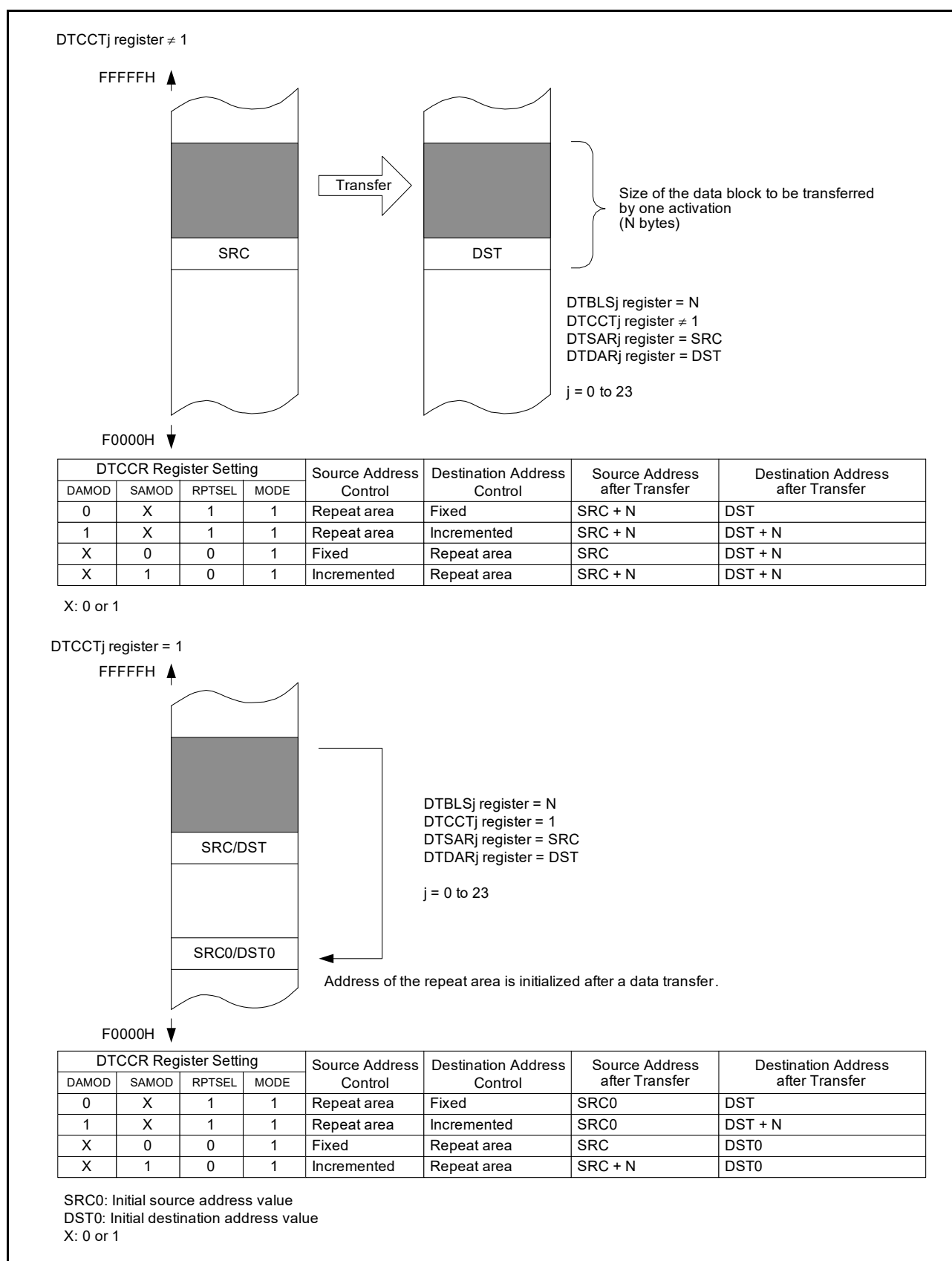
One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (i = 0 to 23) register and the address specified 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 DTCENi7 to DTCENi0 (i = 0 to 6) 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 DTCENi7 to DTCENi0 are not set to 0. **Table 23 - 6** lists the register functions in repeat mode. **Figure 23 - 18** shows the data transfers in repeat mode.

Table 23 - 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 23 - 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.

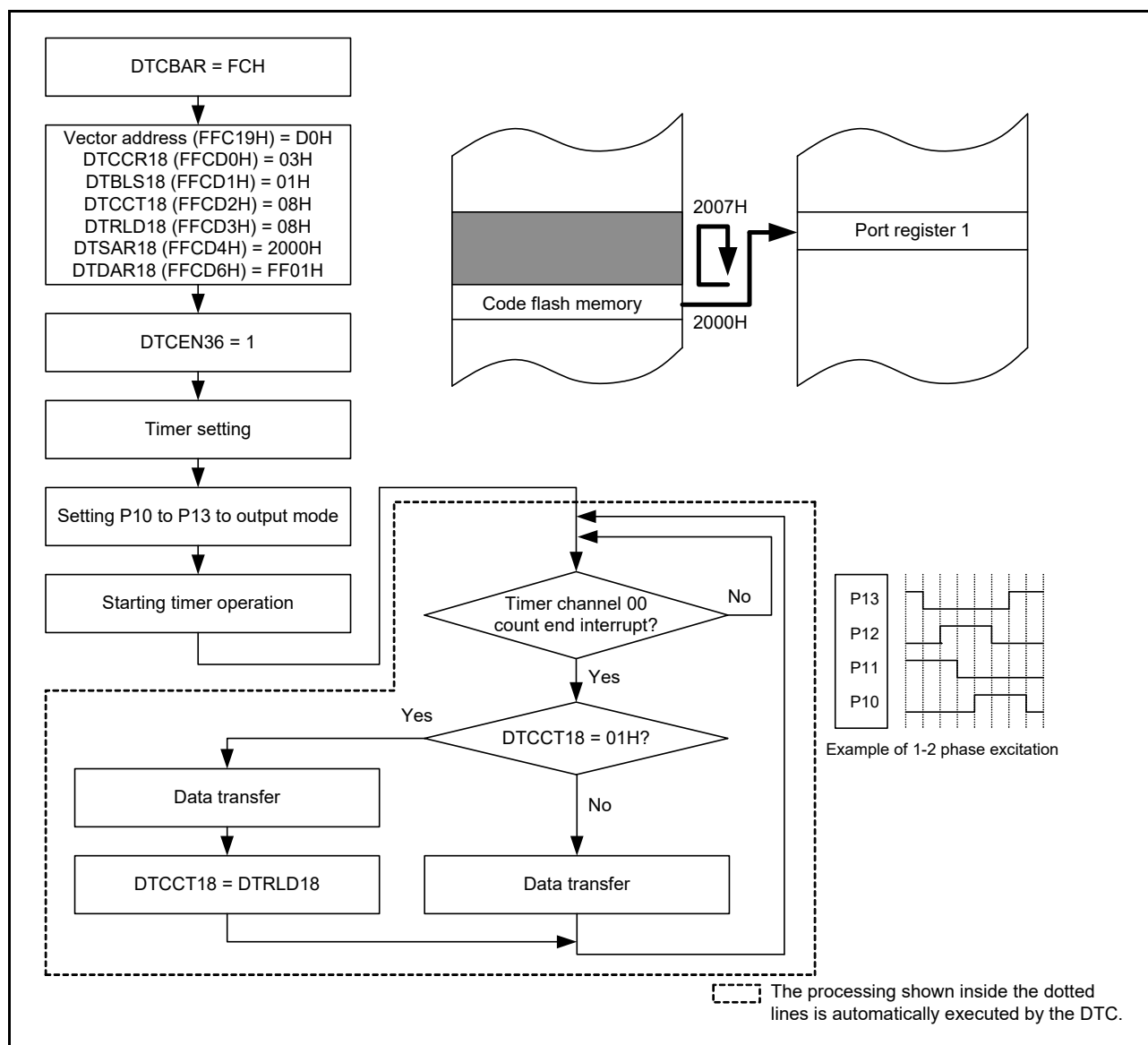
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

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 FFC19H 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 23 - 19 Example 1 of Using Repeat Mode: Outputting Stepping Motor Control Pulses Using Port Pins



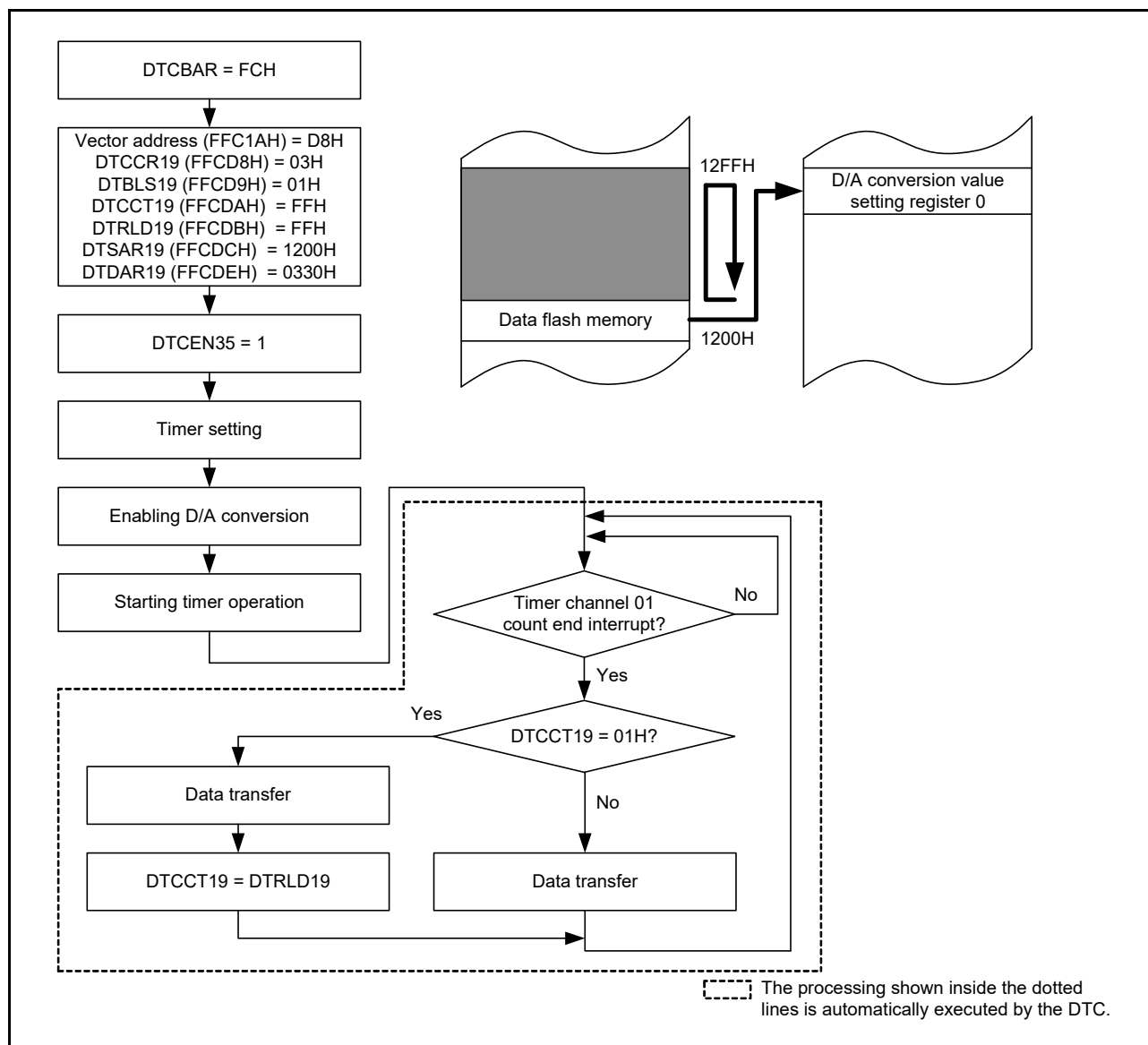
Caution To stop the output, stop the timer first and then clear DTCEN36.

2. Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter

The DTC is activated using an interrupt of the interval timer function of channel 1 of timer array unit 0, and the table of the sine wave stored in the data flash memory is transferred to the 8-bit D/A conversion value setting register 0 (F0330H). The timer interval time is set to the D/A output setup time.

- The vector address is FFC1AH and control data is allocated at FFCD8H to FFCDFH.
- Transfers 255-byte data at addresses from F1200H to F12FEH of the data flash memory to the D/A conversion value setting register 0 (F0330H).
- A repeat mode interrupt is disabled.

Figure 23 - 20 Example 2 of Using Repeat Mode: Outputting a Sine Wave Using the 8-bit D/A Converter

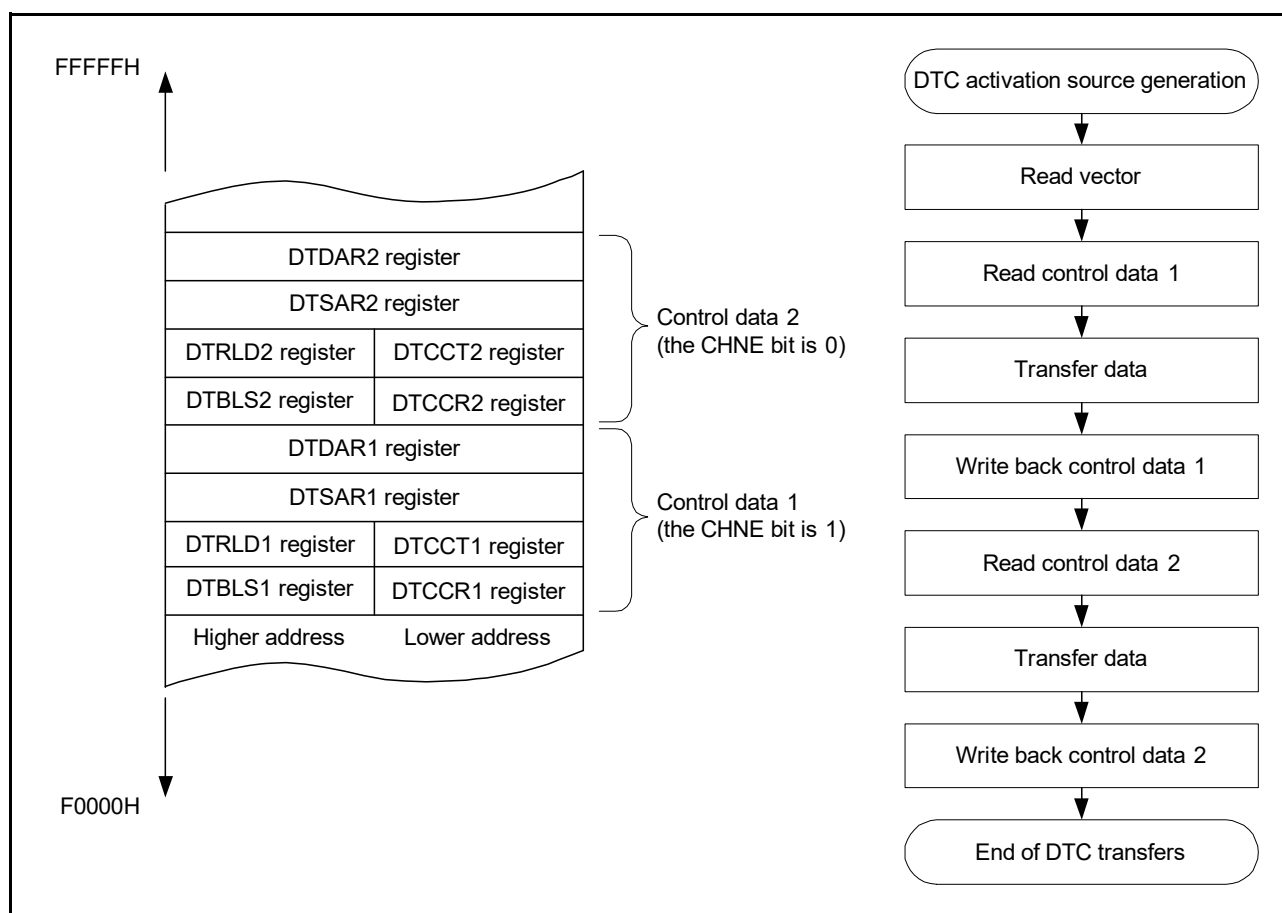


Caution To stop the output, stop the timer first and then clear DTCEN35.

23.4.4 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 23 - 21** shows the data transfers during chain transfers.

Figure 23 - 21 Data Transfers during Chain Transfers



Note 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

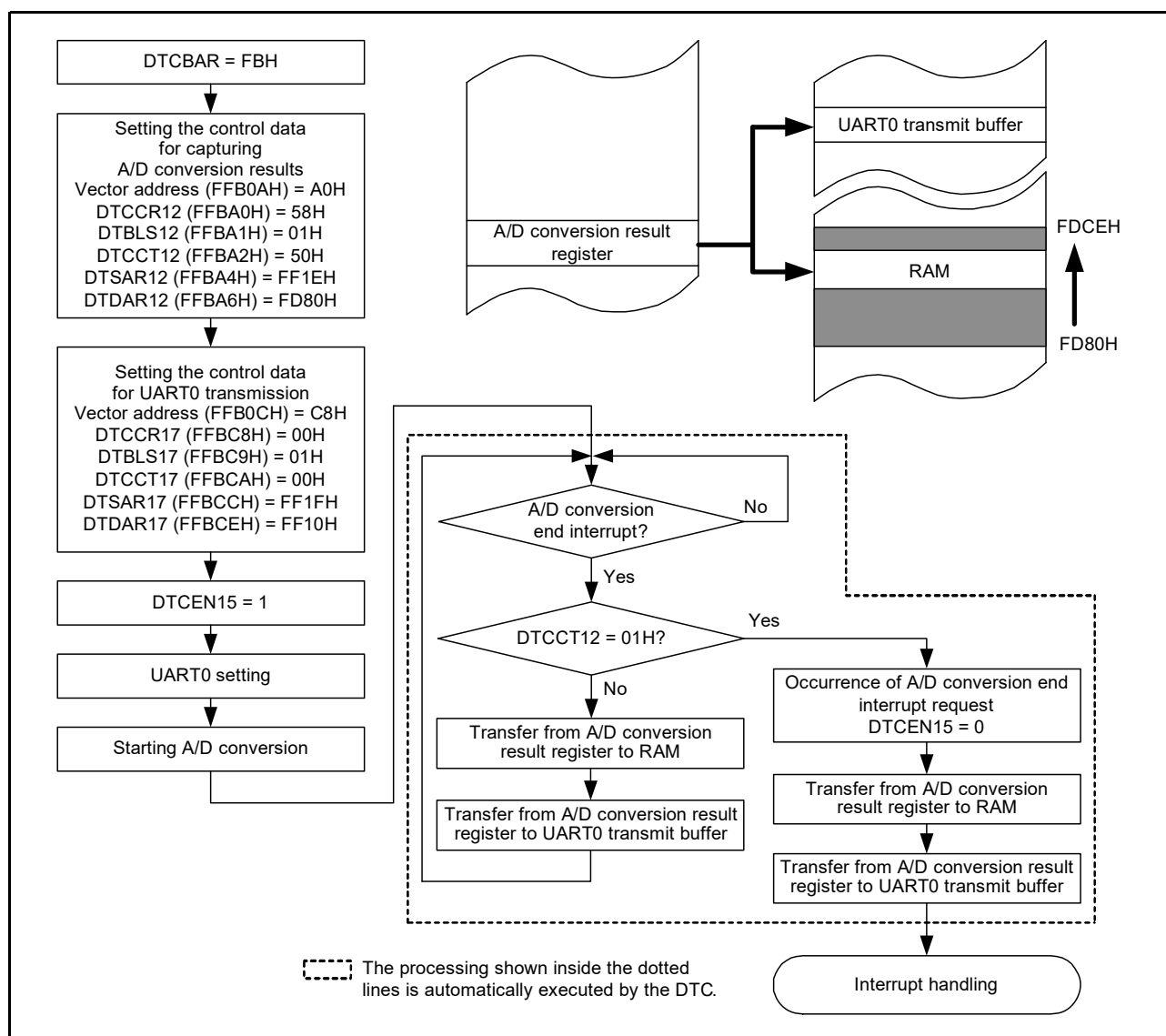
Note 2. During chain transfers, bits DTCENi7 to DTCENi0 (i = 0 to 6) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

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 FFBC8H to FFBCFH.
- Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the 1 higher-order byte (FFF1FH) of the A/D conversion result register to the UART0 transmit buffer (FFF10H).

Figure 23 - 22 Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART0 Transmission



23.5 Points for Caution when the DTC is to be Used

23.5.1 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 DTCENi7 to DTCENi0 (i = 0 to 6) in the DTCENi 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 DTCENi7 to DTCENi0 (i = 0 to 6) in the DTCENi register is 0 (activation disabled).

23.5.2 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.
- If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be used as the DTC control data area or DTC vector table area.

23.5.3 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.

23.5.4 Operation when accessing data flash memory space

When accessing the data flash memory 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

23.5.5 Number of DTC execution clock cycles

Table 23 - 7 lists the operations following DTC activation and required number of cycles for each operation.

Table 23 - 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 23 - 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 23 - 9 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 23 - 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 23 - 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.

23.5.6 DTC response time

Table 23 - 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 23 - 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 **23.5.3 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 a register, which entails a further wait
Maximum response time: Maximum response time for each condition + 1 cycle

Remark 1 cycle: 1/fCLK (fCLK: CPU/peripheral hardware clock)

23.5.7 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 **23.3.3 Vector table**.
- When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, check the monitor flag (CnMON) of the comparator before enabling DTC activation as required (n = 0, 1).
 - The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator (CnEPO = 0), and IVCMP > comparator n reference voltage^{Note}
 - The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator (CnEPO = 1), and IVCMP < comparator n reference voltage^{Note}

Note n = 0: IVREF0 or internal reference voltage
n = 1: IVREF1 or D/A converter output 0

23.5.8 Operation in standby mode

State	DTC Operation
HALT mode	Operable ^{Note 1}
STOP mode	DTC activation sources can be accepted ^{Note 2}
SNOOZE mode	Operable ^{Notes 3, 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. 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 3. 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 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.

Caution If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.

Remark 44-pin products: p = 00; m = 0; q = 0, 2
48- to 100-pin products: p = 00, 20; m = 0, 1; q = 0, 2

Section 24 Logic and Event Link Controller (ELCL)

24.1 Functions of ELCL

The logic and event link controller (ELCL) links signals output by peripheral functions to other specified peripheral functions through internal logic cell blocks, allowing direct communications between peripheral functions without CPU intervention. The ELCL has the following functions.

- Eight event signals selected from among up to 100 signals from peripheral functions can be linked to specified peripheral functions without CPU intervention. The destinations for connection of these eight signals are selected from among up to 25 event input terminals of the specified peripheral functions.
- Event signals from peripheral functions can be input to logic cells (including AND, OR, and EX-OR circuits) to change the link conditions.
- Event signals from peripheral functions can also be input to selectors to specify conditions for activating the selected peripheral functions.
- Event signals from peripheral functions can also be input to flip-flops to link the signals to the selected peripheral functions in synchronization with a selected clock.

24.2 Configuration of ELCL

Figure 24 - 1 shows a block diagram of the ELCL.

Figure 24 - 1 Block Diagram of ELCL

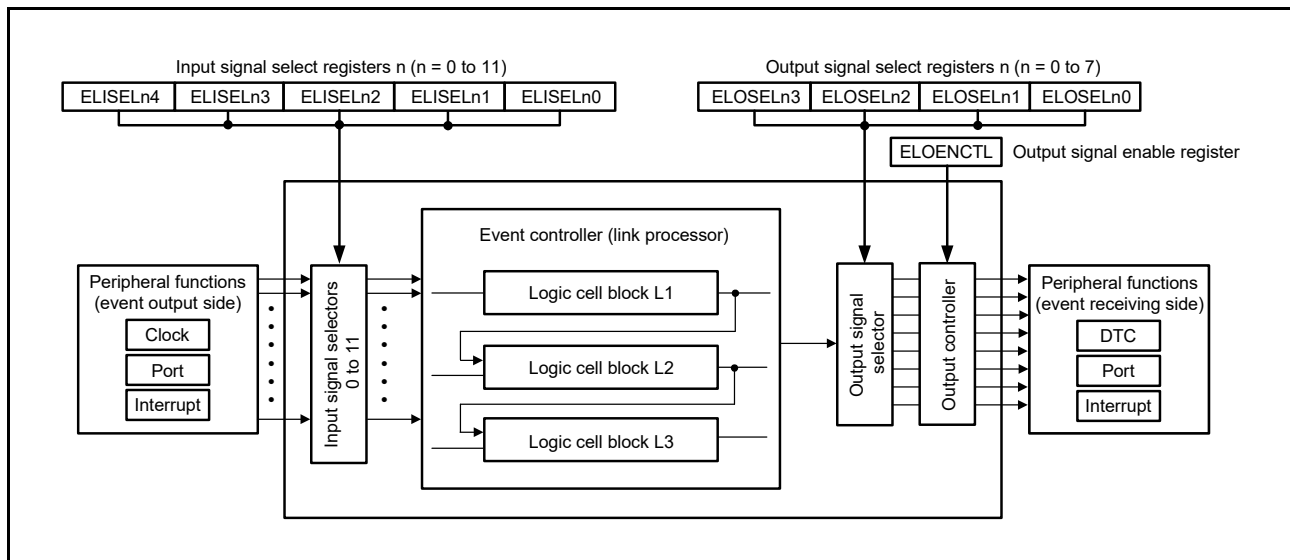


Figure 24 - 2 Block Diagram of Logic Cell Block L1

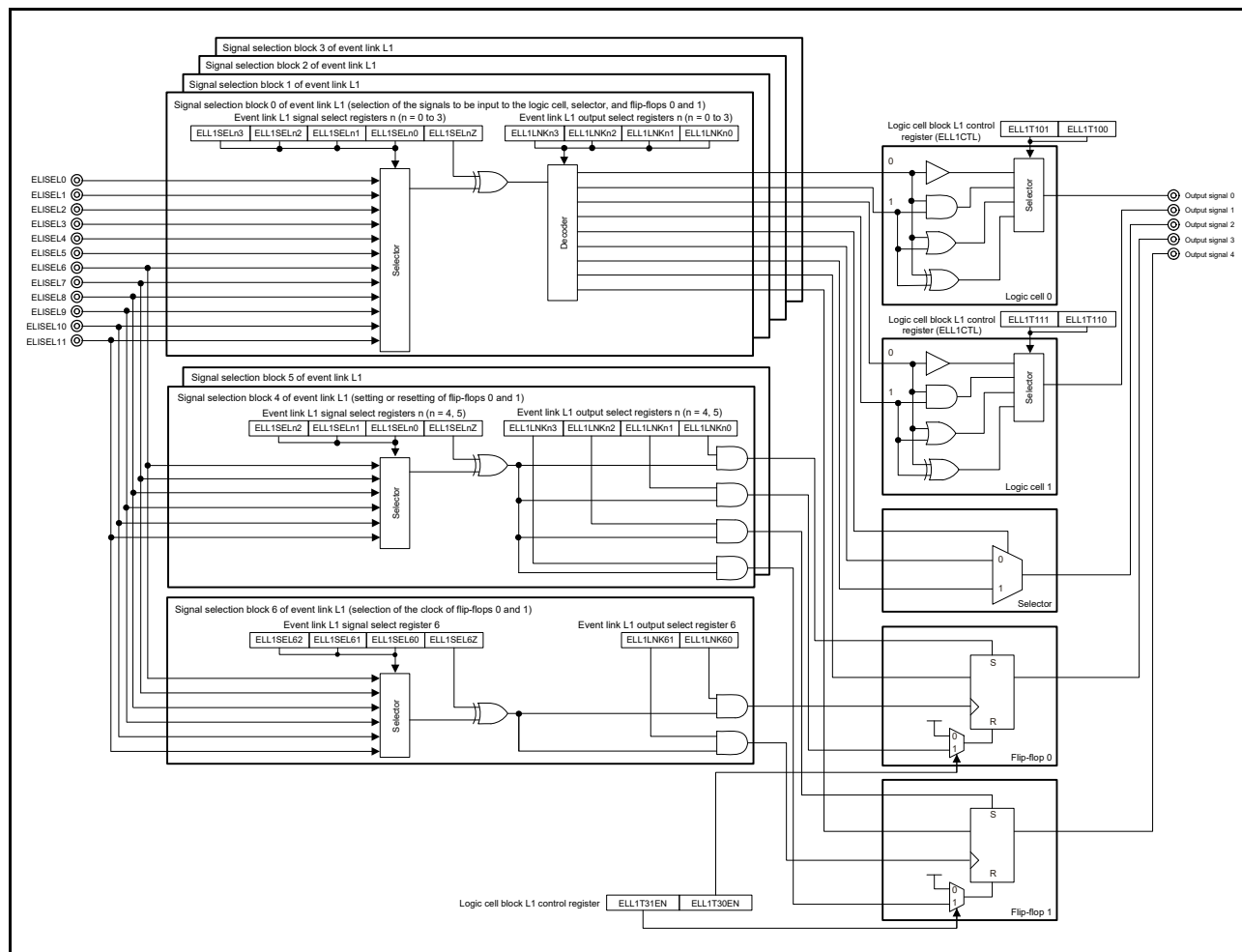


Figure 24 - 3 Block Diagram of Logic Cell Block L2

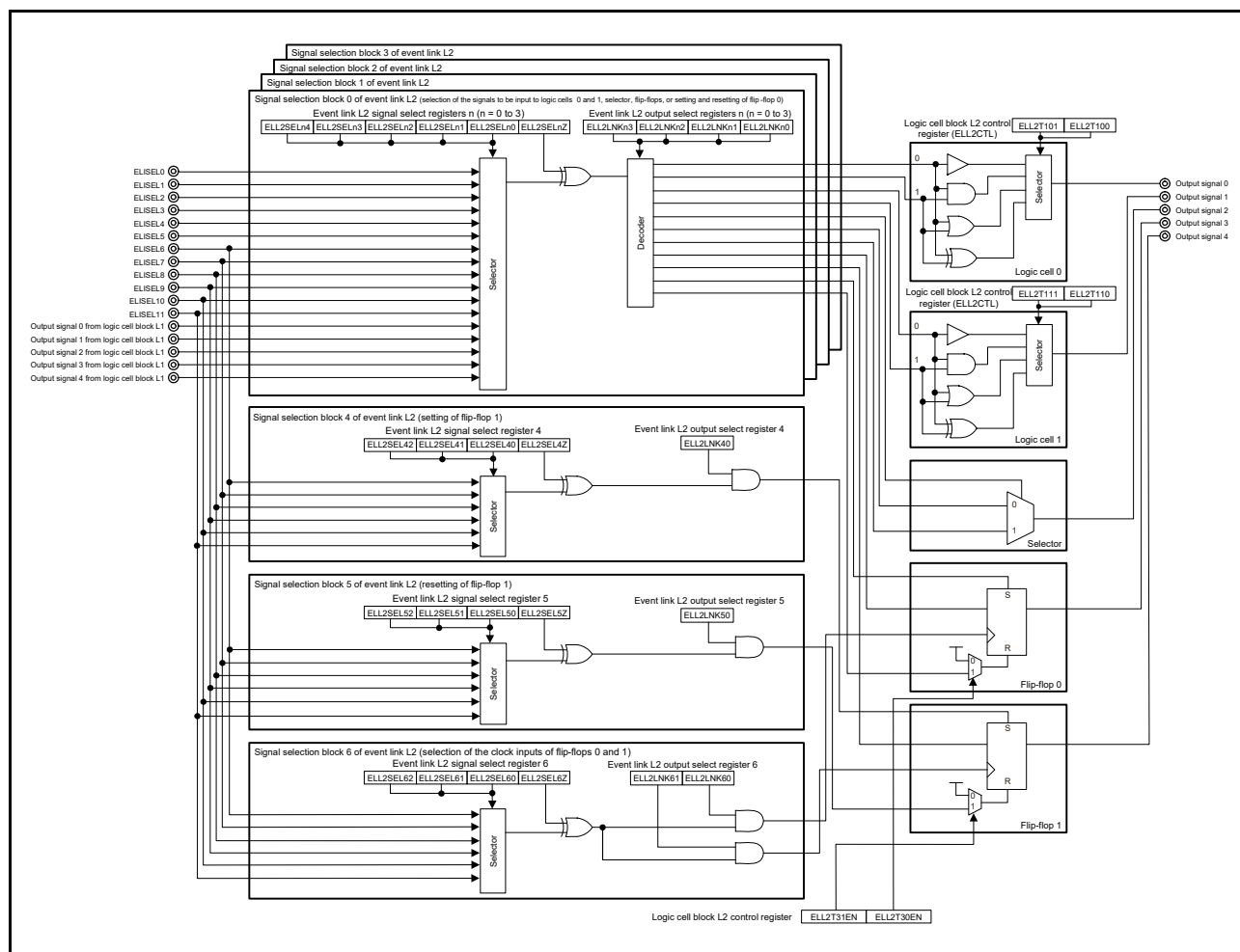


Figure 24 - 4 Block Diagram of Logic Cell Block L3

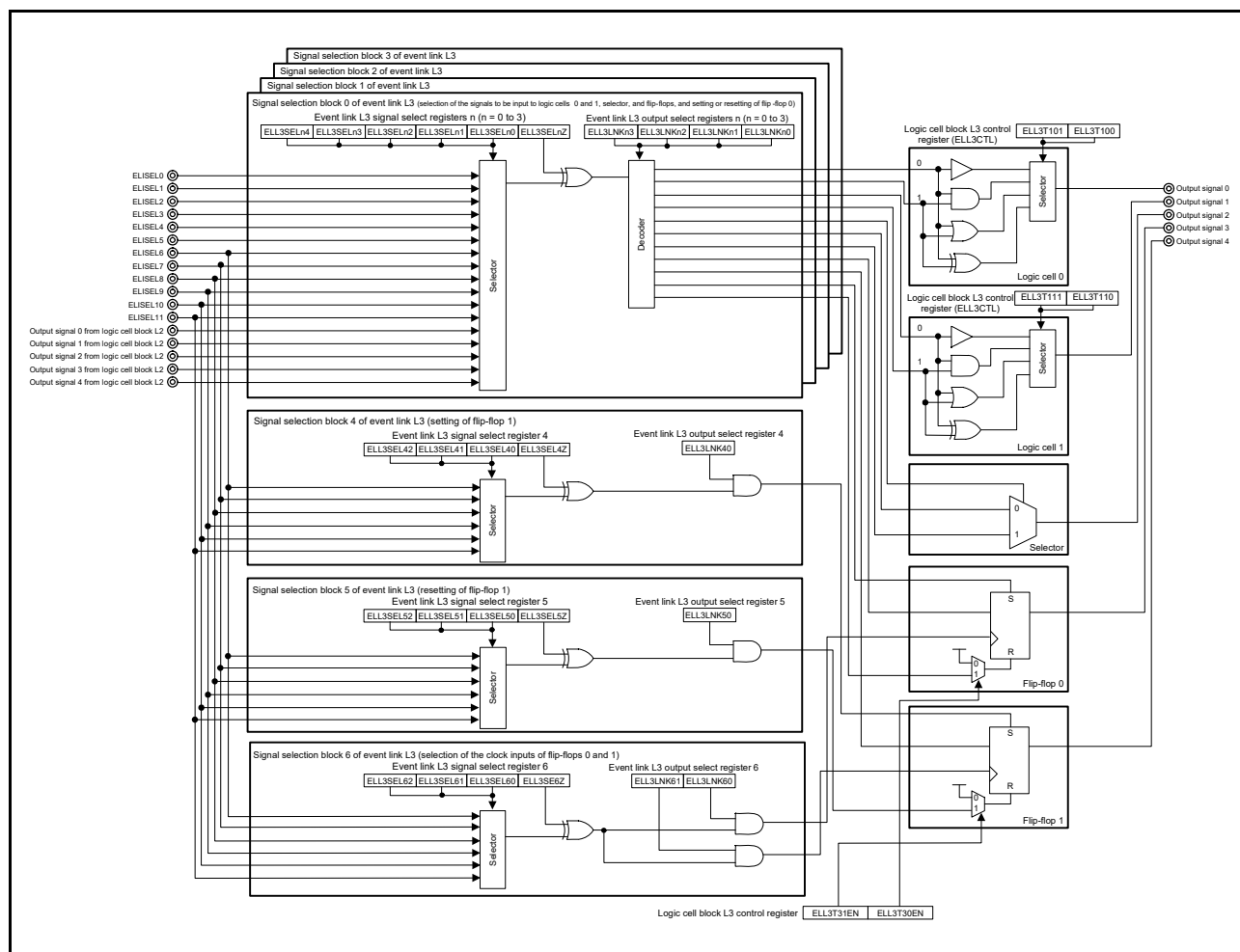


Table 24 - 1 and **Table 24 - 2** list the connections in logic cell block L1. The signals to be input to logic cell block L1 are selectable from among the event-output peripheral functions specified in the input signal select registers n (ELISELn) (n = 0 to 11). The event link L1 signal select registers n (ELL1SELn) (n = 0 to 6) can be used to specify the signal to be input to signal selection block n (n = 0 to 6) of event link L1. The event link L1 output select registers n (ELL1LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL1CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L1 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L2.

Table 24 - 1 Connections in Logic Cell Block L1 (1)

Signal selection block n of event link L1	Input Signal	Destination of the signal to be output from signal selection block n of event link L1
Signal selection block 0 of event link L1 Note 1	- Signals selected by the ELISEL0 to ELISEL11 registers - Use the ELL1SEL0 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Logic cell 0 (input 0 or input 1) - Logic cell 1 (input 0 or input 1) - Selector (selection, input 0 or input 1) - Flip-flop 0 (input) - Flip-flop 1 (input) - Use the ELL1LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L1 Note 1	- Signals selected by the ELISEL0 to ELISEL11 registers - Use the ELL1SEL1 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Logic cell 0 (input 0 or input 1) - Logic cell 1 (input 0 or input 1) - Selector (selection, input 0 or input 1) - Flip-flop 0 (input) - Flip-flop 1 (input) - Use the ELL1LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L1 Note 1	- Signals selected by the ELISEL0 to ELISEL11 registers - Use the ELL1SEL2 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Logic cell 0 (input 0 or input 1) - Logic cell 1 (input 0 or input 1) - Selector (selection, input 0 or input 1) - Flip-flop 0 (input) - Flip-flop 1 (input) - Use the ELL1LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L1 Note 1	- Signals selected by the ELISEL0 to ELISEL11 registers - Use the ELL1SEL3 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Logic cell 0 (input 0 or input 1) - Logic cell 1 (input 0 or input 1) - Selector (selection, input 0 or input 1) - Flip-flop 0 (input) - Flip-flop 1 (input) - Use the ELL1LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L1 Note 2	- Signals selected by the ELISEL6 to ELISEL11 registers - Use the ELL1SEL4 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Flip-flop 0 (set or reset) - Flip-flop 1 (set or reset) - Use the ELL1LNK4 register to select a destination.
Signal selection block 5 of event link L1 Note 2	- Signals selected by the ELISEL6 to ELISEL11 registers - Use the ELL1SEL5 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Flip-flop 0 (set or reset) - Flip-flop 1 (set or reset) - Use the ELL1LNK5 register to select a destination.
Signal selection block 6 of event link L1	- Signals selected by the ELISEL6 to ELISEL11 registers - Use the ELL1SEL6 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	- Flip-flop 0 (clock) - Flip-flop 1 (clock) - Use the ELL1LNK6 register to select a destination.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L1; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 24 - 2 Connections in Logic Cell Block L1 (2)

Signal output from logic cell block L1	Signal source
Signal 0 output from logic cell block L1	Output from logic cell 0 in logic cell block L1
Signal 1 output from logic cell block L1	Output from logic cell 1 in logic cell block L1
Signal 2 output from logic cell block L1	Output from the selector in logic cell block L1
Signal 3 output from logic cell block L1	Output from flip-flop 0 in logic cell block L1
Signal 4 output from logic cell block L1	Output from flip-flop 1 in logic cell block L1

Table 24 - 3 and **Table 24 - 4** list the connections in logic cell block L2. The signals to be input to logic cell block L2 are selectable from among the event-output peripheral functions specified by the input signal select registers n (ELISELn) (n = 0 to 11) and the signals output from logic cell block L1. The event link L2 output select registers n (ELL1LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL2CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L2 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L3.

Table 24 - 3 Connections in Logic Cell Block L2 (1)

Signal selection block n of event link L2	Input Signal	Destination of the signal to be output from signal selection block n of event link L2
Signal selection block 0 of event link L2 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL0 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L2 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL1 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L2 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL2 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L2 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL3 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L2 Note 2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL4 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 1 (set) Use the ELL2LNK4 register to select one of the above destinations.
Signal selection block 5 of event link L2 Note 2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL5 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 1 (reset) Use the ELL2LNK5 register to select one of the above destinations.
Signal selection block 6 of event link L2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL6 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 0 (clock) • Flip-flop 1 (clock) Use the ELL2LNK6 register to select one of the above destinations.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L2; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 24 - 4 Connections in Logic Cell Block L2 (2)

Signal output from logic cell block L2	Signal source
Signal 0 output from logic cell block L2	Output from logic cell 0 in logic cell block L2
Signal 1 output from logic cell block L2	Output from logic cell 1 in logic cell block L2
Signal 2 output from logic cell block L2	Output from the selector in logic cell block L2
Signal 3 output from logic cell block L2	Output from flip-flop 0 in logic cell block L2
Signal 4 output from logic cell block L2	Output from flip-flop 1 in logic cell block L2

Table 24 - 5 and **Table 24 - 6** list the connections in logic cell block L3. The signals to be input to logic cell block L3 are selectable from among the event-output peripheral functions specified in the input signal select registers n (ELISELn) (n = 0 to 11) and the signals output from logic cell block L2. The event link L1 signal select registers n (ELL1SELn) (n = 0 to 6) can be used to specify the signal to be input to signal selection block n (n = 0 to 6) of event link L3. The event link L3 output select registers n (ELL3LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL3CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L3 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L3.

Table 24 - 5 Connections in Logic Cell Block L3 (1)

Signal selection block n of event link L3	Input Signal	Destination of the signal to be output from signal selection block n of event link L3
Signal selection block 0 of event link L3 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL0 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L3 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL1 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L3 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL2 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L3 Notes 1, 2	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL3 register to select one signal to be input to logic cell block 3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L3 Note 2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL4 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 1 (set) Use the ELL3LNK4 register to select one of the above destinations.
Signal selection block 5 of event link L3 Note 2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL5 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 1 (reset) Use the ELL3LNK5 register to select one of the above destinations.
Signal selection block 6 of event link L3	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL6 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 0 (clock) • Flip-flop 1 (clock) Use the ELL3LNK6 register to select one of the above destinations.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L3; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 24 - 6 Connections in Logic Cell Block L3 (2)

Signal output from logic cell block L3	Signal source
Signal 0 output from logic cell block L3	Output from logic cell 0 in logic cell block L3
Signal 1 output from logic cell block L3	Output from logic cell 1 in logic cell block L3
Signal 2 output from logic cell block L3	Output from the selector in logic cell block L3
Signal 3 output from logic cell block L3	Output from flip-flop 0 in logic cell block L3
Signal 4 output from logic cell block L3	Output from flip-flop 1 in logic cell block L3

24.3 Registers for Controlling the ELCL

The following registers are used to control the ELCL.

- Input signal select registers n (ELISELn) (n = 0 to 11)
- Event link L1 signal select registers n (ELL1SELn) (n = 0 to 3)
- Event link L1 signal select registers n (ELL1SELn) (n = 4, 5)
- Event link L1 signal select register 6 (ELL1SEL6)
- Logic cell block L1 control register (ELL1CTL)
- Event link L1 output select registers n (ELL1LNKn) (n = 0 to 3)
- Event link L1 output select registers n (ELL1LNKn) (n = 4, 5)
- Event link L1 output select register 6 (ELL1LNK6)
- Event link L2 signal select registers n (ELL2SELn) (n = 0 to 3)
- Event link L2 signal select registers n (ELL2SELn) (n = 4, 5)
- Event link L2 signal select register 6 (ELL2SEL6)
- Logic cell block L2 control register (ELL2CTL)
- Event link L2 output select registers n (ELL2LNKn) (n = 0 to 3)
- Event link L2 output select registers n (ELL2LNKn) (n = 4, 5)
- Event link L2 output select register 6 (ELL2LNK6)
- Event link L3 signal select registers n (ELL3SELn) (n = 0 to 3)
- Event link L3 signal select registers n (ELL3SELn) (n = 4, 5)
- Event link L3 signal select register 6 (ELL3SEL6)
- Logic cell block L3 control register (ELL3CTL)
- Event link L3 output select registers n (ELL3LNKn) (n = 0 to 3)
- Event link L3 output select registers n (ELL3LNKn) (n = 4, 5)
- Event link L3 output select register 6 (ELL3LNK6)
- Output signal select registers n (ELOSELn) (n = 0 to 7)
- Output signal enable register (ELOENCTL)
- Output signal monitor register (ELOMONI)
- Port mode registers (PMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

24.3.1 Input signal select registers n (ELISELn) (n = 0 to 11)

These registers are used to select event signals to be input from event-outputting peripheral functions. The event signals selected by these registers are linked to the event-receiving peripheral functions through logic cell blocks L1 to L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 5 Format of Input Signal Select Registers n (ELISELn) (1/3)

Address: F0680H (ELISEL0) to F068BH (ELISEL11)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELISELn	0	0	0	ELISEL n4	ELISEL n3	ELISEL n2	ELISEL n1	ELISEL n0

Figure 24 - 5 Format of Input Signal Select Registers n (ELISELn) (2/3)
ELISELn (n = 0 to 5)

ELISEL n4	ELISEL n3	ELISEL n2	ELISEL n1	ELISEL n0	Event Source
0	0	0	0	0	Transmit data output from SAU0 channel 0
0	0	0	0	1	Transmit data output from SAU0 channel 1
0	0	0	1	0	Serial clock output from SAU0 channel 0
0	0	0	1	1	Serial clock output from SAU0 channel 1
0	0	1	0	0	Input pin (P00/EI00)
0	0	1	0	1	Input pin (P17/EI17)
0	0	1	1	0	Serial clock output from UARTA0 ^{Note 1}
0	0	1	1	1	Output from TAU0 channel 0
0	1	0	0	0	Output from TAU0 channel 1
0	1	0	0	1	Output from TAU0 channel 2
0	1	0	1	0	Output from TAU0 channel 3
0	1	0	1	1	Output from TAU0 channel 4
0	1	1	0	0	Output from TAU0 channel 5
0	1	1	0	1	Output from TAU0 channel 6
0	1	1	1	0	Fixed to 0
0	1	1	1	1	Input pin (P76/EI76)
1	0	0	0	0	Input pin (P77/EI77)
1	0	0	0	1	Comparator detection 0
1	0	0	1	0	Comparator detection 1 ^{Note 1}
1	0	0	1	1	Transmit data output 0 from UARTA0
1	0	1	0	0	Input pin (P12/EI12)
1	0	1	0	1	Input pin 0 ^{Note 2}
1	0	1	1	0	Input pin 1 ^{Note 2}
1	0	1	1	1	INTC0 ^{Note 3}
1	1	0	0	0	INTC1 ^{Note 3}
1	1	0	0	1	INTC2 ^{Note 3}
1	1	0	1	0	INTC3 ^{Note 3}
1	1	0	1	1	Input source ^{Note 2}
1	1	1	0	0	Output from flip-flop 0 of logic cell block L1 in the ELCL
1	1	1	0	1	Output from flip-flop 1 of logic cell block L1 in the ELCL
1	1	1	1	0	Output from flip-flop 1 of logic cell block L2 in the ELCL
1	1	1	1	1	Output from flip-flop 1 of logic cell block L3 in the ELCL

Figure 24 - 5 Format of Input Signal Select Registers n (ELISELn) (3/3)
ELISELn (n = 6 to 11)

ELISEL n4	ELISEL n3	ELISEL n2	ELISEL n1	ELISEL n0	Event Source
0	0	0	0	0	Transmit data output from SAU0 channel 0
0	0	0	0	1	Transmit data output from SAU0 channel 1
0	0	0	1	0	Serial clock output from SAU0 channel 0
0	0	0	1	1	Serial clock output from SAU0 channel 1
0	0	1	0	0	Input pin (P00/EI00)
0	0	1	0	1	Input pin (P17/EI17)
0	0	1	1	0	Serial clock output from UARTA0 ^{Note 1}
0	0	1	1	1	Output from TAU0 channel 0
0	1	0	0	0	Output from TAU0 channel 1
0	1	0	0	1	Output from TAU0 channel 2
0	1	0	1	0	Output from TAU0 channel 3
0	1	0	1	1	Output from TAU0 channel 4
0	1	1	0	0	Output from TAU0 channel 5
0	1	1	0	1	Output from TAU0 channel 6
0	1	1	1	0	Fixed to 0
0	1	1	1	1	Input pin (P76/EI76)
1	0	0	0	0	Input pin (P77/EI77)
1	0	0	0	1	Comparator detection 0
1	0	0	1	0	Comparator detection 1 ^{Note 1}
1	0	0	1	1	Transmit data output 0 from UARTA0
1	0	1	0	0	Input pin (P12/EI12)
1	0	1	0	1	INTC4 ^{Note 4}
1	0	1	1	0	INTC5 ^{Note 4}
1	0	1	1	1	INTC6 ^{Note 4}
1	1	0	0	0	INTC7 ^{Note 4}
1	1	0	0	1	INTC8 ^{Note 4}
1	1	0	1	0	INTC9 ^{Note 4}
1	1	0	1	1	fCLK
1	1	1	0	0	fIHP ^{Note 5}
1	1	1	0	1	fIMP
1	1	1	1	0	fSXP
Others					Setting prohibited

Note 1. This is only supported in the 52- to 100-pin products.

Note 2. The event sources selected for input pins 0 and 1 and the input sources depend on which of the ELISELn (n = 0 to 5) registers is being set.

Register	Event Source		
	Input Pin 0	Input Pin 1	Input source
ELISEL0	Input pin (P20/EI20)	Input pin (P16/EI16)	Input pin (P137/EI137)
ELISEL1	Input pin (P21/EI21)	Input pin (P121/EI121)	Input pin (P137/EI137)
ELISEL2	Input pin (P06/EI06)	Input pin (P122/EI122)	ELCITL0
ELISEL3	Input pin (P07/EI07)	Input pin (P06/EI06)	ELCITL1
ELISEL4	Input pin (P26/EI26)	Input pin (P00/EI00)	ELCITL2
ELISEL5	Input pin (P27/EI27)	Input pin (P26/EI26)	ELCITL3

Note 3. The interrupt sources that are selectable as event sources for INTC0 to INTC3 depend on which of the ELISELn (n = 0 to 5) registers is being set. Use the interrupt request signals as the hardware triggers for event-receiving peripheral functions.

Register	Event Source				
	INTC0	INTC1		INTC2	INTC3
ELISEL0	INTP6	INTURE0	INTIT70Note 6	INTUT3Note 7	INTUT1
ELISEL1	INTP7	INTURE1	INTTMKB1 Note 6	INTUR3Note 7	INTUR1
ELISEL2	INTIT10	INTURE2 Note 8	INTTMKB2 Note 6	INTST3/INTCSI30/ INTIIC30Note 9	INTIT31
ELISEL3	INTOSDC	INTSRE3 Note 10	INTTRJ1Note 6	INTSR3/INTCSI31/ INTIIC31Note 11	INTIT71 Note 6
ELISEL4	INTKR	INTUT2Note 8		INTUT0	INTEXSD
ELISEL5	INTIT00	INTUR2Note 8		INTUR0	No connection

Note 4. The interrupt sources that are selectable as event sources for INTC4 to INTC9 depend on which of the ELISELn (n = 6 to 11) registers is being set. Use the interrupt request signals as the hardware triggers for event-receiving peripheral functions.

Register	Event Source								
	INTC4	INTC5		INTC6		INTC7		INTC8	
ELISEL6	INTP0	INTTM00		INTTM06	INTIT50 Note 6	INTST2/INTCSI20/ INTIIC20Note 12	INTIT01	INTSR1/INTCSI11/ INTIIC11	INTIT61 Note 6
ELISEL7	INTP1	INTTM01		INTITL		INTSR2/INTCSI21/ INTIIC21	INTIT41 Note 6	INTSRE1 03H	INTIT40 Note 6
ELISEL8	INTP2	INTTM02		INTWDTI		INTSRE2	INTIT20	INTTRJ0	
ELISEL9	INTP3	INTTM03		INTRTC		INTST0/INTCSI00/ INTIIC00	INTIT21	INTSR0/INTCSI01/ INTIIC01	INTIT51 Note 6
ELISEL10	INTP4	INTTM04		INTTM07	INTIT60 Note 6	INTSRE0	INTTM01 H	INTIT30	INTLV1
ELISEL11	INTP5	INTTM05	INTURE3 Note 7	INTIICA0		INTST1/INTCSI10/ INTIIC10	INTIT11	INTAD	

Note 5. If the FRQSEL4 bit in the option bytes is set to 1, this cannot be used.

Note 6. This is only present in the 44- to 64-pin products with 256 and 512 Kbytes of flash memory, and in the 80- and 100-pin products.

Note 7. This is only present in the 80- and 100-pin products.

Note 8. This is only present in the 48- to 64-pin products with 256 and 512 Kbytes of flash memory, and in the 80- and 100-pin products.

Note 9. INTIIC30 is only present in the 44- and 48-pin products.

Note 10. This is only present in the 52- to 100-pin products.

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

<R>
<R>

Note 11. INTSR3, INTCSI31, and INTIIC31 are only present in the 64- to 100-pin products. INTSR3 is only present in the 52-pin products.

Note 12. INTST2 is only present in the 44-pin products.

Note 13. This is only present in the 52- to 100-pin products.

Caution 1. Be sure to set bits 7 to 5 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark fCLK: CPU/peripheral hardware clock frequency
fHP: High-speed on-chip oscillator peripheral clock frequency
fMP: Middle-speed on-chip oscillator peripheral clock frequency
fSXP: Low-speed peripheral clock frequency

24.3.2 Event link L1 signal select registers n (ELL1SELn) (n = 0 to 3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits), selector, and flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 6 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 0 to 3) (1/2)

Address: F0690H (ELL1SEL0) to F0693H (ELL1SEL3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SELn	ELL1SEL nZ	0	0	0	ELL1SEL n3	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0

ELL1SEL nZ	Control of event signal output level						
0	Positive logic output						
1	Negative logic output (inverted level)						

ELL1SEL n3	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0	Selection of target signal to be linked
0	0	0	0	No selection (fixed to 0).
0	0	0	1	Signal selected by input signal select register 0 is to be linked.
0	0	1	0	Signal selected by input signal select register 1 is to be linked.
0	0	1	1	Signal selected by input signal select register 2 is to be linked.
0	1	0	0	Signal selected by input signal select register 3 is to be linked.
0	1	0	1	Signal selected by input signal select register 4 is to be linked.
0	1	1	0	Signal selected by input signal select register 5 is to be linked.
0	1	1	1	Signal selected by input signal select register 6 is to be linked.
1	0	0	0	Signal selected by input signal select register 7 is to be linked.
1	0	0	1	Signal selected by input signal select register 8 is to be linked.
1	0	1	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	1	Signal selected by input signal select register 10 is to be linked.

Figure 24 - 6 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 0 to 3) (2/2)

ELL1SEL n3	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0	Selection of target signal to be linked
1	1	0	0	Signal selected by input signal select register 11 is to be linked.
Others				Setting prohibited

Caution 1. Be sure to set bits 6 to 4 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.3 Event link L1 signal select registers n (ELL1SELn) (n = 4, 5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 7 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 4, 5)

Address: F0694H (ELL1SEL4), F0695H (ELL1SEL5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SELn	ELL1SEL nZ	0	0	0	0	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0

ELL1SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL1SEL n2	ELL1SEL n1	ELL1SEL n0	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.4 Event link L1 signal select register 6 (ELL1SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 8 Format of Event Link L1 Signal Select Register 6 (ELL1SEL6)

Address: F0696H
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SEL6	ELL1SEL 6Z	0	0	0	0	ELL1SEL 62	ELL1SEL 61	ELL1SEL 60

ELL1SEL 6Z	Control of event signal output level		
0	Positive logic output		
1	Negative logic output (inverted level)		

ELL1SEL 62	ELL1SEL 61	ELL1SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.5 Logic cell block L1 control register (ELL1CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L1. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L1. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 9 Format of Logic Cell Block L1 Control Register (ELL1CTL)

Address: F0697H (ELL1CTL)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1CTL	ELL1T 31EN	ELL1T 30EN	0	0	ELL1T 111	ELL1T 110	ELL1T 101	ELL1T 100
Selection of enabling or stopping flip-flop 1 in logic cell block L1								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of enabling or stopping flip-flop 0 in logic cell block L1								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of the logic circuit to be used in logic cell 1 of logic cell block L1								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								
Selection of the logic circuit to be used in logic cell 0 of logic cell block L1								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								

Caution 1. Be sure to set bits 5 and 4 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.6 Event link L1 output select registers n (ELL1LNKn) (n = 0 to 3)

These registers are used to specify the destinations where the event signals selected by event link L1 signal select registers n (ELL1SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit), the selector, or a flip-flop in logic cell block L1 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 10 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 0 to 3)

Address: F0698H (ELL1LNK0) to F069BH (ELL1LNK3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNKn	0	0	0	0	ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0

ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L1.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L1.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L1.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L1.
0	1	0	1	Linked to the selection control of the selector in logic cell block L1.
0	1	1	0	Linked to input 0 of the selector in logic cell block L1.
0	1	1	1	Linked to input 1 of the selector in logic cell block L1.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L1.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L1.
Others				Setting prohibited

Caution 1. Be sure to set bits 7 to 4 to 0.

Caution 2. Select different destinations for event link L1 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.7 Event link L1 output select registers n (ELL1LNKn) (n = 4, 5)

These registers are used to specify the destinations where the event signals selected by event link L1 signal select registers n (ELL1SELn) are to be linked. The signals can be linked to the set control or reset control of the flip-flops in logic cell block L1. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 11 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 4, 5)

Address: F069CH (ELL1LNK4), F069DH (ELL1LNK5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNKn	0	0	0	0	ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0
ELL1LNK n3	Control of the link to flip-flop in logic cell block L1							
0	No link.							
1	Linked to the reset control of flip-flop 1 in logic cell block L1.							
ELL1LNK n2	Control of the link to flip-flop in logic cell block L1							
0	No link.							
1	Linked to the set control of flip-flop 1 in logic cell block L1.							
ELL1LNK n1	Control of the link to flip-flop in logic cell block L1							
0	No link.							
1	Linked to the reset control of flip-flop 0 in logic cell block L1.							
ELL1LNK n0	Control of the link to flip-flop in logic cell block L1							
0	No link.							
1	Linked to the set control of flip-flop 0 in logic cell block L1.							

Caution 1. Be sure to set bits 7 to 4 to 0.

Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.8 Event link L1 output select register 6 (ELL1LNK6)

This register is used to specify the destinations where the event signal selected by event link L1 signal select register 6 (ELL1SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L1. This 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 Event Link L1 Output Select Register 6 (ELL1LNK6)

Address: F069EH (ELL1LNK6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNK6	0	0	0	0	0	0	ELL1LNK 61	ELL1LNK 60

ELL1LNK 61	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L1.

ELL1LNK 60	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L1.

Caution 1. Be sure to set bits 7 to 2 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.9 Event link L2 signal select registers n (ELL2SELn) (n = 0 to 3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits) and selector, and the input, set control, and reset control of the flip-flops in logic cell block L2. Choose one of the event signals selected by input signal select registers n (ELISELn). In addition, the signals output from logic cell block L1 can be selected as event signals. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 13 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 0 to 3) (1/2)

Address: F06A0H (ELL2SEL0) to F06A3H (ELL2SEL3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SELn	ELL2SEL nZ	0	0	ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0
Control of event signal output level								
0 Positive logic output								
1 Negative logic output (inverted level)								
Selection of target signal to be linked								
ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0				
0	0	0	0	0	No selection (fixed to 0).			
0	0	0	0	1	Signal selected by input signal select register 0 is to be linked.			
0	0	0	1	0	Signal selected by input signal select register 1 is to be linked.			
0	0	0	1	1	Signal selected by input signal select register 2 is to be linked.			
0	0	1	0	0	Signal selected by input signal select register 3 is to be linked.			
0	0	1	0	1	Signal selected by input signal select register 4 is to be linked.			
0	0	1	1	0	Signal selected by input signal select register 5 is to be linked.			
0	0	1	1	1	Signal selected by input signal select register 6 is to be linked.			
0	1	0	0	0	Signal selected by input signal select register 7 is to be linked.			
0	1	0	0	1	Signal selected by input signal select register 8 is to be linked.			
0	1	0	1	0	Signal selected by input signal select register 9 is to be linked.			

Figure 24 - 13 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 0 to 3) (2/2)

ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0	Selection of target signal to be linked
0	1	0	1	1	Signal selected by input signal select register 10 is to be linked.
0	1	1	0	0	Signal selected by input signal select register 11 is to be linked.
0	1	1	0	1	Output signal 0 from logic cell block L1 is to be linked.
0	1	1	1	0	Output signal 1 from logic cell block L1 is to be linked.
0	1	1	1	1	Output signal 2 from logic cell block L1 is to be linked.
1	0	0	0	0	Output signal 3 from logic cell block L1 is to be linked.
1	0	0	0	1	Output signal 4 from logic cell block L1 is to be linked.
Others					Setting prohibited

Caution 1. Be sure to set bits 6 and 5 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.10 Event link L2 signal select registers n (ELL2SELn) (n = 4, 5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L2. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 14 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 4, 5)

Address: F06A4H (ELL2SEL4), F06A5H (ELL2SEL5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SELn	ELL2SEL nZ	0	0	0	0	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0

ELL2SEL nZ	Control of event signal output level		
0	Positive logic output		
1	Negative logic output (inverted level)		

ELL2SEL n2	ELL2SEL n1	ELL2SEL n0	Selection of target signal to be linked
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
Others			Setting prohibited

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.11 Event link L2 signal select register 6 (ELL2SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L2. Specify an event signal from among those selected by input signal select registers n (ELISELn). This 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 Event Link L2 Signal Select Register 6 (ELL2SEL6)

Address: F06A6H (ELL2SEL6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SEL6	ELL2SEL 6Z	0	0	0	0	ELL2SEL 62	ELL2SEL 61	ELL2SEL 60

ELL2SEL 6Z	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL2SEL 62	ELL2SEL 61	ELL2SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 ^{Note} is to be linked.
0	1	0	Signal selected by input signal select register 7 ^{Note} is to be linked.
0	1	1	Signal selected by input signal select register 8 ^{Note} is to be linked.
1	0	0	Signal selected by input signal select register 9 ^{Note} is to be linked.
1	0	1	Signal selected by input signal select register 10 ^{Note} is to be linked.
1	1	0	Signal selected by input signal select register 11 ^{Note} is to be linked.
Others			Setting prohibited

Note Select a clock signal in the corresponding one of input signal select registers 6 to 11.

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.12 Logic cell block L2 control register (ELL2CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L2. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L2. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 16 Format of Logic Cell Block L2 Control Register (ELL2CTL)

Address: F06A7H (ELL2CTL)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2CTL	ELL2T 31EN	ELL2T 30EN	0	0	ELL2T 111	ELL2T 110	ELL2T 101	ELL2T 100
Selection of enabling or stopping flip-flop 1 in logic cell block L2								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of enabling or stopping flip-flop 0 in logic cell block L2								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of the logic circuit to be used in logic cell 1 of logic cell block L2								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								
Selection of the logic circuit to be used in logic cell 0 of logic cell block L2								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								

Caution 1. Be sure to set bits 5 and 4 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.13 Event link L2 output select registers n (ELL2LNKn) (n = 0 to 3)

These registers are used to specify the destinations where the event signals selected by event link L2 signal select registers n (ELL2SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit) or the selector, or the input, set control, or reset control of a flip-flop in logic cell block L2 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 17 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 0 to 3)

Address: F06A8H (ELL2LNK0) to F06ABH (ELL2LNK3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNKn	0	0	0	0	ELL2LNK n3	ELL2LNK n2	ELL2LNK n1	ELL2LNK n0

ELL2LNK n3	ELL2LNK n2	ELL2LNK n1	ELL2LNK n0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L2.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L2.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L2.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L2.
0	1	0	1	Linked to the selection control of the selector in logic cell block L2.
0	1	1	0	Linked to input 0 of the selector in logic cell block L2.
0	1	1	1	Linked to input 1 of the selector in logic cell block L2.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L2.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L2.
1	0	1	0	Linked to the set control of flip-flop 0 in logic cell block L2.
1	0	1	1	Linked to the reset control of flip-flop 0 in logic cell block L2.
Others				Setting prohibited

Caution 1. Be sure to set bits 7 to 4 to 0.

Caution 2. Select different destinations for event link L2 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.14 Event link L2 output select registers n (ELL2LNKn) (n = 4, 5)

These registers are used to specify the destinations where the event signals selected by event link L2 signal select registers n (ELL2SELn) are to be linked. The signals can be linked to the set control or reset control of a flip-flop in logic cell block L2. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 18 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 4, 5)

Address: F06ACH (ELL2LNK4), F06ADH (ELL2LNK5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNKn	0	0	0	0	0	0	0	ELL2LNK n0

<ELL2LNK5 register>

ELL2LNK 50	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the reset control of flip-flop 1 in logic cell block L2.

<ELL2LNK4 register>

ELL2LNK 40	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the set control of flip-flop 1 in logic cell block L2.

Caution 1. Be sure to set bits 7 to 1 to 0.

Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.15 Event link L2 output select register 6 (ELL2LNK6)

This register is used to specify the destinations where the event signal selected by event link L2 signal select register 6 (ELL2SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L2. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 19 Format of Event Link L2 Output Select Register 6 (ELL2LNK6)

Address: F06AEH (ELL2LNK6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNK6	0	0	0	0	0	0	ELL2LNK 61	ELL2LNK 60

ELL2LNK 61	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L2.

ELL2LNK 60	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L2.

Caution 1. Be sure to set bits 7 to 2 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.16 Event link L3 signal select registers n (ELL3SELn) (n = 0 to 3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits) and selector, and the input, set control, and reset control of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). In addition, the signals output from logic cell block L2 can be selected as event signals. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 20 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 0 to 3) (1/2)

Address: F06B0H (ELL3SEL0) to F06B3H (ELL3SEL3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SELn	ELL3SEL nZ	0	0	ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0
	ELL3SEL nZ	Control of event signal output level						
	0	Positive logic output						
	1	Negative logic output (inverted level)						
	ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked		
	0	0	0	0	0	No selection (fixed to 0).		
	0	0	0	0	1	Signal selected by input signal select register 0 is to be linked.		
	0	0	0	1	0	Signal selected by input signal select register 1 is to be linked.		
	0	0	0	1	1	Signal selected by input signal select register 2 is to be linked.		
	0	0	1	0	0	Signal selected by input signal select register 3 is to be linked.		
	0	0	1	0	1	Signal selected by input signal select register 4 is to be linked.		
	0	0	1	1	0	Signal selected by input signal select register 5 is to be linked.		
	0	0	1	1	1	Signal selected by input signal select register 6 is to be linked.		
	0	1	0	0	0	Signal selected by input signal select register 7 is to be linked.		
	0	1	0	0	1	Signal selected by input signal select register 8 is to be linked.		
	0	1	0	1	0	Signal selected by input signal select register 9 is to be linked.		

Figure 24 - 20 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 0 to 3) (2/2)

ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked
0	1	0	1	1	Signal selected by input signal select register 10 is to be linked.
0	1	1	0	0	Signal selected by input signal select register 11 is to be linked.
0	1	1	0	1	Output signal 0 from logic cell block L2 is to be linked.
0	1	1	1	0	Output signal 1 from logic cell block L2 is to be linked.
0	1	1	1	1	Output signal 2 from logic cell block L2 is to be linked.
1	0	0	0	0	Output signal 3 from logic cell block L2 is to be linked.
1	0	0	0	1	Output signal 4 from logic cell block L2 is to be linked.
Others					Setting prohibited

Caution 1. Be sure to set bits 6 and 5 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.17 Event link L3 signal select registers n (ELL3SELn) (n = 4, 5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 21 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 4, 5)

Address: F06B4H (ELL3SEL4), F06B5H (ELL3SEL5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SELn	ELL3SEL nZ	0	0	0	0	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0

ELL3SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.18 Event link L3 signal select register 6 (ELL3SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 22 Format of Event Link L3 Signal Select Register 6 (ELL3SEL6)

Address: F06B6H (ELL3SEL6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SEL6	ELL3SEL 6Z	0	0	0	0	ELL3SEL 62	ELL3SEL 61	ELL3SEL 60

ELL3SEL 6Z	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL3SEL 62	ELL3SEL 61	ELL3SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 ^{Note} is to be linked.
0	1	0	Signal selected by input signal select register 7 ^{Note} is to be linked.
0	1	1	Signal selected by input signal select register 8 ^{Note} is to be linked.
1	0	0	Signal selected by input signal select register 9 ^{Note} is to be linked.
1	0	1	Signal selected by input signal select register 10 ^{Note} is to be linked.
1	1	0	Signal selected by input signal select register 11 ^{Note} is to be linked.
1	1	1	No selection (fixed to 0).

Note Select a clock signal in the corresponding one of input signal select registers 6 to 11.

Caution 1. Be sure to set bits 6 to 3 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.19 Logic cell block L3 control register (ELL3CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L3. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L3. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 23 Format of Logic Cell Block L3 Control Register (ELL3CTL)

Address: F06B7H

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3CTL	ELL3T 31EN	ELL3T 30EN	0	0	ELL3T 111	ELL3T 110	ELL3T 101	ELL3T 100
Selection of enabling or stopping flip-flop 1 in logic cell block L3								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of enabling or stopping flip-flop 0 in logic cell block L3								
0 The flip-flop is stopped (the flip-flop is in the reset state).								
1 The flip-flop is enabled.								
Selection of the logic circuit to be used in logic cell 1 of logic cell block L3								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								
Selection of the logic circuit to be used in logic cell 0 of logic cell block L3								
0 0 Pass-through is selected.								
0 1 AND circuit is selected.								
1 0 OR circuit is selected.								
1 1 EX-OR circuit is selected.								

Caution 1. Be sure to set bits 5 and 4 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.20 Event link L3 output select registers n (ELL3LNKn) (n = 0 to 3)

These registers are used to specify the destinations where the event signals selected by event link L3 signal select registers n (ELL3SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit) or the selector, or the input, set control, or reset control of a flip-flop in logic cell block L3 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 24 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 0 to 3)

Address: F06B8H (ELL3LNK0) to F06BBH (ELL3LNK3)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNKn	0	0	0	0	ELL3LNK n3	ELL3LNK n2	ELL3LNK n1	ELL3LNK n0

ELL3LNK n3	ELL3LNK n2	ELL3LNK n1	ELL3LNK n0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L3.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L3.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L3.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L3.
0	1	0	1	Linked to the selection control of the selector in logic cell block L3.
0	1	1	0	Linked to input 0 of the selector in logic cell block L3.
0	1	1	1	Linked to input 1 of the selector in logic cell block L3.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L3.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L3.
1	0	1	0	Linked to the set control of flip-flop 0 in logic cell block L3.
1	0	1	1	Linked to the reset control of flip-flop 0 in logic cell block L3.
Others				Setting prohibited

Caution 1. Be sure to set bits 7 to 4 to 0.

Caution 2. Select different destinations for event link L3 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.21 Event link L3 output select registers n (ELL3LNKn) (n = 4, 5)

These registers are used to specify the destinations where the event signals selected by event link L3 signal select registers n (ELL3SELn) are to be linked. The signals can be linked to the set control or reset control of a flip-flop in logic cell block L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 25 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 4, 5)

Address: F06BCH (ELL3LNK4), F06BDH (ELL3LNK5)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNKn	0	0	0	0	0	0	0	ELL3LNK n0

<ELL3LNK5 register>

ELL3LNK 50	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the reset control of flip-flop 1 in logic cell block L3.

<ELL3LNK4 register>

ELL3LNK 40	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the set control of flip-flop 1 in logic cell block L3.

Caution 1. Be sure to set bits 7 to 1 to 0.

Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.22 Event link L3 output select register 6 (ELL3LNK6)

This register is used to specify the destinations where the event signal selected by event link L3 signal select register 6 (ELL3SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L3. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 26 Format of Event Link L3 Output Select Register 6 (ELL3LNK6)

Address: F06BEH (ELL3LNK6)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNK6	0	0	0	0	0	0	ELL3LNK 61	ELL3LNK 60

ELL3LNK 61	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L3.

ELL3LNK 60	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L3.

Caution 1. Be sure to set bits 7 to 2 to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

24.3.23 Output signal select registers n (ELOSELn) (n = 0 to 7)

These registers are used to select signals to be output to the event-receiving peripheral functions. Select a signal from among those output from logic cell blocks L1 to L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 24 - 27 Format of Output Signal Select Registers n (ELOSELn) (1/2)

Address: F06C0H (ELOSEL0) to F06C7H (ELOSEL7)

After reset: 00H

R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELOSELn	ELOSELnZ	0	0	0	ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0

ELOSELnZ	Control of output signal level			
0	Positive logic output			
1	Negative logic output (inverted level)			

ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0	Selection of output signal
0	0	0	0	No selection.
0	0	0	1	Output signal 0 from logic cell block L1 (output of logic cell 0) is selected.
0	0	1	0	Output signal 1 from logic cell block L1 (output of logic cell 1) is selected.
0	0	1	1	Output signal 2 from logic cell block L1 (output of selector) is selected.
0	1	0	0	Output signal 3 from logic cell block L1 (output of flip-flop 0) is selected.
0	1	0	1	Output signal 4 from logic cell block L1 (output of flip-flop 1) is selected.
0	1	1	0	Output signal 0 from logic cell block L2 (output of logic cell 0) is selected.
0	1	1	1	Output signal 1 from logic cell block L2 (output of logic cell 1) is selected.
1	0	0	0	Output signal 2 from logic cell block L2 (output of selector) is selected.
1	0	0	1	Output signal 3 from logic cell block L2 (output of flip-flop 0) is selected.
1	0	1	0	Output signal 4 from logic cell block L2 (output of flip-flop 1) is selected.
1	0	1	1	Output signal 0 from logic cell block L3 (output of logic cell 0) is selected.
1	1	0	0	Output signal 1 from logic cell block L3 (output of logic cell 1) is selected.
1	1	0	1	Output signal 2 from logic cell block L3 (output of selector) is selected.

Figure 24 - 27 Format of Output Signal Select Registers n (ELOSELn) (2/2)

ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0	Selection of output signal
1	1	1	0	Output signal 3 from logic cell block L3 (output of flip-flop 0) is selected.
1	1	1	1	Output signal 4 from logic cell block L3 (output of flip-flop 1) is selected.

Caution Be sure to set bits 6 to 4 to 0.

24.3.24 Output signal enable register (ELOENCTL)

This register enables or disables the output of the signals selected by the ELOSELn registers. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 28 Format of Output Signal Enable Register (ELOENCTL)

Address: F06C8H (ELOENCTL)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELOENCTL	ELOEN CTL7	ELOEN CTL6	ELOEN CTL5	ELOEN CTL4	ELOEN CTL3	ELOEN CTL2	ELOEN CTL1	ELOEN CTL0
	Enable or disable of the output of signal [n]							
	0	Output is disabled.						
	1	Output is enabled. Note						

Note Only make the setting to enable the output of a signal after confirming that the event source signal is in a state which does not affect the operation of the peripheral function at the event destination.

Table 24 - 7 shows the correspondence between the bits in the output signal enable register and the event-receiving peripheral functions. Each signal output from the ELCL is connected to multiple destinations. When a destination does not use the signal from the ELCL, make settings in the destination so that the signal from the ELCL is not selected as an input signal for the destination. For the selection of input signals, refer to the section of each destination peripheral function.

Table 24 - 7 Bits in Output Signal Enable Register and Event-receiving Peripheral Functions

Register and Bit Name	Event-Receiving Peripheral Function					
	Destination 1	Destination 2	Destination 3	Destination 4	Destination 5	Destination 6
ELOENCTL. ELOENCTL0	—	—	Event link interrupt (INTELCL) Note 1	Output pin (P00/EO00)	Source for counting by timer RJ0	—
ELOENCTL. ELOENCTL1	—	Output pin (P60/EO60)	DTC activation trigger Note 1	—	Source for counting by timer RJ1 Note 2	—
ELOENCTL. ELOENCTL2	Serial clock input to SAU0 channel 0	Output pin (P61/EO61)	SMS activation trigger Note 1	—	—	—
ELOENCTL. ELOENCTL3	Serial clock input to SAU0 channel 1	Timer input to TAU0 channel 0 Note 3	—	Output pin (P76/EO76)	—	—
ELOENCTL. ELOENCTL4	Receive data input to SAU0 channel 0	Timer input to TAU0 channel 1 Note 3	—	Output pin (P77/EO77)	—	—
ELOENCTL. ELOENCTL5	Receive data input to SAU0 channel 1	—	—	Output pin (P05/EO05)	Hardware trigger for A/D converter Note 1	Clock input to UARTA (UARTA0 to UARTA3) Note 4
ELOENCTL. ELOENCTL6	—	Timer input to TAU0 channel 5 Note 3	—	—	Hardware trigger for D/A converter 0 Note 1	Hardware trigger for CTSU Note 1
ELOENCTL. ELOENCTL7	—	—	—	Output pin (P17/EO17)	Hardware trigger for D/A converter 1 Note 1	TML32 operating clock, capture trigger Note 1

Note 1. In hardware trigger operation, the rising edge is detected as a trigger.

Note 2. This is only supported in the 44- to 64-pin products with 256 to 512 Kbytes of flash memory, and in the 80- and 100-pin products.

Note 3. The widths at high and low level of the timer array unit to be linked should be at least $1/f_{MCK} + 10$ ns. Accordingly, make the following settings to select an operating clock with a frequency which matches that requirement: use timer clock select register 0 (TPS0) to select the operating clock, and use noise filter enable register 1 (NFEN1) to enable or disable the noise filter.

Note 4. Only UARTA0 and UARTA1 are supported in the 44-pin products, and in the 48- to 64-pin products with 64 to 128 Kbytes of flash memory.
Only UARTA0, UARTA1, and UARTA2 are supported in the 48- to 64-pin products with 256 to 512 Kbytes of flash memory.
UARTA0, UARTA1, UARTA2, and UARTA3 are all supported in the 80- and 100-pin products.

24.3.25 Output signal monitor register (ELOMONI)

This register is used to monitor output signals n selected by the ELOSELn registers (n = 0 to 7). Each bit returns the level of the corresponding output signal when this register is read. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 24 - 29 Format of Output Signal Monitor Register (ELOMONI)

Address: F06C9H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ELOMONI	ELOMONI7	ELOMONI6	ELOMONI5	ELOMONI4	ELOMONI3	ELOMONI2	ELOMONI1	ELOMONI0
ELOMONIn	Level of output signal n							
0	The signal level is 0.							
1	The signal level is 1.							

Caution Depending on the timing with which this register is read, the value read for output signal n may not be correct. If the value read is incorrect, wait for enough time to confirm the stability of output signal n and then read the register again.

24.3.26 Registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller

Set the following registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller.

- Port mode registers (PMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

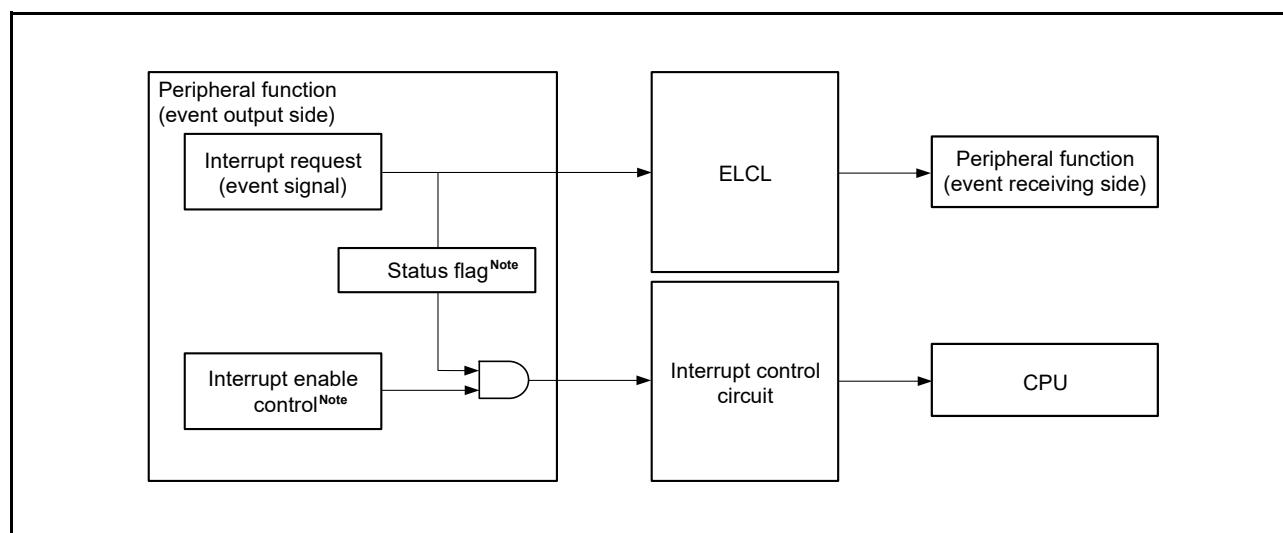
- **4.3.1 Port mode registers (PMxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

24.4 ELCL Operation

The ELCL inputs event signals from peripheral functions to logic cell blocks to convert them to signals that match the desired purpose before sending them to the event-receiving peripheral functions. A logic cell block has logic cells (pass-through, AND, OR, and EX-OR circuits), a selector, and flip-flops. The path for sending an event signal generated by a peripheral function to the interrupt control circuit as an interrupt request is separated from the path for using the signal as an ELCL event. Therefore, the interrupt function settings do not affect the use of an interrupt request signal as an ELCL event signal. For example, an interrupt request signal for which use as a trigger for an interrupt service routing has been disabled by using the corresponding bit of the interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H), can still be used as an ELCL event signal. **Figure 24 - 30** shows the relationship between interrupt service routing and the ELCL. The figure shows an example of a peripheral function that has an interrupt request status flag and an enable bit for controlling enabling and disabling of the interrupt. A peripheral function that receives an event from the ELCL performs the operation specific to the peripheral function after reception of an event.

Figure 24 - 30 Relationship between Interrupt Service Routing and ELCL

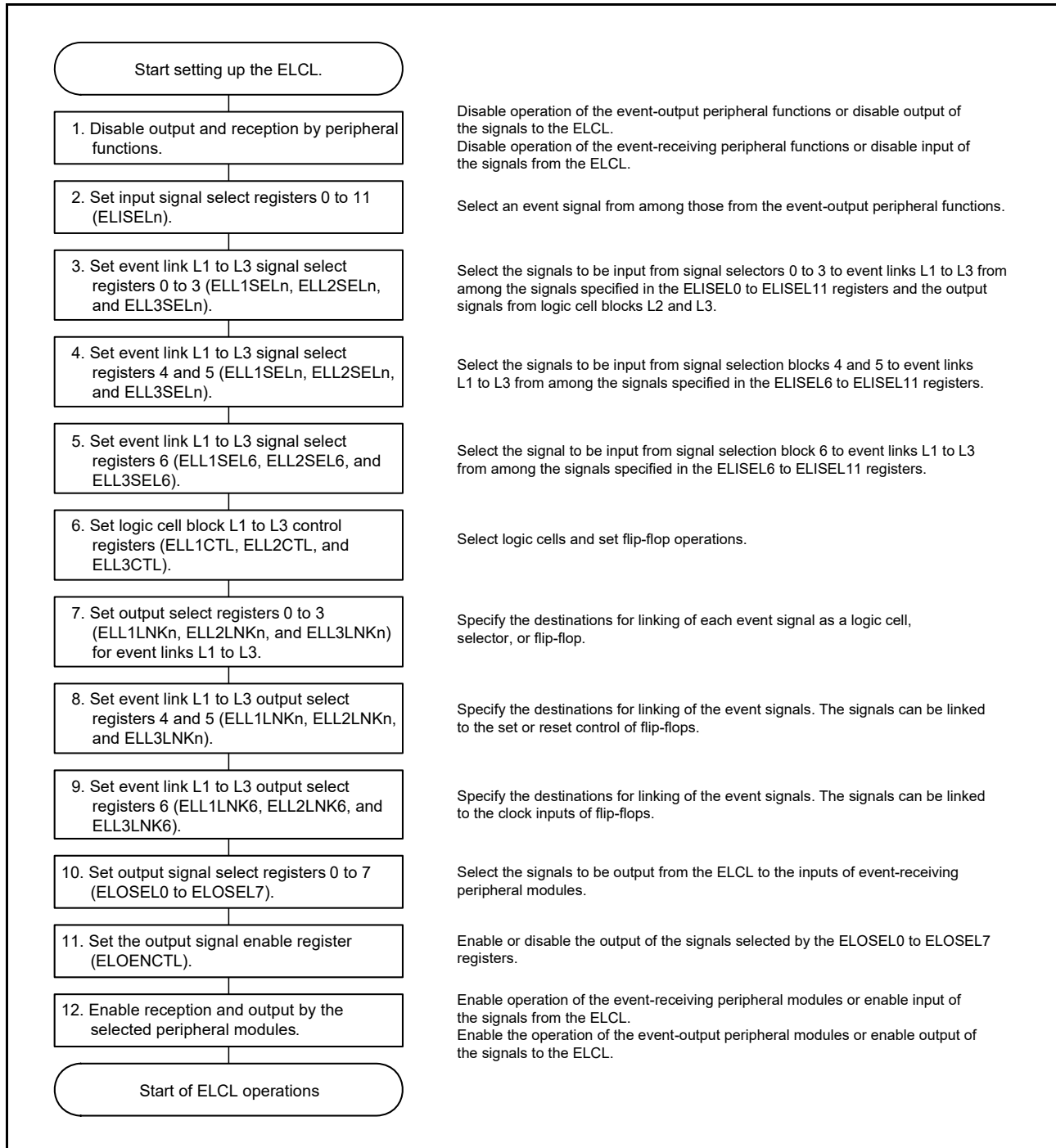


Note Some peripheral functions do not have status flags or interrupt enable control.

24.5 Procedure of Setting the ELCL

Figure 24 - 31 shows a procedure of setting the ELCL. Set up the registers in the ELCL while output by the event-output peripheral functions and reception by the event-receiving peripheral functions are disabled. After enabling output in step 11, enable reception by the event-receiving peripheral functions, and then enable output by the event-output peripheral functions.

Figure 24 - 31 Procedure of Setting the ELCL



1. Example of ELCL settings (OR circuit)

Figure 24 - 32 shows an OR circuit configuration as an example of the result of ELCL settings. **Table 24 - 8** is a list of the register settings for this example.

Figure 24 - 32 Example of ELCL Settings (OR Circuit)

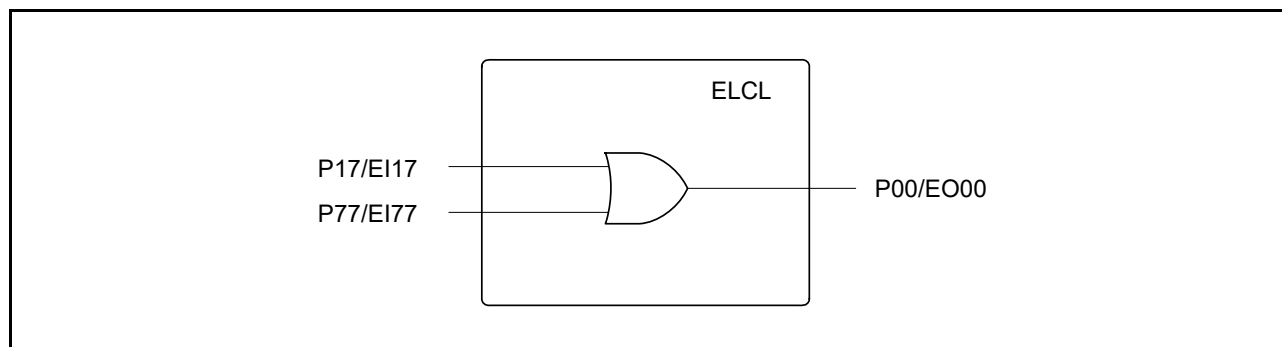


Table 24 - 8 Example of ELCL Register Settings (OR Circuit)

No.	Register Symbol	Register Name	Initial Value	Setting	Description
1	PORT.PMCE0	—	00H	00H	A signal other than the ELCL signal is selected for output from the P00 pin.
2	ELISEL0	Input signal select register 0	00H	05H	P17/EI17 is selected.
2	ELISEL1	Input signal select register 1	00H	10H	P77/EI77 is selected.
3	ELL1SEL0	Event link L1 signal select register 0	00H	01H	Input signal selector 0 is selected.
3	ELL1SEL1	Event link L1 signal select register 1	00H	02H	Input signal selector 1 is selected.
6	ELL1CTL	Logic cell block L1 control register	00H	02H	The OR circuit is selected in logic cell 0.
9	ELL1LNK0	Event link L1 output select register 0	00H	01H	Linked to 0 of logic cell block L1.
9	ELL1LNK1	Event link L1 output select register 1	00H	02H	Linked to 1 of logic cell block L1.
10	ELOSEL0	Output signal select register 0	00H	01H	Output signal [0] from logic cell block L1 is selected.
11	ELOENCTL	Output signal enable register	00H	01H	Output [0] is enabled. The signal is output to P00/EO00.
12	PORT.PMCE0	—	00H	01H	The signal from the ELCL is selected for the output from the P00 pin.

Caution 1. The numbers in the No. column correspond to the step numbers (1 to 12) in Figure 24 - 31 Procedure of Setting the ELCL.

Caution 2. Steps 2 to 10 can be done in any order, but modifying the register values specified in steps 2 to 10 is prohibited after the output signal enable register is set up in step 11.

2. Example of ELCL settings (AND circuit)

Figure 24 - 33 shows an AND circuit configuration as an example of the result of ELCL settings. **Table 24 - 9** is a list of the register settings for this example.

Figure 24 - 33 Example of ELCL Settings (AND Circuit)

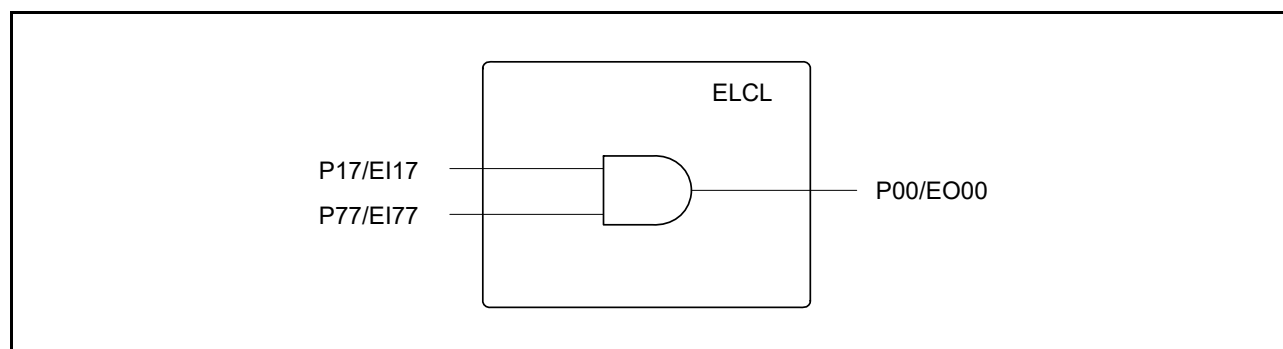


Table 24 - 9 Example of ELCL Register Settings (AND Circuit)

No.	Register Symbol	Register Name	Initial Value	Setting	Description
1	PORT.PMCE0	—	00H	00H	A signal other than the ELCL signal is selected for output from the P00 pin.
2	ELISEL0	Input signal select register 0	00H	05H	P17/EI17 is selected.
2	ELISEL1	Input signal select register 1	00H	10H	P77/EI77 is selected.
3	ELL1SEL0	Event link L1 signal select register 0	00H	01H	Input signal selector 0 is selected.
3	ELL1SEL1	Event link L1 signal select register 1	00H	02H	Input signal selector 1 is selected.
6	ELL1CTL	Logic cell block L1 control register	00H	01H	The AND circuit is selected in logic cell 0.
9	ELL1LNK0	Event link L1 output select register 0	00H	01H	Linked to 0 of logic cell block L1.
9	ELL1LNK1	Event link L1 output select register 1	00H	02H	Linked to 1 of logic cell block L1.
10	ELOSEL0	Output signal select register 0	00H	01H	Output signal [0] from logic cell block L1 is selected.
11	ELOENCTL	Output signal enable register	00H	01H	Output [0] is enabled. The signal is output to P00/EO00.
12	PORT.PMCE0	—	00H	01H	The signal from the ELCL is selected for the output from the P00 pin.

Caution 1. The numbers in the No. column correspond to the step numbers (1 to 12) in Figure 24 - 31 Procedure of Setting the ELCL.

Caution 2. Steps 2 to 10 can be done in any order, but modifying the register values specified in steps 2 to 10 is prohibited after the output signal enable register is set up in step 11.

24.6 Points for Caution when the ELCL is to be Used

1. When the SO00, SO01, TxD0, SCK00, SCK01, SCL00, SCL01, TxDA0, CLKA0, or TO00 to TO07 is to be selected as an input to the ELCL but not for output to a pin, set the corresponding PFOEmn (m = 0, 1, n = 0 to 7) bit in the PFOEm register to 0. For details of the setting, see **Section 4 Port Functions**.
2. Each signal output from the ELCL is linked to multiple peripheral functions as shown in **Table 24 - 7**. When a destination peripheral function is not to use a signal output from the ELCL, make settings for the destination peripheral function so that the signal from the ELCL is not selected as an input signal at the destination. For details on how to select the signal from the ELCL, refer to the sections on each of the destination peripheral functions.
3. Do not change the values of the following registers after output of signals is enabled in the output signal enable register (ELOENCTL).
 - Input signal select registers n (ELISELn) (n = 0 to 11)
 - Event link L1 signal select registers n (ELL1SELn) (n = 0 to 3)
 - Event link L1 signal select registers n (ELL1SELn) (n = 4, 5)
 - Event link L1 signal select register 6 (ELL1SEL6)
 - Logic cell block L1 control register (ELL1CTL)
 - Event link L1 output select registers n (ELL1LNKn) (n = 0 to 3)
 - Event link L1 output select registers n (ELL1LNKn) (n = 4, 5)
 - Event link L1 output select register 6 (ELL1LNK6)
 - Event link L2 signal select registers n (ELL2SELn) (n = 0 to 3)
 - Event link L2 signal select registers n (ELL2SELn) (n = 4, 5)
 - Event link L2 signal select register 6 (ELL2SEL6)
 - Logic cell block L2 control register (ELL2CTL)
 - Event link L2 output select registers n (ELL2LNKn) (n = 0 to 3)
 - Event link L2 output select registers n (ELL2LNKn) (n = 4, 5)
 - Event link L2 output select register 6 (ELL2LNK6)
 - Event link L3 signal select registers n (ELL3SELn) (n = 0 to 3)
 - Event link L3 signal select registers n (ELL3SELn) (n = 4, 5)
 - Event link L3 signal select register 6 (ELL3SEL6)
 - Logic cell block L3 control register (ELL3CTL)
 - Event link L3 output select registers n (ELL3LNKn) (n = 0 to 3)
 - Event link L3 output select registers n (ELL3LNKn) (n = 4, 5)
 - Event link L3 output select register 6 (ELL3LNK6)
 - Output signal select registers n (ELOSELn) (n = 0 to 7)
4. The ELCL outputs signals with the use of multiple input signals, logic cell blocks, and output controllers. The logic cell blocks L1, L2, and L3 produce outputs with some delay with respect to inputs. The output signals from the flip-flops are delayed by up to one cycle of the clock selected for synchronization of clock signals by the signal selection blocks 6 of event links L1 to L3. Furthermore, deviations in the timing of input signals may lead to the generation of glitches or expected outputs not being obtained. If an expected output not being obtained may create serious problems for a user system, stop attempting to use the ELCL or employ an external circuit as a workaround.

Section 25 Interrupt Functions

The interrupt function switches program execution to other processing at a branch destination. When the branch processing is finished, the program returns to the point at which the main processing was interrupted. The number of interrupt sources depends on the product as stated in the table below.

		44-pin	48-pin	52- and 64-pin	80- and 100-pin
Maskable interrupts	External	9	9	9	9
	Internal	47	49	50	52

25.1 Types of Interrupt

This product has the following two general types of interrupt.

1. Maskable interrupts

These interrupts are subject to 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, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H). Multiple (nested) interrupt servicing can be applied when high-priority interrupts are generated during the processing of low-priority interrupts. If two or more interrupt requests having the same priority are generated at the same time, then they are processed according to the default order of priority for vectored interrupt servicing. For the default order of priority, see **Table 25 - 1**. A standby release signal is generated and release from STOP, HALT, or SNOOZE mode proceeds. 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 is not subject to interrupt priority control.

25.2 Interrupt Sources and Configuration

This product has two general types of interrupt, specifically maskable interrupts and the software interrupt. It also has up to seven sources for reset (see **Table 25 - 1**). The size of the vector codes that indicate the program start address when a branch is caused by a reset or any of the various interrupt requests is two bytes each, so the CPU jumps to an address within the 64-Kbyte address space from 00000H to 0FFFFH.

Table 25 - 1 Interrupt Source List (1/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Name	Trigger									
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/4fIL)	Internal	00004H	<a>	✓	✓	✓	✓	✓	✓
	1	INTLVI	Voltage detection Note 4		00006H		✓	✓	✓	✓	✓	✓
	2	INTP0	Pin input edge detection	External	00008H		✓	✓	✓	✓	✓	✓
	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>	✓	✓	✓	✓	✓	Note 5
		INTIT01	Compare match in 8-bit interval timer channel 01				✓	✓	✓	✓	✓	
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		✓	✓	✓	✓	✓	
		INTIT41	Compare match in 8-bit interval timer channel 41				✓	✓	Note 6	Note 6	Note 6	Note 6
	10	INTSRE2	UART2 reception communication error occurrence		00018H		✓	✓	✓	✓	✓	✓
		INTIT20	Compare match in 8-bit interval timer channel 20 (at 8-bit/16- bit timer operation)				✓	✓	✓	✓	✓	✓
	11	INTELCL	Event link interrupt		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		✓	✓	✓	✓	✓	✓
		INTIT21	Compare match in 8-bit interval timer channel 21				✓	✓	✓	✓	✓	✓
	14	INTTM00	End of timer channel 00 count or capture		00020H		✓	✓	✓	✓	✓	✓
	15	INTSRE0	UART0 reception communication error occurrence		00022H		✓	✓	✓	✓	✓	✓
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓
		INTIT30	Compare match in 8-bit interval timer channel 30 (at 8-bit/16- bit timer operation)				✓	✓	✓	✓	✓	✓
	16	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00024H		✓	✓	✓	✓	✓	✓
		INTIT11	Compare match in 8-bit interval timer channel 11				✓	✓	✓	✓	✓	✓
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end		00026H		✓	✓	✓	✓	✓	✓
		INTIT61	Compare match in 8-bit interval timer channel 61				✓	✓	Note 6	Note 6	Note 6	Note 6

Table 25 - 1 Interrupt Source List (2/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Name	Trigger									
Maskable	18	INTSRE1	UART1 reception communication error occurrence	Internal	00028H	<a>	✓	✓	✓	✓	✓	✓
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓
		INTIT40	Compare match in 8-bit interval timer channel 40 (at 8-bit/16-bit timer operation)				✓	✓	Note 6	Note 6	Note 6	Note 6
	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		✓	✓	✓	✓	✓	✓
		INTIT51	Compare match in 8-bit interval timer channel 51				✓	✓	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		✓	✓	✓	✓	✓	✓
	23	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)		00032H		✓	✓	✓	✓	✓	✓
	24	INTAD	End of A/D conversion		00034H		✓	✓	✓	✓	✓	✓
	25	INTRTC	Fixed-cycle signal of realtime clock/alarm match detection		00036H		✓	✓	✓	✓	✓	✓
	26	INTITL	Interval signal detection of 32-bit interval timer		00038H		✓	✓	✓	✓	✓	✓
	27	INTKR	Key return signal detection	External	0003AH	<c>	✓	✓	✓	✓	✓	✓
	28	INTST3/ INTCSI30/ INTIIC30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt/IIC30 transfer end	Internal	0003CH	<a>	✓	✓	✓	✓	Note 7	Note 7
		INTIT31	Compare match in 8-bit interval timer channel 31				✓	✓	✓	✓	✓	✓
	29	INTSR3/ INTCSI31/ INTIIC31	UART3 reception transfer end/CSI31 transfer end or buffer empty interrupt/IIC31 transfer end		0003EH		✓	✓	✓	Note 8	—	—
		INTIT71	Compare match in 8-bit interval timer channel 71				✓	✓	Note 6	Note 6	Note 6	Note 6
	30	INTIT00	Compare match in 8-bit interval timer channel 00 (at 8-bit/16-bit timer operation)		00040H		✓	✓	✓	✓	✓	✓
	31	INTTM04	End of timer channel 04 count or capture		00042H		✓	✓	✓	✓	✓	✓
	32	INTTM05	End of timer channel 05 count or capture		00044H		✓	✓	✓	✓	✓	✓
		INTURE3	UARTA3 reception communication error occurrence				✓	✓	—	—	—	—
	33	INTTM06	End of timer channel 06 count or capture		00046H		✓	✓	✓	✓	✓	✓
		INTIT50	Compare match in 8-bit interval timer channel 50 (at 8-bit/16-bit timer operation)				✓	✓	Note 6	Note 6	Note 6	Note 6
	34	INTTM07	End of timer channel 07 count or capture		00048H		✓	✓	✓	✓	✓	✓
		INTIT60	Compare match in 8-bit interval timer channel 60 (at 8-bit/16-bit timer operation)				✓	✓	Note 6	Note 6	Note 6	Note 6

Table 25 - 1 Interrupt Source List (3/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Name	Trigger									
Maskable	35	INTP6	Pin input edge detection	External	0004AH		✓	✓	✓	✓	✓	✓
	36	INTP7			0004CH		✓	✓	✓	✓	✓	✓
	37	INTIT10	Compare match in 8-bit interval timer channel 10 (at 8-bit/16-bit timer operation)	Internal	0004EH		✓	✓	✓	✓	✓	✓
	38	INTOSDC	Oscillation stop detection		00050H		✓	✓	✓	✓	✓	✓
	39	INTFL	Reserved		00052H		✓	✓	✓	✓	✓	✓
	40	INTCMP0	Comparator detection 0		00054H		✓	✓	✓	✓	✓	✓
	41	INTCMP1	Comparator detection 1		00056H		✓	✓	✓	✓	—	—
	42	INTURE0	UARTA0 reception communication error occurrence		00058H	<a>	✓	✓	✓	✓	✓	✓
		INTIT70	Compare match in 8-bit interval timer channel 70 (at 8-bit/16-bit timer operation)				✓	✓	Note 6	Note 6	Note 6	Note 6
	43	INTURE1	UARTA1 reception communication error occurrence		0005AH		✓	✓	✓	✓	✓	✓
		INTTMKB1	End of timer KB41 count				✓	✓	Note 6	Note 6	Note 6	Note 6
	44	INTURE2	UARTA2 reception communication error occurrence		0005CH		✓	✓	Note 6	Note 6	Note 6	—
		INTTMKB2	End of timer KB42 count				✓	✓	Note 6	Note 6	Note 6	Note 6
	45	INTSRE3	UART3 reception communication error occurrence		0005EH		✓	✓	✓	✓	—	—
		INTTRJ1	Timer RJ1 interrupt				✓	✓	Note 6	Note 6	Note 6	Note 6
	46	INTCTSUWR	Request to write to a configuration register of an individual CTSU channel		00060H		✓	✓	✓	✓	✓	✓
	47	INTIICA1	End of IICA1 communication		00062H		✓	✓	✓	✓	✓	✓
	48	INTCTSURD	Request to transfer data measured by the CTSU		00064H		✓	✓	✓	✓	✓	✓
	49	INTCTSUFN	CTSU measurement end		00066H		✓	✓	✓	✓	✓	✓
	50	INTTRJ0	Timer RJ0 interrupt		00068H		✓	✓	✓	✓	✓	✓
	51	INTUT0	UARTA0 transmission transfer end or buffer empty interrupt		0006AH		✓	✓	✓	✓	✓	✓
	52	INTUR0	UARTA0 reception transfer end		0006CH		✓	✓	✓	✓	✓	✓
	53	INTUT1	UARTA1 transmission transfer end or buffer empty interrupt		0006EH		✓	✓	✓	✓	✓	✓
	54	INTUR1	UARTA1 reception transfer end		00070H		✓	✓	✓	✓	✓	✓
	55	INTUT2	UARTA2 transmission transfer end or buffer empty interrupt		00072H		✓	✓	Note 6	Note 6	Note 6	—
	56	INTUR2	UARTA2 reception transfer end		00074H		✓	✓	Note 6	Note 6	Note 6	—
	57	INTUT3	UARTA3 transmission transfer end or buffer empty interrupt		00076H		✓	✓	—	—	—	—
	58	INTUR3	UARTA3 reception transfer end		00078H		✓	✓	—	—	—	—
	59	INTTMKB0	End of timer KB40 count		0007AH		✓	✓	✓	✓	✓	✓
	60	INTEXSD	External signal sampler edge detection		0007CH		✓	✓	✓	✓	✓	✓

Table 25 - 1 Interrupt Source List (4/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Name	Trigger									
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 25 - 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.

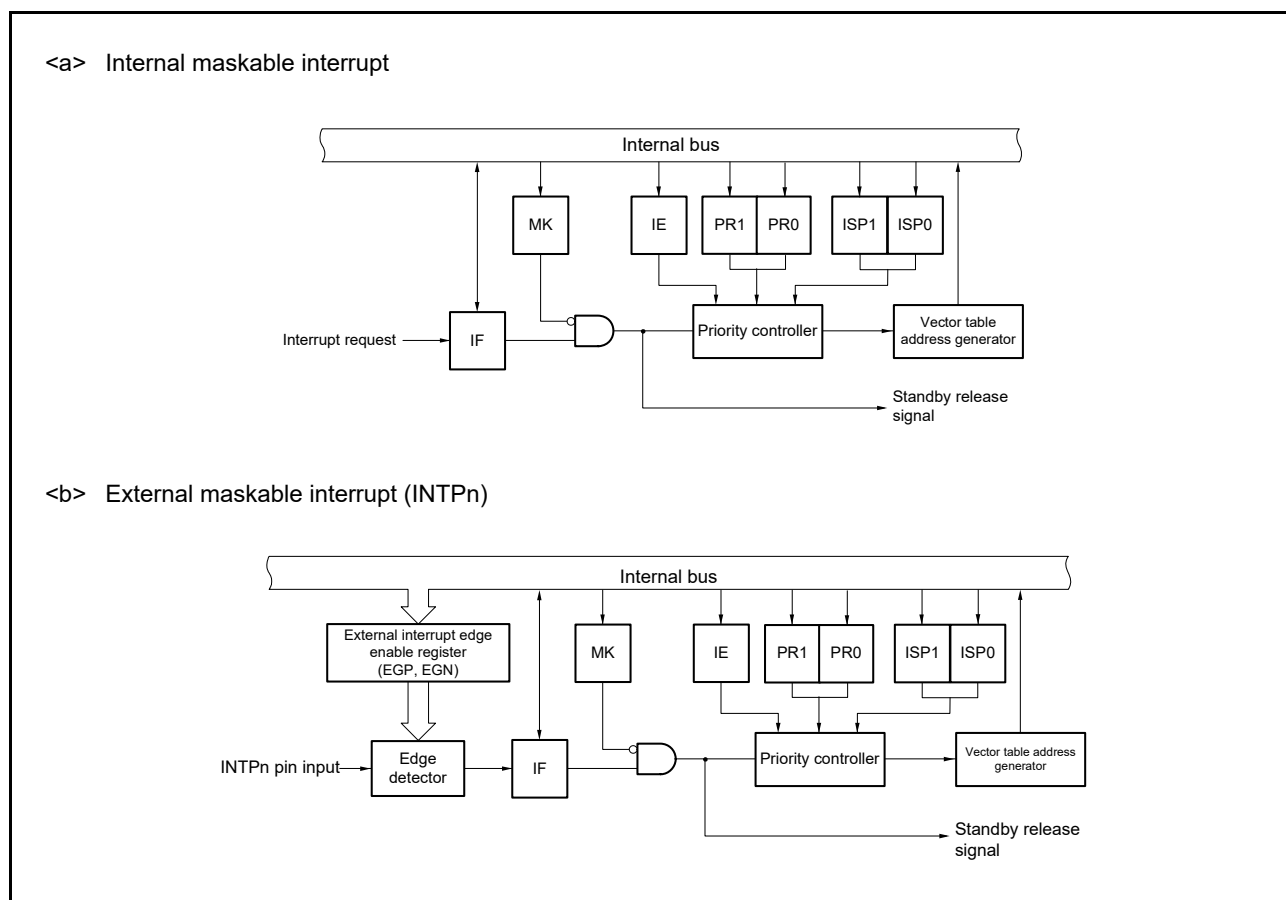
Note 5. Only INTST2 is present in this product.

Note 6. Only present in the products with 256 to 512 Kbytes of flash memory.

Note 7. Only INTIIC30 is present in this product.

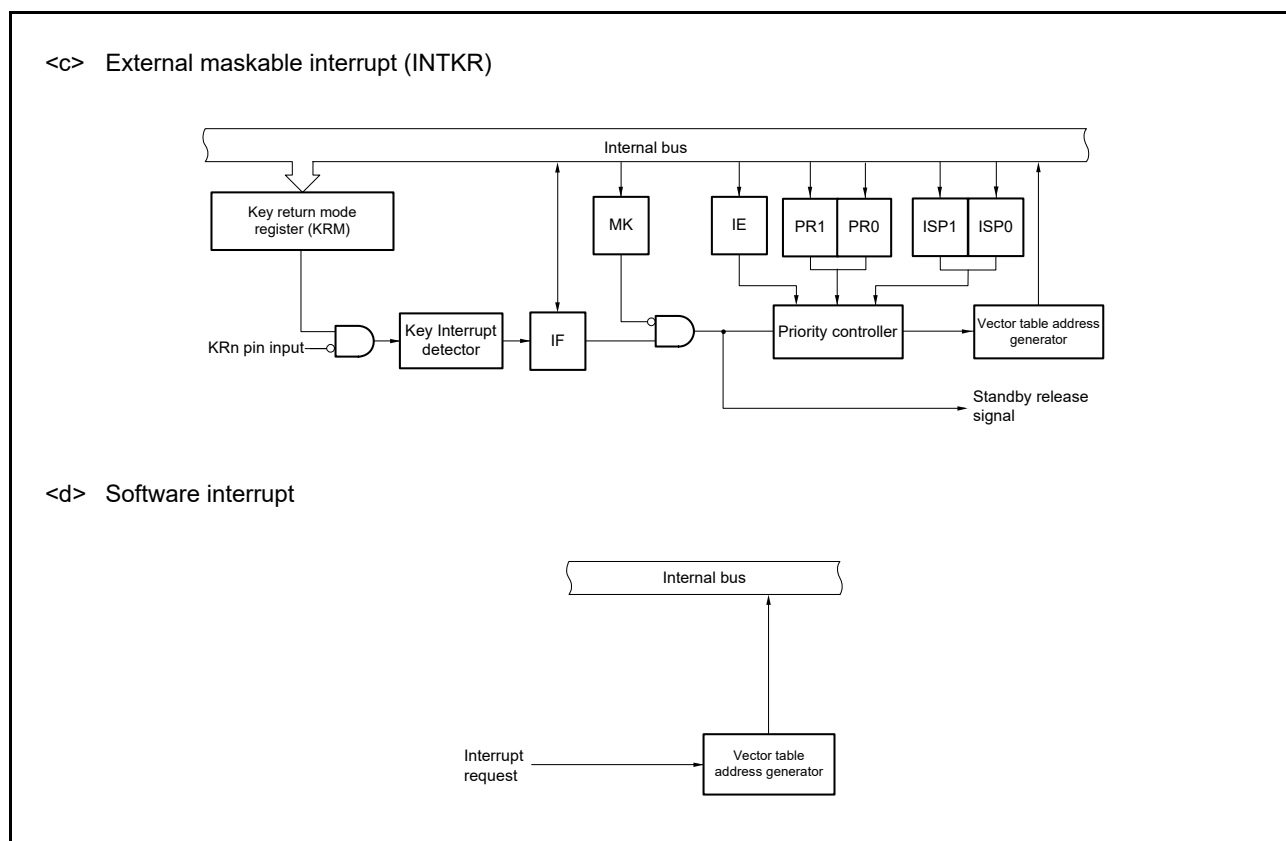
Note 8. Only INTSR3 is present in this product.

Figure 25 - 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.** n = 0 to 7

Figure 25 - 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. n = 0 to 7

25.3 Registers for Controlling the Interrupts

The following registers are used to control the interrupts.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)
- External interrupt rising edge enable register (EGP0), External interrupt falling edge enable register (EGN0)
- Program status word (PSW)
- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

Table 25 - 2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 25 - 2 Flags Corresponding to Interrupt Request Sources (1/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Register		Register		Register						
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓	✓	✓	✓	✓
INTLV ^{Note}	LVIIF ^{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 **30.3.1 Voltage detection register (LVIM)**.

Table 25 - 2 Flags Corresponding to Interrupt Request Sources (2/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		100-pin	80-pin	64-pin	52-pin	48-pin	44-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}		✓	✓	✓	✓	✓	—
INTIT01 ^{Note 1}	ITIF01 ^{Note 1}		ITMK01 ^{Note 1}		ITPR001, ITPR101 ^{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}		✓	✓	✓	✓	✓	✓
INTIT41 ^{Note 2}	ITIF41 ^{Note 2}		ITMK41 ^{Note 2}		ITPR041, ITPR141 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTSRE2 ^{Note 3}	SREIF2 ^{Note 3}		SREMK2 ^{Note 3}		SREPR02, SREPR12 ^{Note 3}		✓	✓	✓	✓	✓	✓
INTIT20 ^{Note 3}	ITIF20 ^{Note 3}		ITMK20 ^{Note 3}		ITPR020, ITPR120 ^{Note 3}		✓	✓	✓	✓	✓	✓
INTELCL	ELCLIF		ELCLMK		ELCLPR0, ELCLPR1		✓	✓	✓	✓	✓	✓
INTSMSE	SMSEIF		SMSEMK		SMSEPR0, SMSEPR1		✓	✓	✓	✓	✓	✓
INTST0 ^{Note 4}	STIF0 ^{Note 4}		STMK0 ^{Note 4}		STPR00, STPR10 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTCSI00 ^{Note 4}	CSIF00 ^{Note 4}		CSIMK00 ^{Note 4}		CSIPR000, CSIPR100 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTIIC00 ^{Note 4}	IICIF00 ^{Note 4}		IICMK00 ^{Note 4}		IICPR000, IICPR100 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTIT21 ^{Note 4}	ITIF21 ^{Note 4}		ITMK21 ^{Note 4}		ITPR021, ITPR121 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓	✓	✓	✓	✓
INTSRE0 ^{Note 5}	SREIF0 ^{Note 5}		SREMK0 ^{Note 5}		SREPR00, SREPR10 ^{Note 5}		✓	✓	✓	✓	✓	✓
INTTM01H ^{Note 5}	TMIF01H ^{Note 5}		TMMK01H ^{Note 5}		TMPR001H, TMPR101H ^{Note 5}		✓	✓	✓	✓	✓	✓
INTIT30 ^{Note 5}	ITIF30 ^{Note 5}		ITMK30 ^{Note 5}		ITPR030, ITPR130 ^{Note 5}		✓	✓	✓	✓	✓	✓

- Note 1.** If one of the interrupt sources INTST2, INTCSI20, INTIIC20, and INTIT01 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these four interrupt sources.
- Note 2.** If one of the interrupt sources INTSR2, INTCSI21, INTIIC21, and INTIT41 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these four interrupt sources.
- Note 3.** Do not use a UART2 reception error interrupt and an INTIT20 interrupt at the same time because they share flags for the interrupt request sources. If the UART2 reception error interrupt is not used (EOC01 = 0), UART2 and INTIT20 can be used at the same time. If one of the interrupt sources INTSRE2 and INTIT20 is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.
- Note 4.** If one of the interrupt sources INTST0, INTCSI00, INTIIC00, and INTIT21 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these four interrupt sources.
- Note 5.** Do not use a UART0 reception error interrupt, an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation), and an INTIT30 interrupt 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, channel 1 of TAU0 (at higher 8-bit timer operation), and INTIT30 can be used at the same time. If one of the interrupt sources INTSRE0, INTTM01H, and INTIT30 is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Table 25 - 2 Flags Corresponding to Interrupt Request Sources (3/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Register		Register		Register						
INTST1 ^{Note 1}	STIF1 ^{Note 1}	IF1L	STMK1 ^{Note 1}	MK1L	STPR01, STPR11 ^{Note 1}	PR01L, PR11L	✓	✓	✓	✓	✓	✓
INTCSI10 ^{Note 1}	CSIF10 ^{Note 1}		CSIMK10 ^{Note 1}		CSIPR010, CSIPR110 ^{Note 1}		✓	✓	✓	✓	✓	✓
INTIIC10 ^{Note 1}	IICIF10 ^{Note 1}		IICMK10 ^{Note 1}		IICPR010, IICPR110 ^{Note 1}		✓	✓	✓	✓	✓	✓
INTIT11 ^{Note 1}	ITIF11 ^{Note 1}		ITMK11 ^{Note 1}		ITPR011, ITPR111 ^{Note 1}		✓	✓	✓	✓	✓	✓
INTSR1 ^{Note 2}	SRIF1 ^{Note 2}		SRMK1 ^{Note 2}		SRPR01, SRPR11 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTCSI11 ^{Note 2}	CSIF11 ^{Note 2}		CSIMK11 ^{Note 2}		CSIPR011, CSIPR111 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTIIC11 ^{Note 2}	IICIF11 ^{Note 2}		IICMK11 ^{Note 2}		IICPR011, IICPR111 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTIT61 ^{Note 2}	ITIF61 ^{Note 2}		ITMK61 ^{Note 2}		ITPR061, ITPR161 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTSRE1 ^{Note 3}	SREIF1 ^{Note 3}		SREMK1 ^{Note 3}		SREPR01, SREPR11 ^{Note 3}		✓	✓	✓	✓	✓	✓
INTTM03H ^{Note 3}	TMIF03H ^{Note 3}		TMMK03H ^{Note 3}		TMPR003H, TMPR103H ^{Note 3}		✓	✓	✓	✓	✓	✓
INTIT40 ^{Note 3}	ITIF40 ^{Note 3}		ITMK40 ^{Note 3}		ITPR040, ITPR140 ^{Note 3}		✓	✓	✓	✓	✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓	✓	✓	✓	✓
INTSR0 ^{Note 4}	SRIF0 ^{Note 4}		SRMK0 ^{Note 4}		SRPR00, SRPR10 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTCSI01 ^{Note 4}	CSIF01 ^{Note 4}		CSIMK01 ^{Note 4}		CSIPR001, CSIPR101 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTIIC01 ^{Note 4}	IICIF01 ^{Note 4}		IICMK01 ^{Note 4}		IICPR001, IICPR101 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTIT51 ^{Note 4}	ITIF51 ^{Note 4}		ITMK51 ^{Note 4}		ITPR051, ITPR151 ^{Note 4}		✓	✓	✓	✓	✓	✓
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		✓	✓	✓	✓	✓	✓
INTITL	ITLIF		ITLMK		ITLPR0, ITPR1		✓	✓	✓	✓	✓	✓
INTKR	KRIF		KRMK		KRPR0, KRPR1		✓	✓	✓	✓	✓	✓
INTST3 ^{Note 5}	STIF3 ^{Note 5}		STMK3 ^{Note 5}		STPR03, STPR13 ^{Note 5}		✓	✓	✓	✓	—	—
INTCSI30 ^{Note 5}	CSIF30 ^{Note 5}		CSIMK30 ^{Note 5}		CSIPR030, CSIPR130 ^{Note 5}		✓	✓	✓	✓	—	—
INTIIC30 ^{Note 5}	IICIF30 ^{Note 5}		IICMK30 ^{Note 5}		IICPR030, IICPR130 ^{Note 5}		✓	✓	✓	✓	✓	✓
INTIT31 ^{Note 5}	ITIF31 ^{Note 5}		ITMK31 ^{Note 5}		ITPR031, ITPR131 ^{Note 5}		✓	✓	✓	✓	✓	✓
INTSR3 ^{Note 6}	SRIF3 ^{Note 6}		SRMK3 ^{Note 6}		SRPR03, SRPR13 ^{Note 6}		✓	✓	✓	✓	—	—
INTCSI31 ^{Note 6}	CSIF31 ^{Note 6}		CSIMK31 ^{Note 6}		CSIPR031, CSIPR131 ^{Note 6}		✓	✓	✓	—	—	—
INTIIC31 ^{Note 6}	IICIF31 ^{Note 6}		IICMK31 ^{Note 6}		IICPR031, IICPR131 ^{Note 6}		✓	✓	✓	—	—	—
INTIT71 ^{Note 6}	ITIF71 ^{Note 6}		ITMK71 ^{Note 6}		ITPR071, ITPR171 ^{Note 6}		✓	✓	✓	✓	✓	✓
INTIT00	ITIF00		ITMK00		ITPR000, ITPR100		✓	✓	✓	✓	✓	✓
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		✓	✓	✓	✓	✓	✓

Note 1. If one of the interrupt sources INTST1, INTCSI10, INTIIC10, and INTIT11 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these four interrupt sources.

Note 2. If one of the interrupt sources INTSR1, INTCSI11, INTIIC11, and INTIT61 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these four interrupt sources.

(Notes continue on the next page.)

- Note 3.** Do not use a UART1 reception error interrupt, an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation), and an INTIT40 interrupt 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, channel 3 of TAU0 (at higher 8-bit timer operation), and INTIT40 can be used at the same time. If one of the interrupt sources INTSRE1, INTTM03H, and INTIT40 is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.
- Note 4.** If one of the interrupt sources INTSR0, INTCSI01, INTIIC01, and INTIT51 is generated, bit 4 of the IF1L register is set to 1. Bit 4 of the MK1L, PR01L, and PR11L registers supports these four interrupt sources.
- Note 5.** If one of the interrupt sources INTST3, INTCSI30, INTIIC30, and INTIT31 is generated, bit 4 of the IF1H register is set to 1. Bit 4 of the MK1H, PR01H, and PR11H registers supports these four interrupt sources.
- Note 6.** If one of the interrupt sources INTSR3, INTCSI31, INTIIC31, and INTIT71 is generated, bit 5 of the IF1H register is set to 1. Bit 5 of the MK1H, PR01H, and PR11H registers supports these four interrupt sources.

Table 25 - 2 Flags Corresponding to Interrupt Request Sources (4/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Register		Register		Register						
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	✓	✓	✓	✓	✓	✓
INTURE3	UREIF3		UREMK3		UREPR03, UREPR13		✓	✓	—	—	—	—
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓	✓	✓	✓	✓
INTIT50	ITIF50		ITMK50		ITPR050, ITPR150		✓	✓	✓	✓	✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓	✓	✓	✓	✓
INTIT60	ITIF60		ITMK60		ITPR060, ITPR160		✓	✓	✓	✓	✓	✓
INTP6	PIF6		PMK6		PPR06, PPR16		✓	✓	✓	✓	✓	✓
INTP7	PIF7		PMK7		PPR07, PPR17		✓	✓	✓	✓	✓	✓
INTIT10	ITIF10		ITMK10		ITPR010, ITPR110		✓	✓	✓	✓	✓	✓
INTOSDC	OSDIF		OSDMK		OSDPR0, OSDPR1		✓	✓	✓	✓	✓	✓
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓	✓	✓	✓	✓
INTCMP0	CMPIF0	IF2H	CMPMK0	MK2H	CMPPR00, CMPPR10	PR02H, PR12H	✓	✓	✓	✓	✓	✓
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		✓	✓	✓	✓	—	—
INTURE0 ^{Note 1}	UREIF0 ^{Note 1}		UREMK0 ^{Note 1}		UREPR00, UREPR10 ^{Note 1}		✓	✓	✓	✓	✓	✓
INTIT70 ^{Note 1}	ITIF70 ^{Note 1}		ITMK70 ^{Note 1}		ITPR070, ITPR170 ^{Note 1}		✓	✓	✓	✓	✓	✓
INTURE1 ^{Note 2}	UREIF1 ^{Note 2}		UREMK1 ^{Note 2}		UREPR01, UREPR11 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTTMKB1 ^{Note 2}	TMKBIF1 ^{Note 2}		TMKBMK1 ^{Note 2}		TMKBPR01, TMKBPR11 ^{Note 2}		✓	✓	✓	✓	✓	✓
INTURE2 ^{Note 3}	UREIF2 ^{Note 3}		UREMK2 ^{Note 3}		UREPR02, UREPR12 ^{Note 3}		✓	✓	✓	✓	✓	—
INTTMKB2 ^{Note 3}	TMKBIF2 ^{Note 3}		TMKBMK2 ^{Note 3}		TMKBPR02, TMKBPR12 ^{Note 3}		✓	✓	✓	✓	✓	✓
INTSRE3 ^{Note 4}	SREIF3 ^{Note 4}		SREMK3 ^{Note 4}		SREPR03, SREPR13 ^{Note 4}		✓	✓	✓	✓	—	—
INTTRJ1 ^{Note 4}	TRJIF1 ^{Note 4}		TRJMK1 ^{Note 4}		TRJPR01, TRJPR11 ^{Note 4}		✓	✓	✓	✓	✓	✓
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		✓	✓	✓	✓	✓	✓
INTIICA1	IICAIF1		IICAMK1		IICAPR01, IICAPR11		✓	✓	✓	✓	✓	✓

Note 1. Do not use a UARTA0 reception error interrupt and an INTIT70 interrupt at the same time because they share flags for the interrupt request sources. If the UARTA0 reception error interrupt is not used (ISRMA0 = 1), UARTA0 and INTIT70 can be used at the same time. If one of the interrupt sources INTURE0 and INTIT70 is generated, bit 2 of the IF2H register is set to 1. Bit 2 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 2. Do not use a UARTA1 reception error interrupt and a timer KB1 count end interrupt at the same time because they share flags for the interrupt request sources. If the UARTA1 reception error interrupt is not used (ISRMA1 = 1), UARTA1 and timer KB1 can be used at the same time. If one of the interrupt sources INTURE1 and INTTMKB1 is generated, bit 3 of the IF2H register is set to 1. Bit 3 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 3. Do not use a UARTA2 reception error interrupt and a timer KB2 count end interrupt at the same time because they share flags for the interrupt request sources. If the UARTA2 reception error interrupt is not used (ISRMA2 = 1), UARTA2 and timer KB2 can be used at the same time. If one of the interrupt sources INTURE2 and INTTMKB2 is generated, bit 4 of the IF2H register is set to 1. Bit 4 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 4. Do not use a UART3 reception error interrupt and a timer RJ1 interrupt at the same time because they share flags for the interrupt request sources. If the UART3 reception error interrupt is not used (EOC03 = 0), UART3 and timer RJ1 can be used at the same time. If one of the interrupt sources INTSRE3 and INTTRJ1 is generated, bit 5 of the IF2H register is set to 1. Bit 5 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Table 25 - 2 Flags Corresponding to Interrupt Request Sources (5/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		100-pin	80-pin	64-pin	52-pin	48-pin	44-pin
		Register		Register		Register						
INTCTSURD	CTSURDIF	IF3L	CTSURDMK	MK3L	CTSURDPR0, CTSURDPR1	PR03L, PR13L	✓	✓	✓	✓	✓	✓
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		✓	✓	✓	✓	✓	✓
INTTRJ0	TRJIF0		TRJMK0		TRJPR00, TRJPR10		✓	✓	✓	✓	✓	✓
INTUT0	UTIF0		UTMK0		UTPR00, UTPR10		✓	✓	✓	✓	✓	✓
INTUR0	URIF0		URMK0		URPR00, URPR10		✓	✓	✓	✓	✓	✓
INTUT1	UTIF1		UTMK1		UTPR01, UTPR11		✓	✓	✓	✓	✓	✓
INTUR1	URIF1		URMK1		URPR01, URPR11		✓	✓	✓	✓	✓	✓
INTUT2	UTIF2		UTMK2		UTPR02, UTPR12		✓	✓	✓	✓	✓	—
INTUR2	URIF2	IF3H	URMK2	MK3H	URPR02, URPR12	PR03H, PR13H	✓	✓	✓	✓	✓	—
INTUR3	URIF3		URMK3		URPR03, URPR13		✓	✓	—	—	—	—
INTUR3	URIF3		URMK3		URPR03, URPR13		✓	✓	—	—	—	—
INTTMKB0	TMKBIF0		TMKBMK0		TMKBPR00, TMKBPR10		✓	✓	✓	✓	✓	✓
INTEXSD	EXSDIF		EXSDMK		EXSDPR0, EXSDPR1		✓	✓	✓	✓	✓	✓

25.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)

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, IF3L, and IF3H 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 and IF3H registers 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 that writes data to this register is executed, the number of instruction execution clock cycles increases by 2 cycles.

Figure 25 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (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 ITIF30	TMIF00	STIF0 CSIF00 IICIF00 ITIF21	SMSEIF	ELCLIF	SREIF2 ITIF20	SRIF2 CSIF21 IICIF21 ITIF41	STIF2 CSIF20 IICIF20 ITIF01

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 ITIF51	IICAF0	SREIF1 TMIF03H ITIF40	SRIF1 CSIF11 IICIF11 ITIF61	STIF1 CSIF10 IICIF10 ITIF11

Address: FFFE3H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	ITIF00	SRIF3 CSIF31 IICIF31 ITIF71	STIF3 CSIF30 IICIF30 ITIF31	KRIF	ITLIF	RTCIF	ADIF

Figure 25 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (2/2)

Address: FFFD0H
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	FLIF	PIF9 OSDIF	ITIF10	PIF7	PIF6	TMIF07 ITIF60	TMIF06 ITIF50	TMIF05 UREIF3

Address: FFFD1H
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	IICAIF1	CTSUWRIF	SREIF3 TRJIF1	UREIF2 TMKBIF2	UREIF1 TMKBIF1	UREIF0 ITIF70	CMPIF1	CMPIF0

Address: FFFD2H
 After reset: 00H
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	UTIF2	URIF1	UTIF1	URIF0	UTIF0	TRJIF0	CTSUFNIF	CTSURDIF

Address: FFFD3H
 After reset: 00H
 R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	EXSDIF	TMKBIF0	URIF3	UTIF3	URIF2

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 25 - 2 Flags Corresponding to Interrupt Request Sources. 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.

25.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt. The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, and MK3H 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 and MK3H registers 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 that writes data to this register is executed, the number of instruction execution clock cycles increases by 2 cycles.

Figure 25 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)
(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 ITMK30	TMMK00	STMK0 CSIMK00 IICMK00 ITMK21	SMSEMK	ELCLMK	SREMK2 ITMK20	SRMK2 CSIMK21 IICMK21 ITMK41	STMK2 CSIMK20 IICMK20 ITMK01

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 ITMK51	IICAMK0	SREMK1 TMMK03H ITMK40	SRMK1 CSIMK11 IICMK11 ITMK61	STMK1 CSIMK10 IICMK10 ITMK11

Address: FFFE7H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	ITMK00	SRMK3 CSIMK31 IICMK31 ITMK71	STMK3 CSIMK30 IICMK30 ITMK31	KRMK	ITLMK	RTCMK	ADMK

Figure 25 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (2/2)

Address: FFFD4H
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	FLMK	OSDMK	ITMK10	PMK7	PMK6	TMMK07 ITMK60	TMMK06 ITMK50	TMMK05 UREMK3

Address: FFFD5H
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	IICAMK1	CTSUWRMK	SREMK3 TRJMK1	UREMK2 TMKBMK2	UREMK1 TMKBMK1	UREMK0 ITMK70	CMPMK1	CMPMK0

Address: FFFD6H
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	UTMK2	URMK1	UTMK1	URMK0	UTMK0	TRJMK0	CTSUFNMK	CTSURDMK

Address: FFFD7H
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	EXSDMK	TMKBMK0	URMK3	UTMK3	URMK2

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 25 - 2 Flags Corresponding to Interrupt Request Sources. Be sure to set bits that are not available to the initial value.

25.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)

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, 3L, or 3H). The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H 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 and PR03H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers, and the PR13L and PR13H registers 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 that writes data to this register is executed, the number of instruction execution clock cycles increases by 2 cycles.

Figure 25 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (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 ITPR030	TMPR000	STPR00 CSIPR000 IICPR000 ITPR021	SMSEPR0	ELCLPR0	SREPR02 ITPR020	SRPR02 CSIPR021 IICPR021 ITPR041	STPR02 CSIPR020 IICPR020 ITPR001

Figure 25 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (2/4)

Address: FFFEDH

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H ITPR130	TMPR100	STPR10 CSIPR100 IICPR100 ITPR121	SMSEPR1	ELCLPR1	SREPR12 ITPR120	SRPR12 CSIPR121 IICPR121 ITPR141	STPR12 CSIPR120 IICPR120 ITPR101

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 ITPR051	IICAPR00	SREPR01 TMPR003H ITPR040	SRPR01 CSIPR011 IICPR011 ITPR061	STPR01 CSIPR010 IICPR010 ITPR011

Address: FFFECH

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	SRPR10 CSIPR101 IICPR101 ITPR151	IICAPR10	SREPR11 TMPR103H ITPR140	SRPR11 CSIPR111 IICPR111 ITPR161	STPR11 CSIPR110 IICPR110 ITPR111

Address: FFFEBH

After reset: FFH

R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	ITPR000	SRPR03 CSIPR031 IICPR031 ITPR071	STPR03 CSIPR030 IICPR030 ITPR031	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	ITPR100	SRPR13 CSIPR131 IICPR131 ITPR171	STPR13 CSIPR130 IICPR130 ITPR131	KRPR1	ITLPR1	RTCPR1	ADPR1

Figure 25 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (3/4)

Address: FFFD8H
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	FLPR0	OSDPR0	ITPR010	PPR07	PPR06	TMPR007 ITPR060	TMPR006 ITPR050	TMPR005 UREPR03

Address: FFFDCH
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	FLPR1	OSDPR1	ITPR110	PPR17	PPR16	TMPR107 ITPR160	TMPR106 ITPR150	TMPR105 UREPR13

Address: FFFD9H
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	IICAPR01	CTSUWRP R0	SREPR03 TRJPR01	UREPR02 TMKBPR02	UREPR01 TMKBPR01	UREPR00 ITPR070	CMPPR01	CMPPR00

Address: FFFDDH
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	IICAPR11	CTSUWRP R1	SREPR13 TRJPR11	UREPR12 TMKBPR12	UREPR11 TMKBPR11	UREPR10 ITPR170	CMPPR11	CMPPR10

Address: FFFDAH
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	UTPR02	URPR01	UTPR01	URPR00	UTPR00	TRJPR00	CTSUFNPR 0	CTSURDPR 0

Address: FFFDEH
 After reset: FFH
 R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	UTPR12	URPR11	UTPR11	URPR10	UTPR10	TRJPR10	CTSUFNPR 1	CTSURDPR 1

Figure 25 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (4/4)

Address: FFFDBH
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	EXSDPR0	TMKBPR00	URPR03	UTPR03	URPR02

Address: FFFDFH
 After reset: FFH
 R/W: R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	EXSDPR1	TMKBPR10	URPR13	UTPR13	URPR12

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 25 - 2 Flags Corresponding to Interrupt Request Sources. Be sure to set bits that are not available to the initial value.

25.3.4 External interrupt rising edge enable register (EGP0), External interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to INTP7. The EGP0 and EGN0 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 25 - 5 Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

Address: FFF38H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	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	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 25 - 3 shows the ports (INTPx pins) corresponding to the EGPn and EGNn bits.

Table 25 - 3 Ports (INTPx Pins) Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Port (INTPx Pin)	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin
EGP0	EGN0	INTP0	✓	✓	✓	✓	✓	✓
EGP1	EGN1	INTP1	(✓)	(✓)	(✓)	✓	✓	✓
EGP2	EGN2	INTP2	(✓)	(✓)	✓	✓	✓	✓
EGP3	EGN3	INTP3	(✓)	(✓)	(✓)	✓	✓	✓
EGP4	EGN4	INTP4	(✓)	(✓)	(✓)	(✓)	(✓)	✓
EGP5	EGN5	INTP5	(✓)	(✓)	(✓)	(✓)	✓	✓
EGP6	EGN6	INTP6	✓	✓	✓	✓	✓	✓
EGP7	EGN7	INTP7	(✓)	(✓)	(✓)	✓	✓	✓

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.

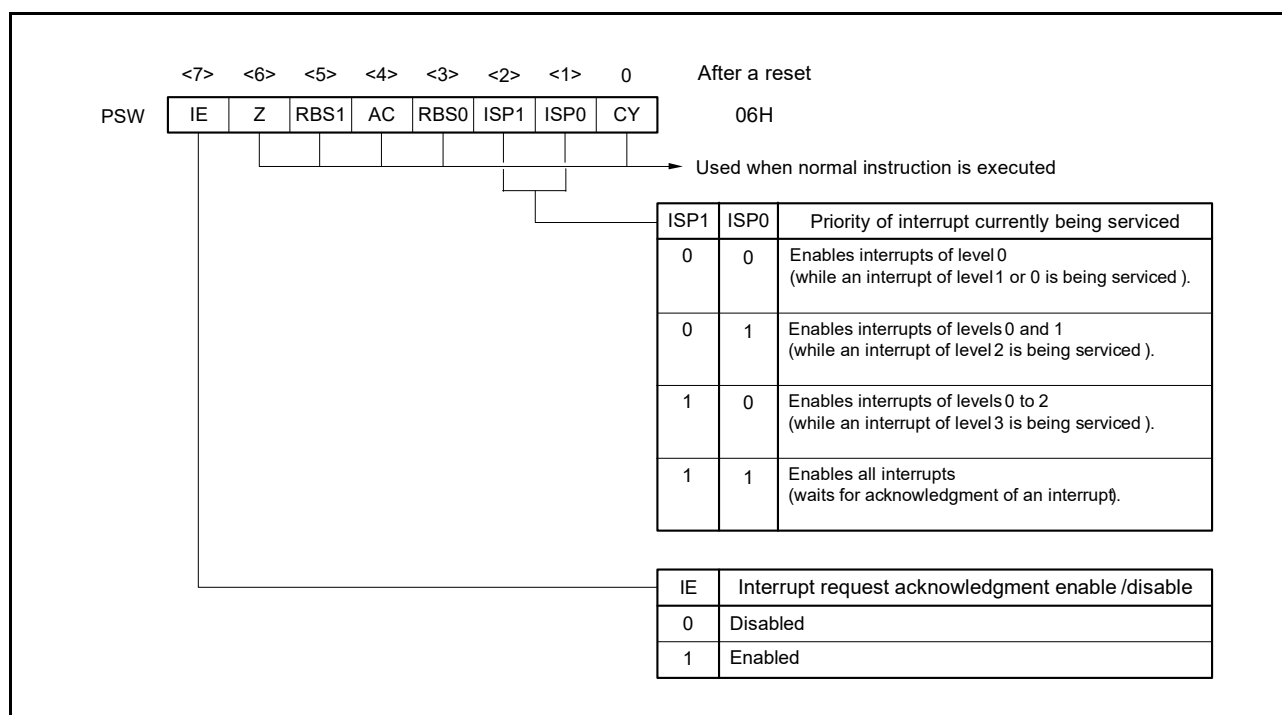
Remark 2. n = 0 to 7

Remark 3. Parentheses indicate that the pin can be assigned via settings in the peripheral I/O redirection registers (PIORx). For details, see 4.3.10 Peripheral I/O redirection registers (PIORx).

25.3.5 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 25 - 6 Configuration of Program Status Word



25.3.6 Registers for controlling the port functions multiplexed with the interrupt inputs

Set the following registers for controlling the port functions multiplexed with the interrupt inputs.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting registers for controlling the port functions, see **4.5.4 Examples of register settings for port and alternate functions**.

25.4 Interrupt Servicing Operations

25.4.1 Maskable interrupt request acknowledgment

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. The times required from acceptance of an interrupt request in the form of the generation of a maskable interrupt request until vectored interrupt servicing proceeds are in the range indicated in **Table 25 - 4** below. For the interrupt request acknowledgment timing, see **Figure 25 - 8** and **Figure 25 - 9**.

Table 25 - 4 Times from Generation of Maskable Interrupt Request until Vectored Interrupt Servicing

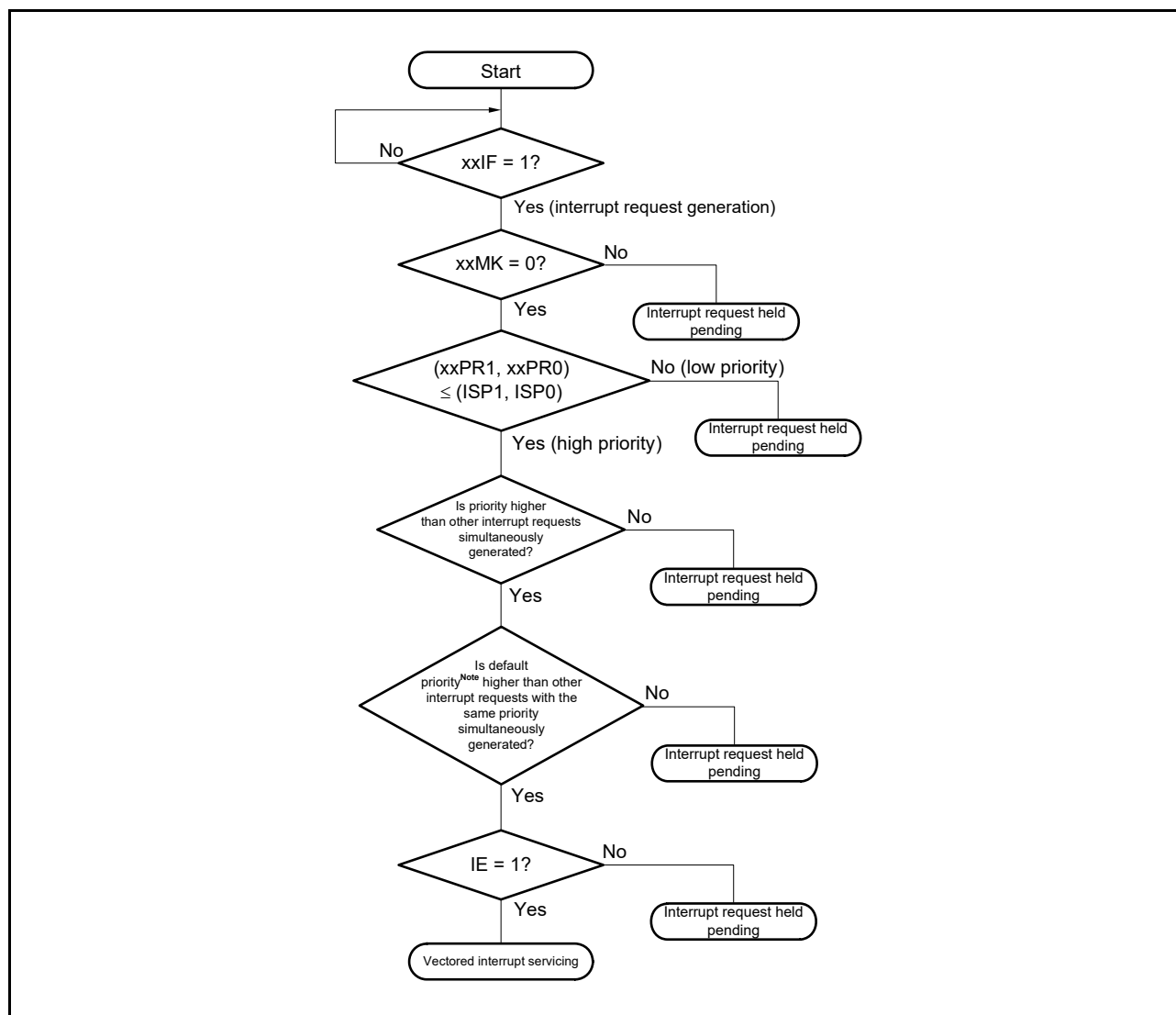
	Minimum Time	Maximum Time ^{Note}
Times required for interrupt request acceptance	9 clock cycles	16 clock cycles

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock cycle: $1/f_{CLK}$ (f_{CLK} : CPU clock frequency)

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 25 - 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 to 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 25 - 7 Interrupt Request Acknowledgment Processing Algorithm

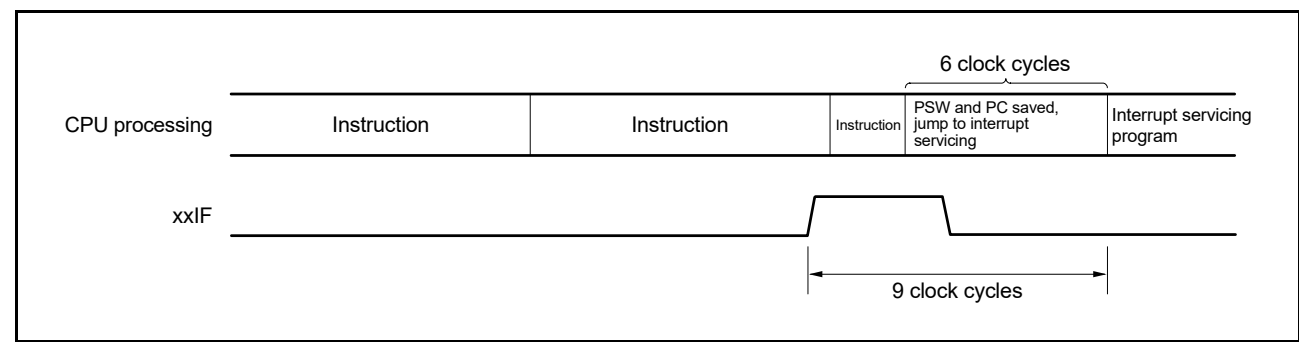


Note For the default priority, refer to **Table 25 - 1 Interrupt Source List**.

Remark

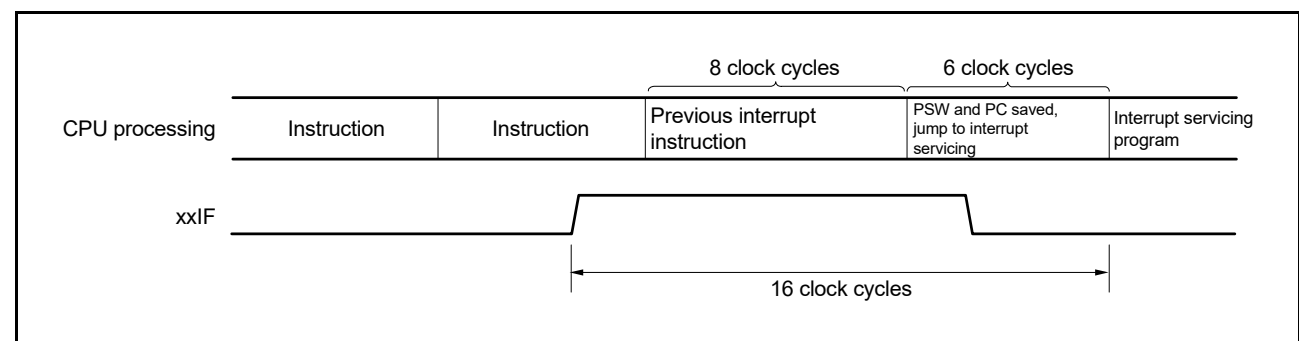
- xxIF: Interrupt request flag
- xxMK: Interrupt mask flag
- xxPR0: Priority specification flag 0
- xxPR1: Priority specification flag 1
- IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
- ISP0, ISP1: Flags that indicate the priority level of the interrupt currently being serviced (see **Figure 25 - 6**)

Figure 25 - 8 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock cycle: 1/fCLK (fCLK: CPU clock frequency)

Figure 25 - 9 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock cycle: 1/fCLK (fCLK: CPU clock frequency)

25.4.2 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 to 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.

25.4.3 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 25 - 5** shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 25 - 10** shows multiple interrupt servicing examples.

Table 25 - 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, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)

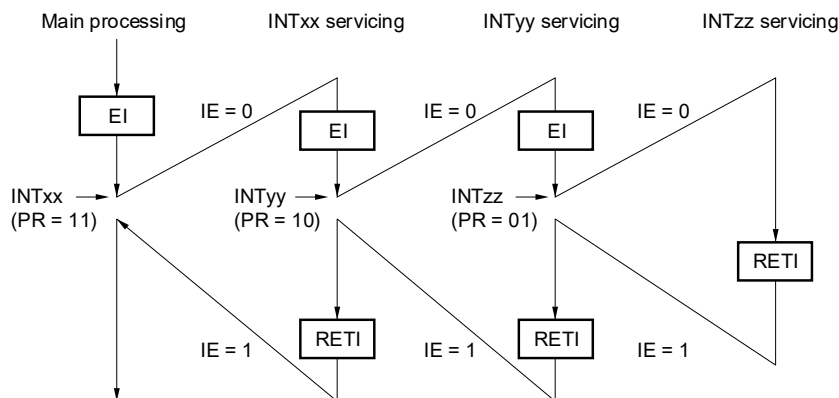
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1

PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0

PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

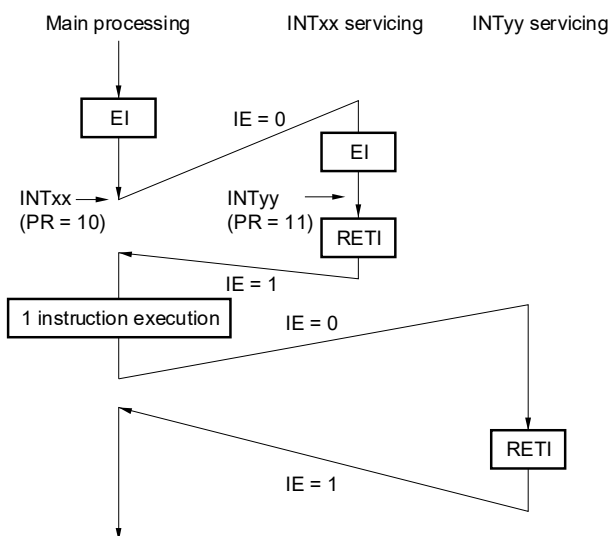
Figure 25 - 10 Examples of Multiple Interrupt Servicing (1/2)

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



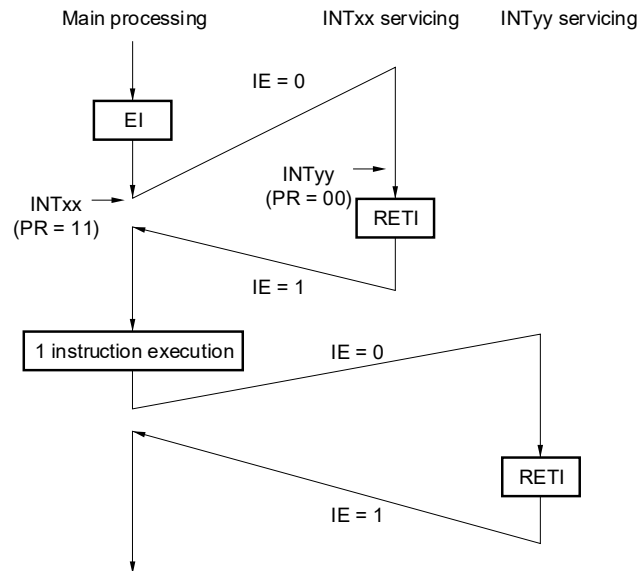
Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

Remark

- 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 25 - 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.

Remark

- 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.

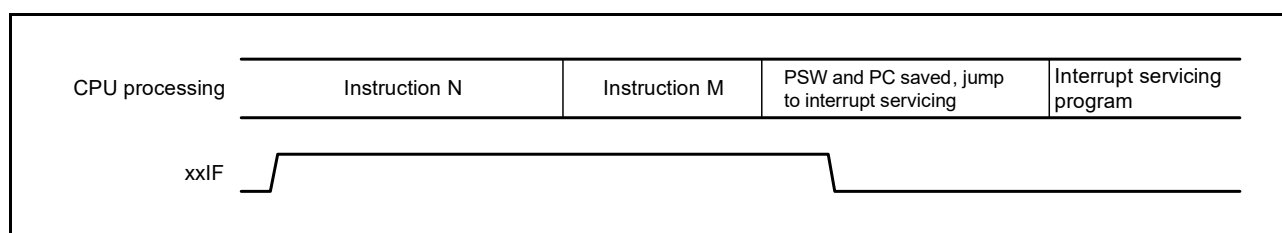
25.4.4 Interrupt request held pending

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, IF3H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers

Figure 25 - 11 shows the timing at which interrupt requests are held pending.

Figure 25 - 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 26 Key Interrupt Function

The number of key interrupt input channels differs, depending on the product.

	44-pin products	48-pin products	52-pin products	64-pin products	80- and 100-pin products
Number of the key interrupt input channels	2	3	4	5	8

26.1 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 KR7).

Table 26 - 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
KR6	KRM06
KR7	KRM07

Remark Pins KR6 and KR7 are only present in the 44-pin products.
Pins KR5 to KR7 are only present in the 48-pin products.
Pins KR4 to KR7 are only present in the 52-products.
Pins KR0 and KR4 to KR7 are only present in the 64-products.
Pins KR0 to KR7 are only present in the 80- and 100-pin products.

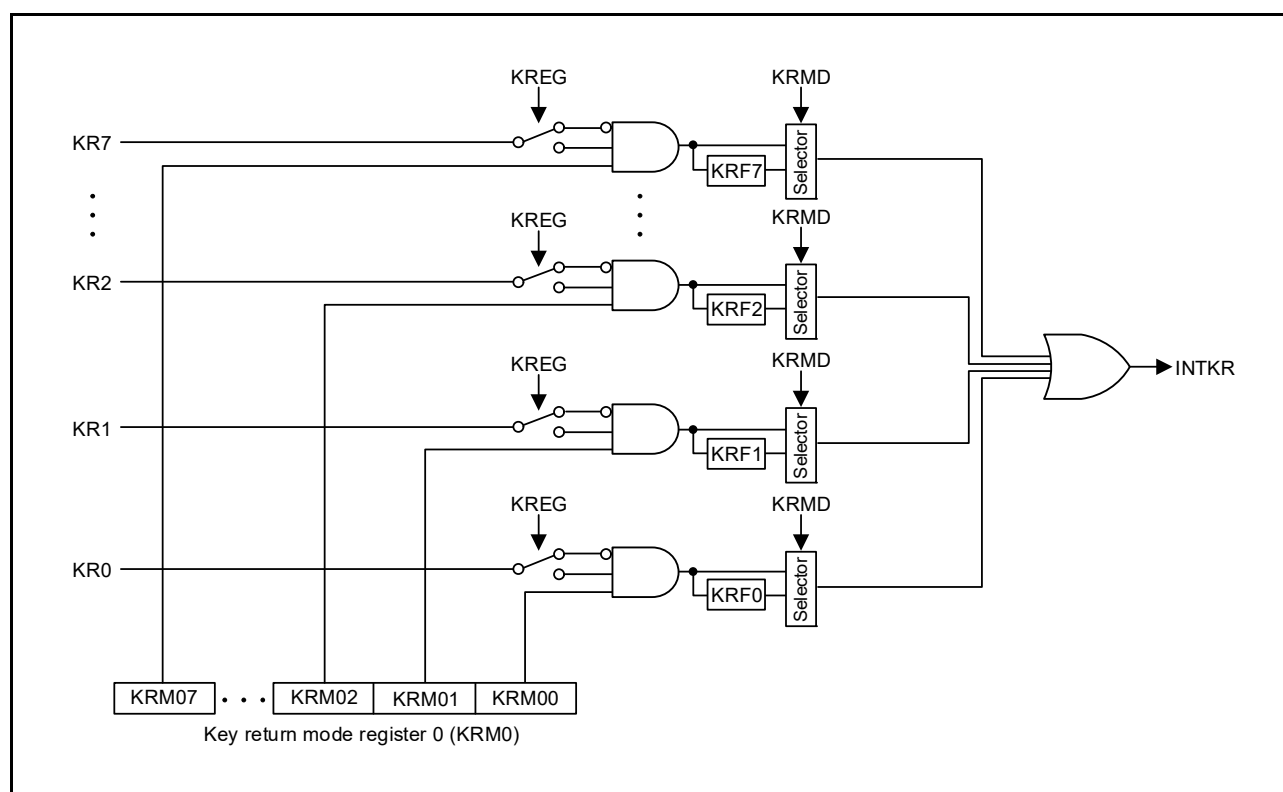
26.2 Configuration of the Key Interrupt

The key interrupt includes the following hardware blocks.

Table 26 - 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 registers (PMxx) Port mode control T registers (PMCTxx) Port mode control E registers (PMCExx) LCD port function registers (PFSEGx)

Figure 26 - 1 Block Diagram of the Key Interrupt



Remark Pins KR6 and KR7 are only present in the 44-pin products.
Pins KR5 to KR7 are only present in the 48-pin products.
Pins KR4 to KR7 are only present in the 52-products.
Pins KR0 and KR4 to KR7 are only present in the 64-products.
Pins KR0 to KR7 are only present in the 80- and 100-pin products.

26.3 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 registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

26.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF7) 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 26 - 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 KRF7)						
	0	Does not use key return flags						
	1	Uses key return flags						
	KREG	Selection of detection edge (KR0 to KR7)						
	0	Falling edge						
	1	Rising edge						

26.3.2 Key return mode register 0 (KRM0)

The KRM0 register controls the KR0 to KR7 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 26 - 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	KRM07	KRM06	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 42.4 AC Characteristics).

Caution 3. The pins not used in the key interrupt mode can be used as normal port pins.

Caution 4. Be sure to set the following bits to 0.

Bits 5 to 0 in the 44-pin products

Bits 4 to 0 in the 48-pin products

Bits 3 to 0 in the 52-pin products

Bits 3 to 1 in the 64-pin products

Remark n = 0 to 7

26.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF7). The KRF register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 26 - 4 Format of Key Return Flag Register (KRF)

Address: FFF35H
 After reset: 00H
 R/W: R/W>Note

Symbol	7	6	5	4	3	2	1	0
KRF	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 7)
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.

Remark n = 0 to 7

26.3.4 Registers for controlling the port functions multiplexed with the key interrupt inputs

Set the following registers for controlling the port functions multiplexed with the key interrupt inputs.

- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and alternate functions**.

Section 27 Standby Function

27.1 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/L23 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 16 A/D Converter (ADC)**
- **Section 19 Serial Array Unit (SAU)**
- **Section 23 Data Transfer Controller (DTC)**
- **Section 34 SNOOZE Mode Sequencer (SMS)**
- **Section 35 Capacitive Sensing Unit (CTSU2La)**

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. **Note** The states of the output latches and the output buffers for I/O port pins are also retained.

Note Memory which has been placed in shutdown mode by the PSMCR register does not retain values it had immediately before the transition to the standby mode.

- Caution 1.** Shifting to the STOP mode is only enabled when the CPU is operating with 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 with 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 the WDTON bit of the option byte or the WUTMMCK0 bit 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.

27.2 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)
- Memory power reduction control register (PSMCR)

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 16 A/D Converter (ADC)**
- **Section 19 Serial Array Unit (SAU)**
- **Section 23 Data Transfer Controller (DTC)**
- **Section 34 SNOOZE Mode Sequencer (SMS)**
- **Section 35 Capacitive Sensing Unit (CTSU2La)**

27.2.1 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 27 - 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. ^{Notes 3, 4}

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/L23 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 **27.3.2 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 42 Electrical Characteristics**.

Note 4. This is only specifiable when the setting of the FRQSEL4 bit of the option byte (000C2H) is 0.

27.2.2 Memory power reduction control register (PSMCR)

The PSMCR register is used to control the reduction of power consumption by the RAM. The leakage current can be reduced by placing the RAM in shutdown mode. The supply of power to the shut-down part of the RAM stops. Accordingly, that RAM does not retain data. The PSMCR 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 - 2 Format of Memory Power Reduction Control Register (PSMCR)

Address: F0216H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PSMCR	0	0	0	0	0	0	RAMSDMD	RAMSDS

RAMSDMD	RAMSDS	Operating mode of the RAM
0	0	Normal mode (continues to operate)
1	0	Standby mode
1	1	Shutdown mode
Other than above		Setting prohibited

Caution 1. Shutdown mode applies to all RAM other than that in the range from FF000H to FFEFFH. The RAM that in the range from FF000H to FFEFFH continues to operate and retains data.

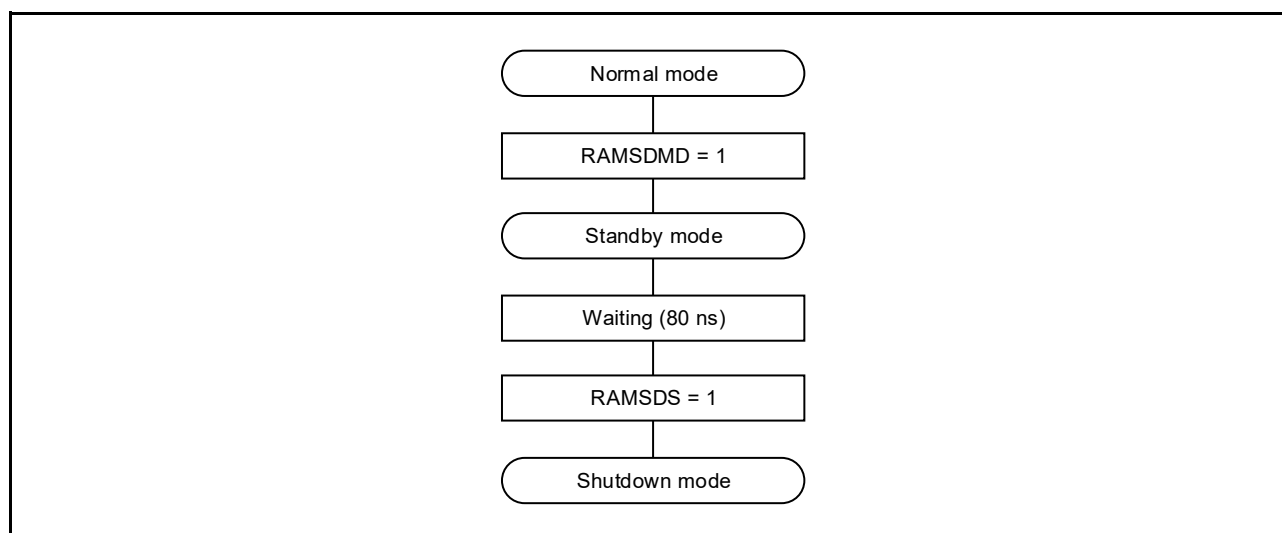
Caution 2. Do not access RAM while it is in the standby mode or shutdown mode.

Caution 3. When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from FF000H to FFEFFH are undefined.

Caution 4. Access to the address ranges other than FF000H to FFEFFH in the RAM in the shutdown mode is still possible in on-chip debugging. When the RAM returns to normal mode from shutdown mode in on-chip debugging, the contents of the RAM other than in the range from FF000H to FFEFFH are not undefined.

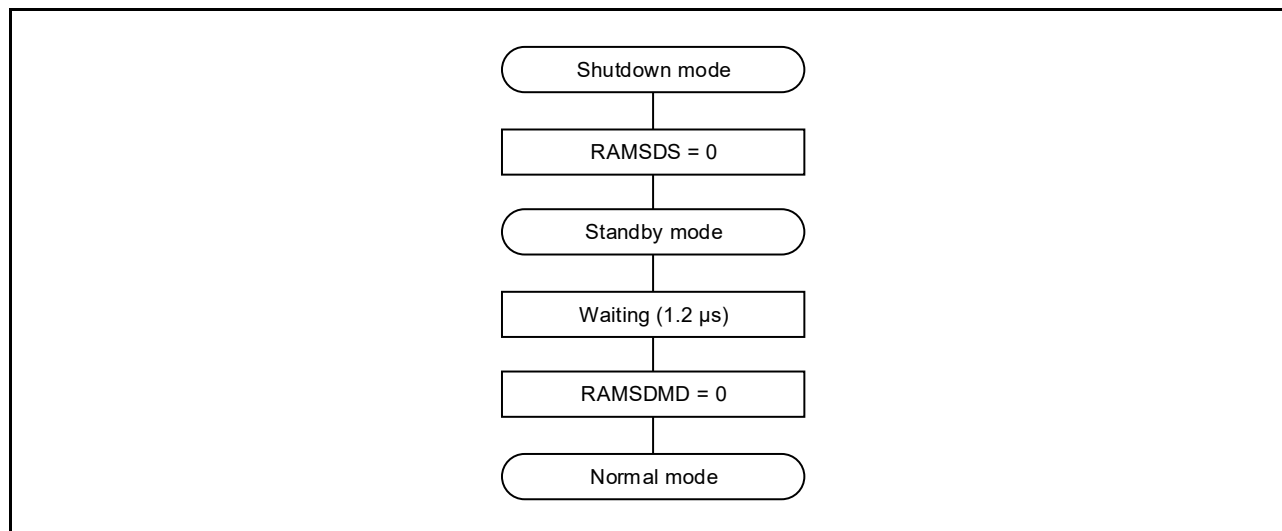
Follow the procedure below to switch the operating mode of the RAM from normal mode to shutdown mode.

Figure 27 - 3 Procedure for Settings to Switch from Normal Mode to Shutdown Mode



Follow the procedure below to switch the operating mode of the RAM from shutdown mode to normal mode.

Figure 27 - 4 Procedure for Settings to Switch from Shutdown Mode to Normal Mode



Caution When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from FF000H to FFEFFH are undefined. Initialize the RAM area to be used.

27.3 Standby Function Operation

27.3.1 HALT mode

1. HALT mode setting and operating states

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.

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.

The operating states in the HALT mode are shown below.

Table 27 - 1 Operating States in the HALT Mode (1) (1/2)

Item		HALT Mode Setting		When a HALT Instruction is Executed While the CPU is Operating with the Main System Clock			
				When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{ih})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f _{im})	When the CPU is Operating with the X1 Clock (f _x)	When the CPU is Operating with the External Main System Clock (f _{ex})
System clock				Clock supply to the CPU is stopped.			
	Main system clock	f _{ih}	Operation continues. Stopping this clock signal is impossible.	Operation prohibited			
		f _{im}	Operation prohibited	Operation continues. Stopping this clock signal is impossible.	Operation prohibited		
		f _x	Operation prohibited		Operation continues. Stopping this clock signal is impossible.	Operation impossible	
		f _{ex}			Operation impossible	Operation continues. Stopping this clock signal is impossible.	
	Subsystem clock	f _{xt}	Operation before the transition to the HALT mode continues.				
		f _{exs}					
		f _{il}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of the option byte (000C0H), and the WUTMMCK0 bit of the 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 clock X (fsx) or XR (fsxr) is the operating clock. 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; however, operation in response to access by the DTC or SMS is possible.			
Port (latch)				Retains the output before the transition to the HALT mode; however, operation in response to access by the DTC, SMS, or ELCL is possible.			
Timer array unit				Operation possible			
Timer RJ				Operation possible			
16-bit timers KB40, KB41, and KB42				Operation possible			
RTC				Operation possible			
8-bit interval timers				Operation possible			

Table 27 - 1 Operating States in the HALT Mode (1) (2/2)

HALT Mode Setting Item		When a HALT Instruction is Executed While the CPU is Operating with the Main System Clock			
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f _{IM})	When the CPU is Operating with the X1 Clock (f _X)	When the CPU is Operating with the External Main System Clock (f _{EX})
32-bit interval timer		Operation possible			
External signal sampler		Operation possible only when the subsystem clock is the operating clock for the external signal sampler			
Watchdog timer		See Section 15 Watchdog Timer (WDT) .			
Clock output/buzzer output		Operation possible			
A/D converter					
D/A converter					
Comparator					
Serial array unit					
Serial interface IICA					
Serial interface UARTA					
LCD controller/driver					
Data transfer controller (DTC)		Operation possible			
SNOOZE mode sequencer					
Logic and event link controller (ELCL)		Operation-possible function blocks can be linked.			
Power-on-reset function		Operation possible			
Voltage detection function					
Oscillation stop detection		Operation possible only when f _{IL} is oscillating			
External interrupt		Operation possible			
Key interrupt function					
Capacitive sensing unit (CTS02La) ^{Note}					
CRC operation function	High-speed CRC				
	General-purpose CRC	Operation possible in response to access by the DTC or SMS to obtain data for calculations from the RAM area			
Illicit memory access detection function		Operation possible 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 possible			

Note The measurement accuracy of the capacitive sensing unit (CTS2La) depends on the accuracy of the operating clock. When CTS2La is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or the high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.
Operation prohibited: Stop operation before switching to the HALT mode.

Remark 2. 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 27 - 2 Operating States in the HALT Mode (2) (1/2)

HALT Mode Setting Item			When a HALT Instruction is Executed While the CPU is Operating with the Subsystem Clock		
			When the CPU is Operating with the XT1 Clock (fXT)	When the CPU is Operating with the External Subsystem Clock (fEXS)	When the CPU is Operating with the Low-speed On-chip Oscillator Clock (fIL)
System clock			Clock supply to the CPU is stopped.		
	Main system clock	fIH	Operation prohibited		
		fIM			
		fX			
		fEX			
	Subsystem clock	fXT	Operation continues. Stopping this clock signal is impossible.	Operation impossible	Operation prohibited
		fEXS	Operation impossible	Operation continues. Stopping this clock signal is impossible.	Operation prohibited
		fIL	Set by bits 0 (WDSTBYON) and 4 (WDTON) of the option byte (000C0H), and the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC). WUTMMCK0 = 1: Oscillates Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock X (fsx) or XR (fsxR) is the operating clock. 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. Stopping this clock signal is impossible.
CPU		Operation stopped			
Code flash memory					
Data flash memory					
RAM		Operation stopped; however, operation in response to access by the DTC or SMS is possible.			
Port (latch)		Retains the output before the transition to the HALT mode; however, operation in response to access by the DTC, SMS, or ELCL is possible.			
Timer array unit		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible	
Timer RJ		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)			
16-bit timers KB40, KB41, and KB42		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)			
RTC		Operation possible			
8-bit interval timers		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)			
32-bit interval timer		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible	
External signal sampler		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)			
Watchdog timer		See Section 15 Watchdog Timer (WDT) .			
Clock output/buzzer output		Operation possible			
A/D converter		Operation prohibited			
D/A converter		Retains the output before the transition to the HALT mode; however, operation in response to access by the DTC, SMS, or ELCL when the RTCLPC bit is 0 is possible.			
Comparator		Operation possible when the RTCLPC bit is 0 or when the digital filters are not in use		Operation possible	
Serial array unit		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible	
Serial interface IICA		Operation prohibited			
Serial interface UARTA		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible	
LCD controller/driver		Operation possible depending on the state of the clock selected as the LCD source clock; operation proceeds or is stopped according to whether the clock is operating or is stopped, respectively.			

Table 27 - 2 Operating States in the HALT Mode (2) (2/2)

Item \ HALT Mode Setting		When a HALT Instruction is Executed While the CPU is Operating with the Subsystem Clock		
		When the CPU is Operating with the XT1 Clock (fXT)	When the CPU is Operating with the External Subsystem Clock (fEXS)	When the CPU is Operating with the Low-speed On-chip Oscillator Clock (fIL)
Data transfer controller (DTC)		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible
SNOOZE mode sequencer		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		Operation possible
Logic and event link controller (ELCL)		Operation-possible function blocks can be linked.		
Power-on-reset function		Operation possible		
Voltage detection function				
Oscillation stop detection		Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)		
External interrupt		Operation possible		
Key interrupt function				
Capacitive sensing unit (CTS2La) ^{Note}		Operation prohibited		
CRC operation function	High-speed CRC	Operation prohibited		
	General-purpose CRC	Operation possible in response to access by the DTC or SMS to obtain data for calculations from the RAM area		
Illicit memory access detection function		Operation possible 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 possible		

Note The measurement accuracy of the capacitive sensing unit (CTS2La) depends on the accuracy of the operating clock. When CTS2La is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or the high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.
Operation prohibited: Stop operation before switching to the HALT mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock fx: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

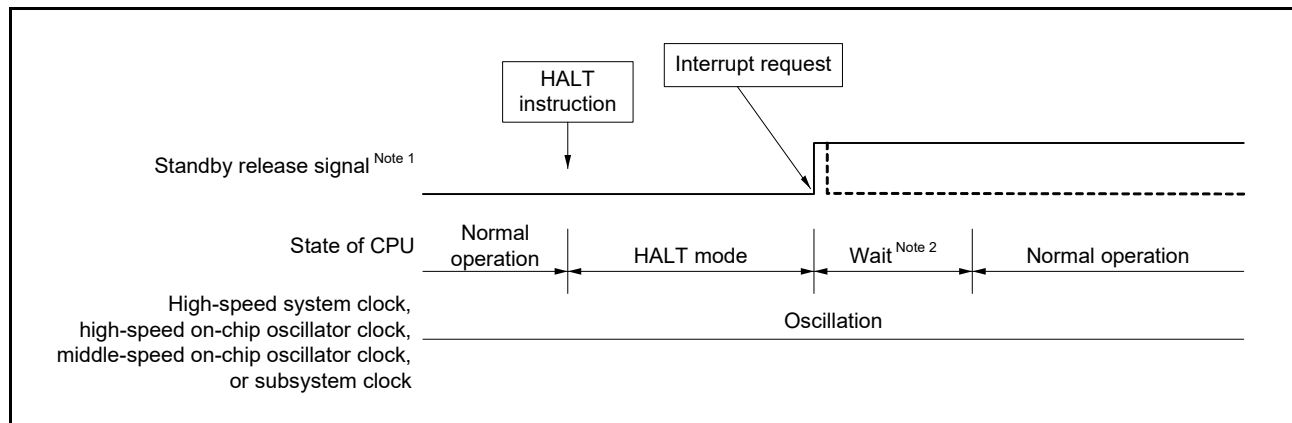
2. HALT mode release

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 27 - 5 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for HALT mode release

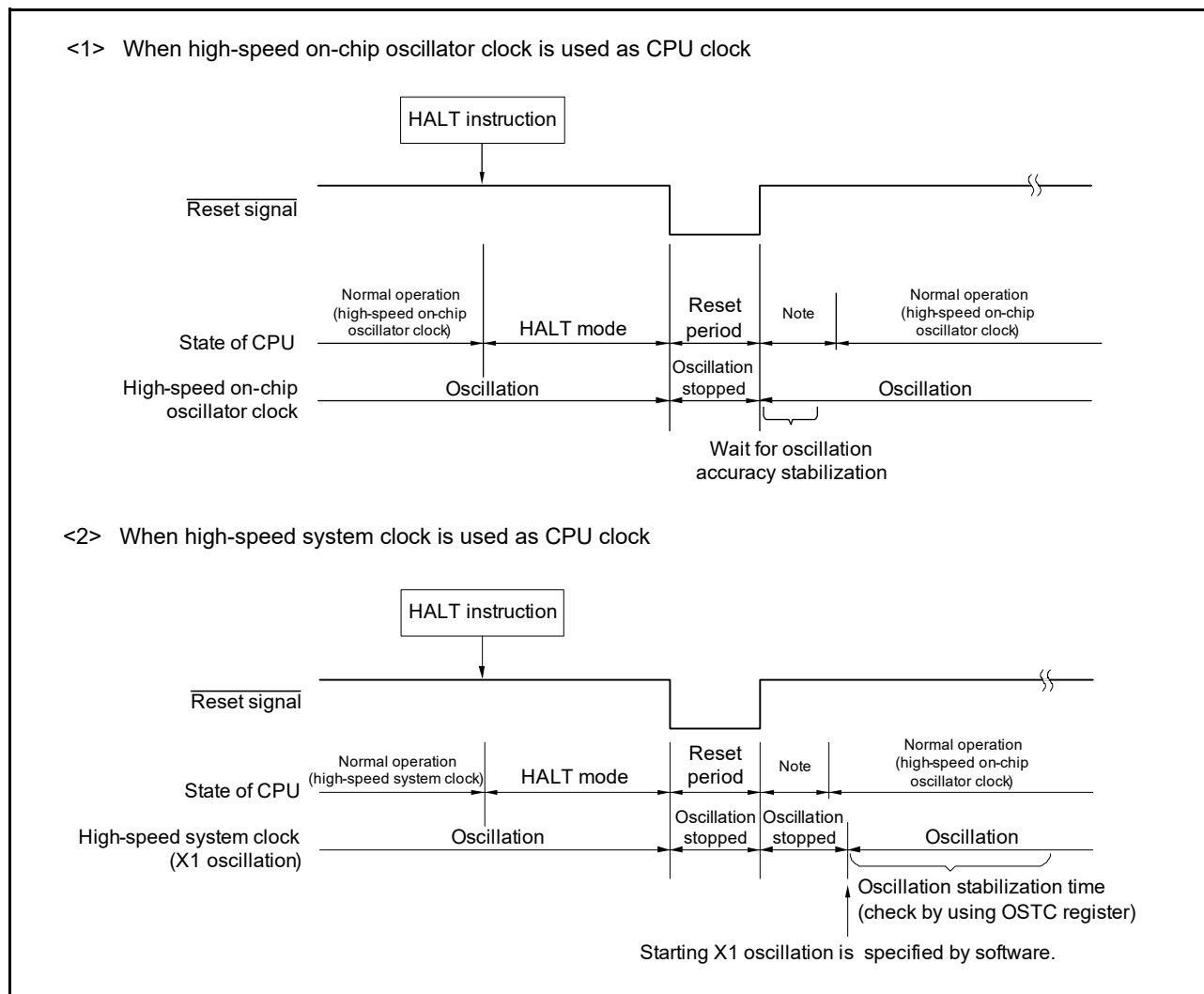
- When vectored interrupt servicing is carried out
 - Main system clock: 15 to 16 clock cycles
 - Subsystem clock (RTCLPC = 0): 10 to 11 clock cycles
 - Subsystem clock (RTCLPC = 1): 11 to 12 clock cycles
- When vectored interrupt servicing is not carried out
 - Main system clock: 9 to 10 clock cycles
 - 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

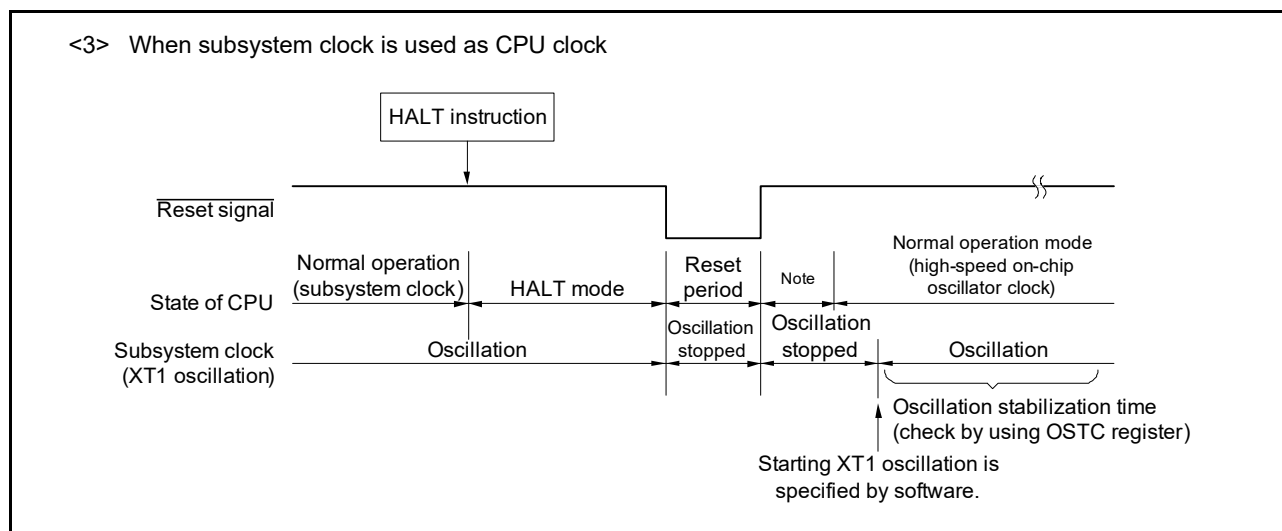
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 27 - 6 HALT Mode Release by Reset (1/2)



Note For the reset processing time, see **Section 28 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 29 Power-on-reset Circuit (POR)**.

Figure 27 - 6 HALT Mode Release by Reset (2/2)



Note For the reset processing time, see **Section 28 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 29 Power-on-reset Circuit (POR)**.

27.3.2 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 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 27 - 3 Operating States in the STOP Mode (1/2)

STOP Mode Setting			When a STOP Instruction is Executed While the CPU is Operating with the Main System Clock			
			When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f _{IM})	When the CPU is Operating with the X1 Clock (f _X)	When the CPU is Operating with the External Main System Clock (f _{EX})
Item						
System clock			Clock supply to the CPU is stopped.			
	Main system clock	f _{IH}	Stopped			
		f _{IM}	Stopped			
		f _X	Stopped			
		f _{EX}				
	Subsystem clock	f _{XT}	Operation before the transition to the STOP mode continues.			
		f _{EXS}				
f _{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of the option byte (000C0H), and the WUTMMCK0 bit of the 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 clock X (f _{sx}) or XR (f _{sXR}) is the operating clock. 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 output before the transition to the STOP mode.			
Timer array unit			Operation stopped			
Timer RJ			• Operation possible in the event counter mode when the TRJION input with no filter is selected • Operation possible when a subsystem clock is selected as the source for counting and the RTCLPC bit in the OSMC register is 0 • Operation possible when the low-speed on-chip oscillator is selected as the source for counting • Operation prohibited under any conditions other than those listed above			
16-bit timers KB40, KB41, and KB42			Operation prohibited			
RTC			Operation possible			
8-bit interval timers			Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)			
32-bit interval timer			Operation possible when f _{sxp} is selected and the RTCLPC bit is 0			
External signal sampler			Operation possible when the RTCLPC bit is 0 and the EXSDLPC bit is 1 (operation prohibited otherwise)			
Watchdog timer			See Section 15 Watchdog Timer (WDT) .			
Clock output/buzzer output			Operation possible only when f _{subr} is selected			
A/D converter			Waking up possible; switching to the SNOOZE mode			
D/A converter			Retains the output before the transition to the STOP mode.			
Comparator			Operation possible only when the digital filters are not in use			
Serial array unit			Waking up possible only for CSI00, CSI20, UART0, and UART2; switching to the SNOOZE mode (operation prohibited otherwise)			
Serial interface IICA			Waking up possible in response to address matching			
Serial interface UARTA			Operation possible when f _{sxp} is selected and the RTCLPC bit is 0			
LCD controller/driver			Operation possible depending on the state of the clock selected as the LCD source clock; operation proceeds or is stopped according to whether the clock is operating or is stopped, respectively.			
Data transfer controller (DTC)			DTC activation source acceptance possible; switching to the SNOOZE mode			
SNOOZE mode sequencer			SMS activation source acceptance possible; switching to the SNOOZE mode			
Logic and event link controller (ELCL)			Operation-possible function blocks can be linked.			

Table 27 - 3 Operating States in the STOP Mode (2/2)

STOP Mode Setting Item		When a STOP Instruction is Executed While the CPU is Operating with the Main System Clock			
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f _{IM})	When the CPU is Operating with the X1 Clock (f _x)	When the CPU is Operating with the External Main System Clock (f _{EX})
Power-on-reset function		Operation possible			
Voltage detection function					
Oscillation stop detection		Operation possible when f _{IL} is oscillating and the RTCLPC bit is 0 (operation prohibited otherwise)			
External interrupt		Operation possible			
Key interrupt function					
Capacitive sensing unit (CTS02La) ^{Note}		CTS0 activation source acceptance possible; switching to the SNOOZE mode			
CRC operation function	High-speed CRC	Operation stopped			
	General-purpose CRC				
Illicit memory access detection function					
RAM parity error detection function					
RAM guard function					
SFR guard function					
True random number generator					

Note The measurement accuracy of the capacitive sensing unit (CTS2La) depends on the accuracy of the operating clock. When CTS2La is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or the high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.
Operation prohibited: Stop operation before switching to the STOP mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock fx: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock f_{SXP}: Low-speed peripheral clock frequency
f_{SUBR}: Subsystem clock R frequency

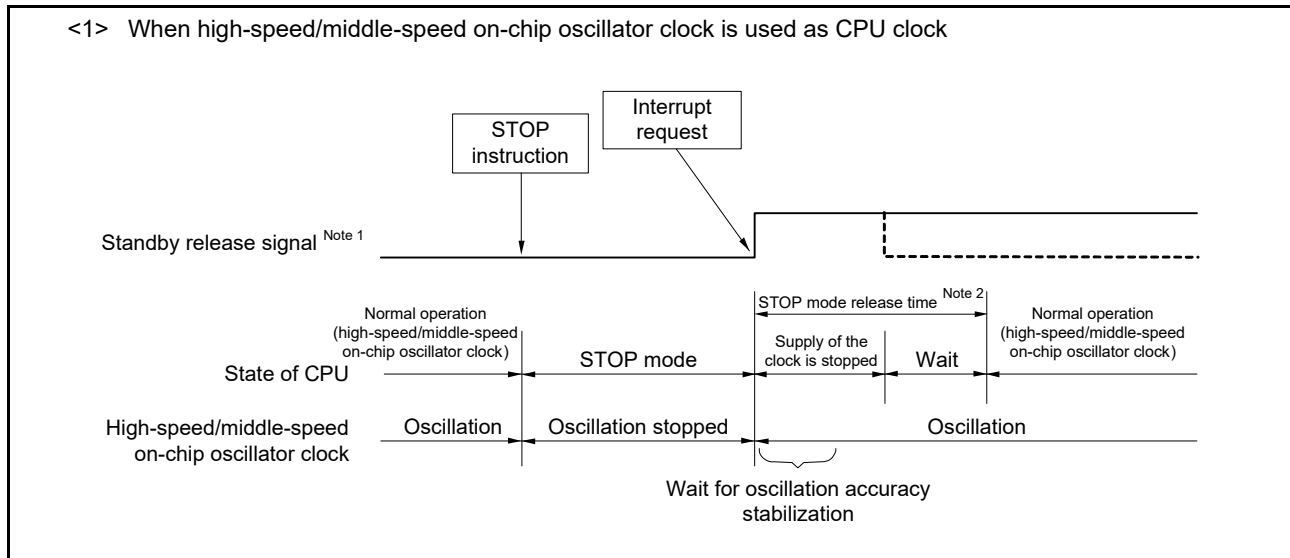
2. STOP mode release

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 27 - 7 STOP Mode Release by Interrupt Request Generation (1/3)



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

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 + 3 to 4 clock cycles (FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)

0.6 to 0.8 μ s + 3 to 4 clock cycles (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 42 Electrical Characteristics**.

For the middle-speed on-chip oscillator clock: 1.5 to 2.5 μ s + 3 to 4 clock cycles

Wait:

(common to the high-speed/middle-speed on-chip oscillator clock)

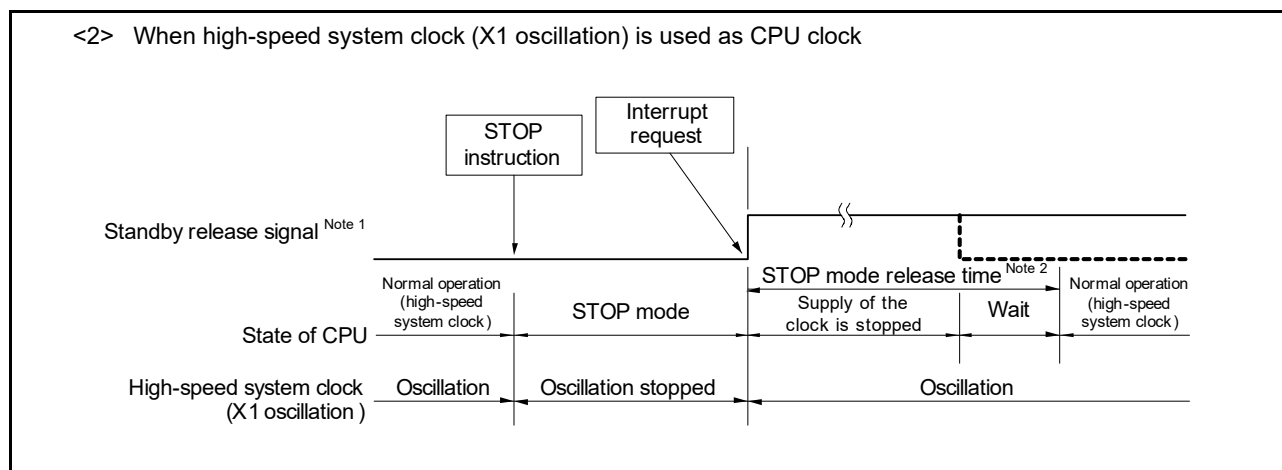
- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

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 27 - 7 STOP Mode Release by Interrupt Request Generation (2/3)



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

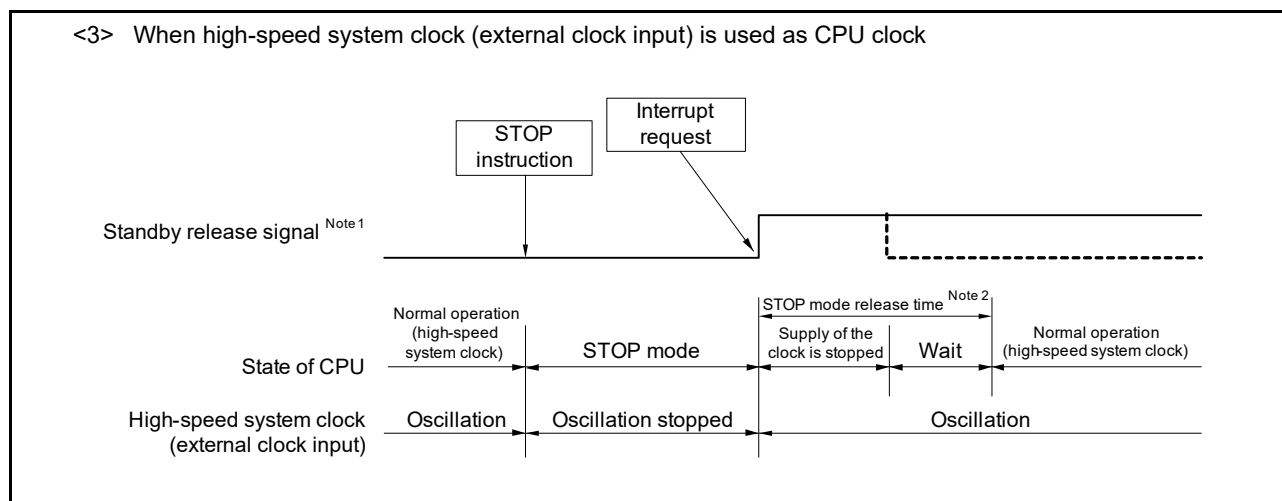
Supply of the clock is stopped:

Oscillation stabilization time (set by OSTS) + 3 to 4 clock cycles

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Figure 27 - 7 STOP Mode Release by Interrupt Request Generation (3/3)



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

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 the CPU operates with the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

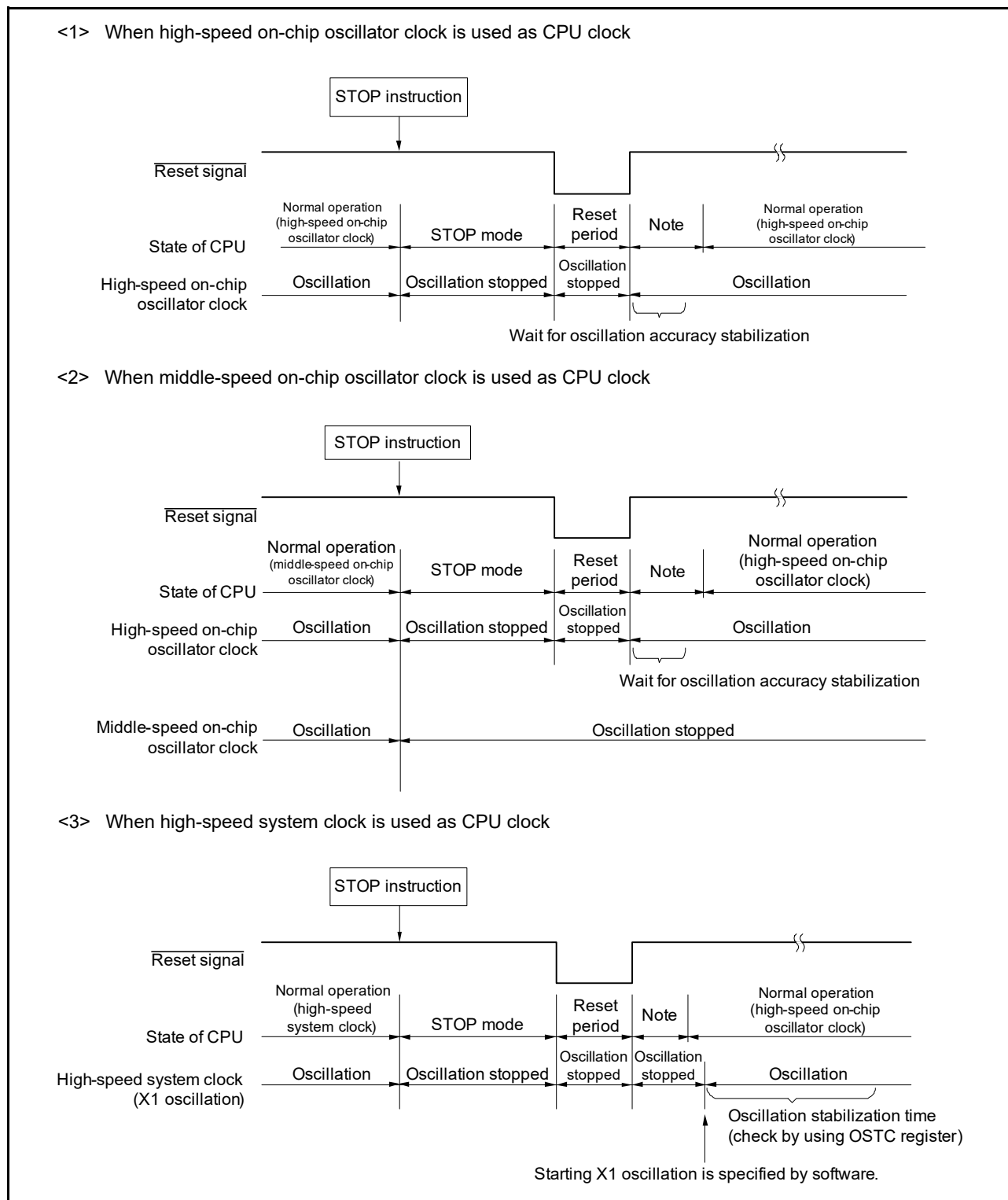
Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

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 27 - 8 STOP Mode Release by Reset



27.3.3 SNOOZE mode

1. SNOOZE mode setting and operating states

The RL78/L23 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 16 A/D Converter (ADC)**
- **Section 19 Serial Array Unit (SAU)**
- **Section 23 Data Transfer Controller (DTC)**
- **Section 34 SNOOZE Mode Sequencer (SMS)**
- **Section 35 Capacitive Sensing Unit (CTS2La)**

Also, the RL78/L23 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 42 Electrical Characteristics**.

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" + 10 to 11 clock cycles
- When vectored interrupt servicing is not carried out:
"0.3 to 0.4 μ s" + 4 to 5 clock cycles

For the middle-speed on-chip oscillator clock^{Note}:

- When vectored interrupt servicing is carried out:
"0.6 to 1.2 μ s" + 10 to 11 clock cycles
- When vectored interrupt servicing is not carried out:
"0.6 to 1.2 μ s" + 4 to 5 clock cycles

Note This is selected when the setting of the MIOTRM register is its initial value.

The operating states in the SNOOZE mode are shown below.

Table 27 - 4 Operating States in the SNOOZE Mode (1/2)

STOP Mode Setting Item			Generation of the Source Conditions Which Lead to Transitions to the SNOOZE Mode during the STOP Mode	
			When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{IH})	When the CPU is Operating with the 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 possible	
		f _{EXS}		
		f _{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of the option byte (000C0H), and the WUTMMCK0 bit of the 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 clock X (fs _X) or XR (fs _{XR}) is the operating clock. 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; however, operation in response to access by the DTC or SMS is possible.	
Port (latch)			Retains the output before the transition to the SNOOZE mode; however, operation in response to access by the DTC, SMS, or ELCL is possible.	
Timer array unit			Operation possible in response to access by the DTC, SMS or ELCL	
Timer RJ			• Operation possible in the event counter mode when the TRJION input with no filter is selected • Operation possible when a subsystem clock is selected as the source for counting and the RTCLPC bit in the OSMC register is 0 • Operation possible when the low-speed on-chip oscillator is selected as the source for counting • Operation prohibited under any conditions other than those listed above	
16-bit timers KB40, KB41, and KB42			Operation prohibited	
RTC			Operation possible	
8-bit interval timers			Operation possible when the RTCLPC bit is 0 (operation prohibited otherwise)	
32-bit interval timer			Operation possible when fs _{XP} is selected and the RTCLPC bit is 0	
External signal sampler			Operation possible when the RTCLPC bit is 0 and the EXSDLPC bit is 1 (operation prohibited otherwise)	
Watchdog timer			See Section 15 Watchdog Timer (WDT) .	
Clock output/buzzer output			Operation possible only when f _{SUBR} is selected	
A/D converter			Operation possible	
D/A converter			Retains the output before the transition to the SNOOZE mode; however, operation in response to access by the DTC, SMS, or ELCL when the RTCLPC bit is 0 is possible.	
Comparator			Operation possible only when the digital filters are not in use	
Serial array unit			Operation possible only for CSI00, CSI20, UART0, and UART2 (operation prohibited otherwise)	
Serial interface IICA			Waking up possible in response to address matching	
Serial interface UARTA			Operation possible when fs _{XP} is selected and the RTCLPC bit is 0	
LCD controller/driver			Operation possible depending on the state of the clock selected as the LCD source clock; operation proceeds or is stopped according to whether the clock is operating or is stopped, respectively.	
Data transfer controller (DTC)			Operation possible	
SNOOZE mode sequencer			Operation possible	
Logic and event link controller (ELCL)			Operation-possible function blocks can be linked.	

Table 27 - 4 Operating States in the SNOOZE Mode (2/2)

STOP Mode Setting Item		Generation of the Source Conditions Which Lead to Transitions to the SNOOZE Mode during the STOP Mode	
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f _{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f _{IM})
Power-on-reset function		Operation possible	
Voltage detection function			
Oscillation stop detection		Operation possible when f _{IL} is oscillating and the RTCLPC bit is 0 (operation prohibited otherwise)	
External interrupt		Operation possible	
Key interrupt function			
Capacitive sensing unit (CTS2La) ^{Note}		Operation possible	
CRC operation function	High-speed CRC	Operation stopped	
	General-purpose CRC	Operation possible in response to access by the DTC or SMS to obtain data for calculations from the RAM area	
Illicit memory access detection function		Operation possible in response to access by the DTC or SMS	
RAM parity error detection function			
RAM guard function			
SFR guard function			
True random number generator			

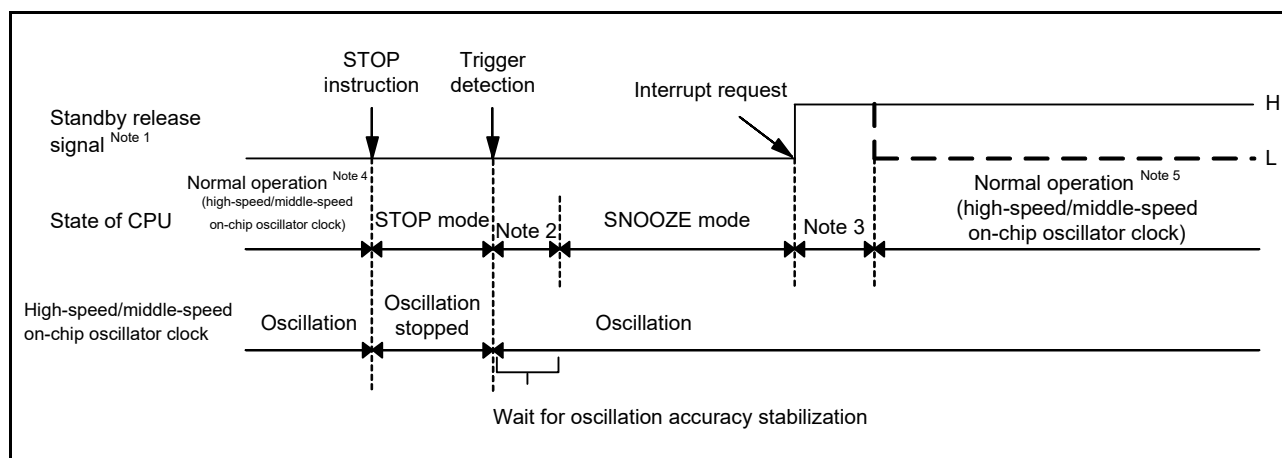
Note The measurement accuracy of the capacitive sensing unit (CTS2La) depends on the accuracy of the operating clock. When CTS2La is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.
Operation prohibited: Stop operation before switching to the STOP mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock fx: 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
f_{SUBR}: Subsystem clock R frequency

2. Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 27 - 9 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

Note 2. Transition time from STOP mode to SNOOZE mode

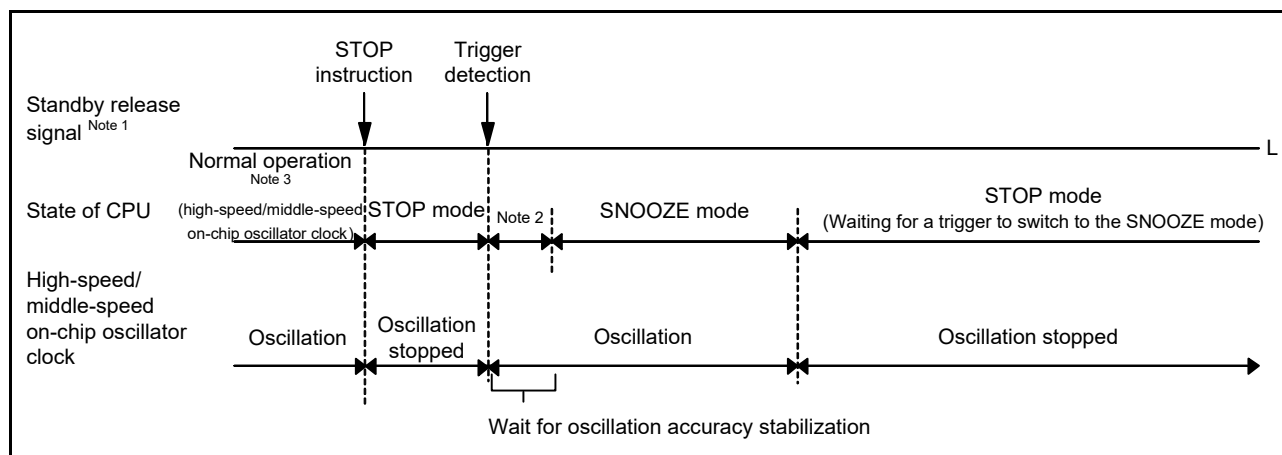
Note 3. Transition time from SNOOZE mode to normal operation

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 27 - 10 When the Interrupt Request Signal is Not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 25 - 1 Basic Configuration of Interrupt Function**.

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 16 A/D Converter (ADC)**
- **Section 19 Serial Array Unit (SAU)**
- **Section 23 Data Transfer Controller (DTC)**
- **Section 34 SNOOZE Mode Sequencer (SMS)**
- **Section 35 Capacitive Sensing Unit (CTSU2La)**

Section 28 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 28 - 1 Operating States during a Reset**.

Note In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the 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 42.4 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.

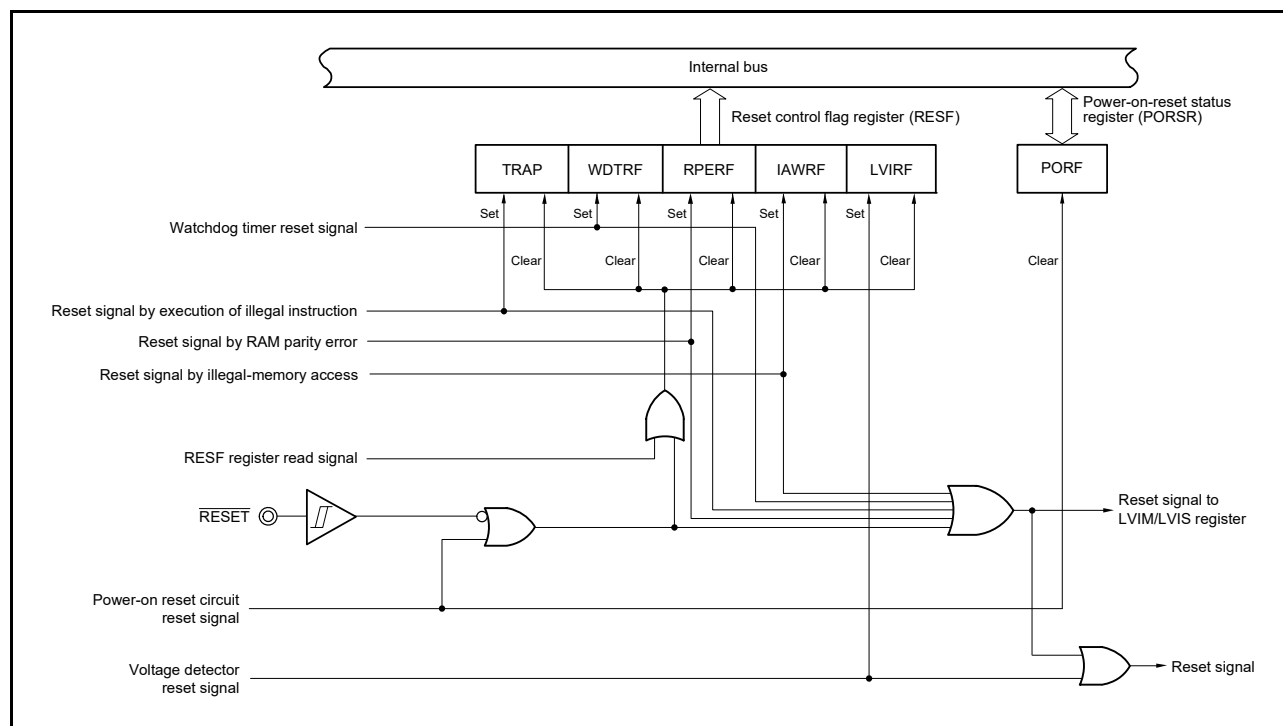
Caution 3. The port pins become the following state because SFRs and 2nd SFRs are initialized after a reset.

- 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, AMPHS1, and AMPHS0 bits of the CMC register

Figure 28 - 1 Block Diagram of Reset Function

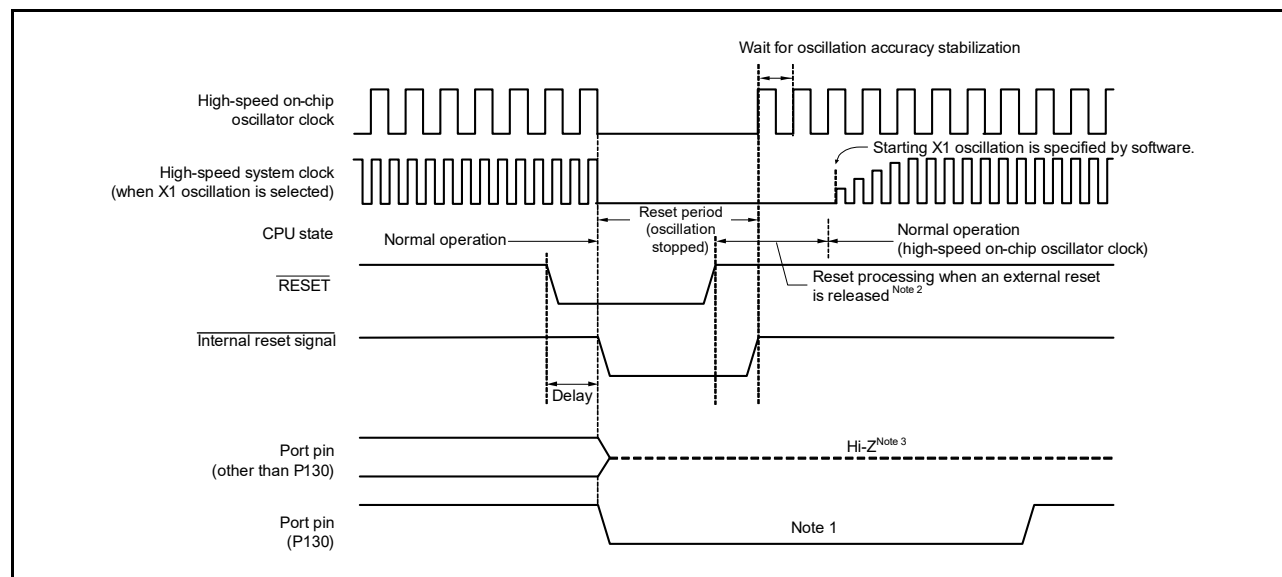


Caution An internal reset by the LVD0 circuit does not reset the LVD0 circuit.

28.1 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.

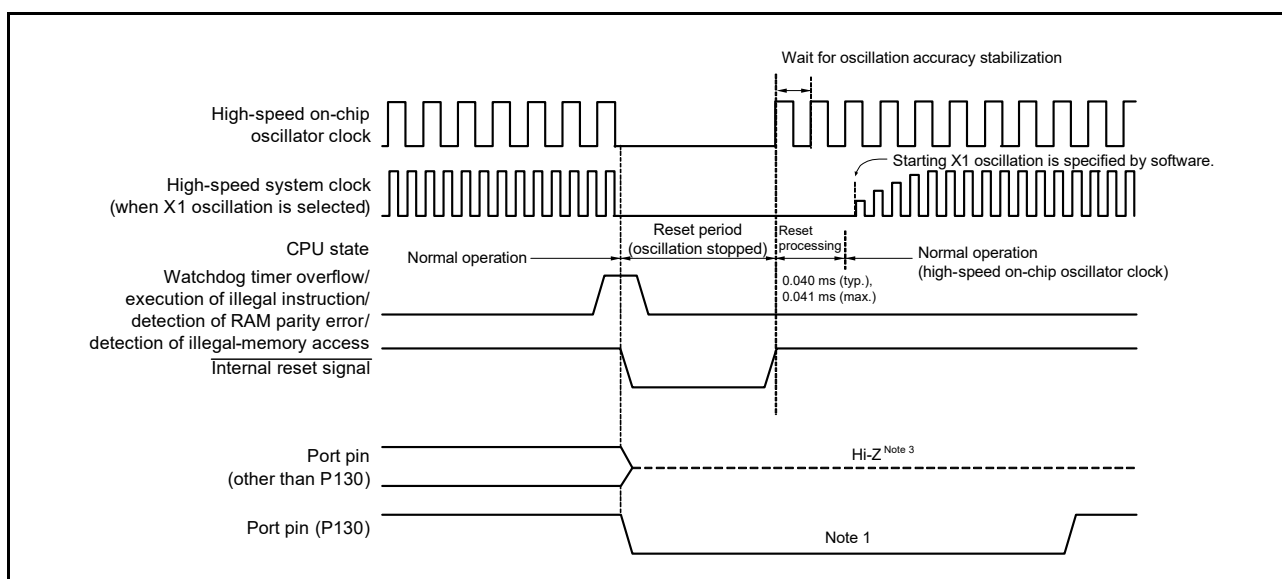
Figure 28 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



(Notes and Remark 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 28 - 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 and Remark 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 29 Power-on-reset Circuit (POR)** or **Section 30 Voltage Detector (LVD)**.

Remark V_{POR} : POR power supply rise detection voltage
 V_{LVD0} : LVD0 detection voltage

Table 28 - 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	fIH	Operation stopped		
	fIM			
	fX	Operation stopped (the X1 and X2 pins are input port mode)		
	fEX	Clock input invalid (the pin is input port mode)		
	Subsystem clock		fXT	Operable (operation stops in the POR reset state, the XT1 and XT2 pins are input port mode)
			fEXS	Operable (operation stops in the POR reset state, the EXCLKS pin is input port mode)
fIL			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.	
Oscillation stop detector			Operation stopped	
32-bit interval timer				
8-bit interval timer				
Watchdog timer				
Clock output/buzzer output				
External signal sampler				
Timer RJ				
16-bit timers KB40, KB41, and KB42				
A/D converter				
D/A converter				
Comparator				
Serial array unit				
Serial interface IICA				
Serial interface UARTA				
LCD controller/driver				
Data transfer controller (DTC)				
SNOOZE mode sequencer				
Logic and event link controller (ELCL)				
Power-on-reset function			Detection operation possible	
Voltage detection function			LVD0: LVD0 operation is possible following an LVD0 reset but is stopped following other types of reset. LVD1: Operation stopped	

Table 28 - 1 Operating States during a Reset (2/2)

Item		Operating State during a Reset
External interrupt		Operation stopped
Key interrupt function		
Capacitive sensing unit (CTSU2La)		
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 28 - 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**.

28.2 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)
- Peripheral reset control register 2 (PRR2)

28.2.1 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 $\overline{\text{RESET}}$ pin
- Reset by the power-on-reset (POR) circuit
- The RESF register is accessed.

Figure 28 - 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 28 - 3**.
 (Note and Cautions continue on the next page.)

Note 2. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. 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 32.3.4 RAM parity error detection.

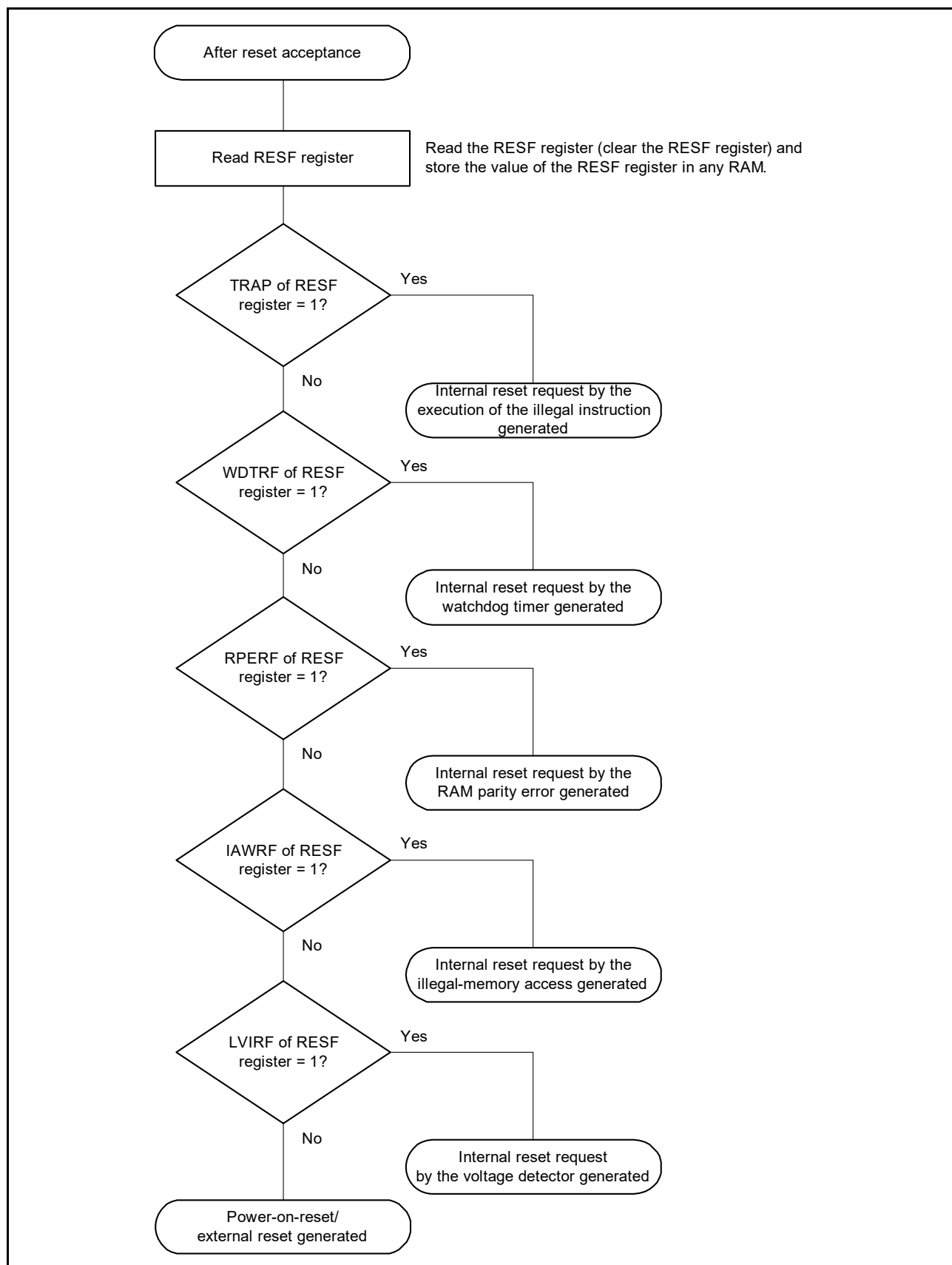
The state of the RESF register when a reset request is generated is shown in **Table 28 - 3**.

Table 28 - 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 to 0	Cleared to 0	Set to 1	Retained	Retained	Retained	Retained
WDTRF			Retained	Set to 1			
RPERF				Retained	Set to 1		
IAWRF					Retained	Set to 1	
LVIRF						Retained	Set to 1

The RESF register is automatically cleared after it is read by using an 8-bit memory manipulation instruction. **Figure 28 - 5** shows the procedure for checking a reset source.

Figure 28 - 5 Example of Procedure for Checking Reset Source



The above flow is an example of the procedure for checking a reset source.

28.2.2 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 is 00H following a power-on reset.

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 28 - 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.

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

Figure 28 - 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	IICA1RES	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 = 6 to 2, 0

Table 28 - 4 shows the on-chip peripheral modules controlled by individual bits.

Table 28 - 4 On-chip Peripheral Modules Controlled by Individual Bits in the PRR0 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
6	IICA1RES	Serial interface IICA1
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 clear bits 7 and 1 to 0.

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

Figure 28 - 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	DACRES	SMSRES	CMPRES	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 = 7 to 4, 0

Table 28 - 5 shows the on-chip peripheral modules controlled by individual bits.

Table 28 - 5 On-chip Peripheral Modules Controlled by Individual Bits in the PRR1 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
7	DACRES	D/A converter
6	SMSRES	SNOOZE mode sequencer
5	CMPRES	Comparator
4	TML32RES	32-bit interval timer
0	CTSURES	CTSU

Caution Be sure to clear bits 3 to 1 to 0.

28.2.5 Peripheral reset control register 2 (PRR2)

The PRR2 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 PRR2 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 28 - 9 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
PRR2	EXSDRES	OSDCRES	TKBRES	0	0	0	TRJ1RES	TRJ0RES

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 = 7 to 5, 1, 0

Table 28 - 6 shows the on-chip peripheral modules controlled by individual bits.

Table 28 - 6 On-chip Peripheral Modules Controlled by Individual Bits in the PRR2 Register

Bit	Bit Name	Controlled On-chip Peripheral Modules
7	EXSDRES	External signal sampler
6	OSDCRES	Oscillation stop detector
5	TKBRES	16-bit timers KB40, KB41, and KB42
1	TRJ1RES	Timer RJ1
0	TRJ0RES	Timer RJ0

Caution Be sure to clear bits 4 to 2 to 0.

Section 29 Power-on-reset Circuit (POR)

29.1 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 (VDD) exceeds the detection voltage (VPOR). Note that the reset state must be retained until the operating voltage becomes in the range defined in **42.4 AC Characteristics**. This is done by utilizing LVD0 or controlling the externally input reset signal.

- Compares supply voltage (VDD) and detection voltage (VPDR), generates internal reset signal when $VDD < VPDR$. 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 **42.4 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 28 Reset Function**.

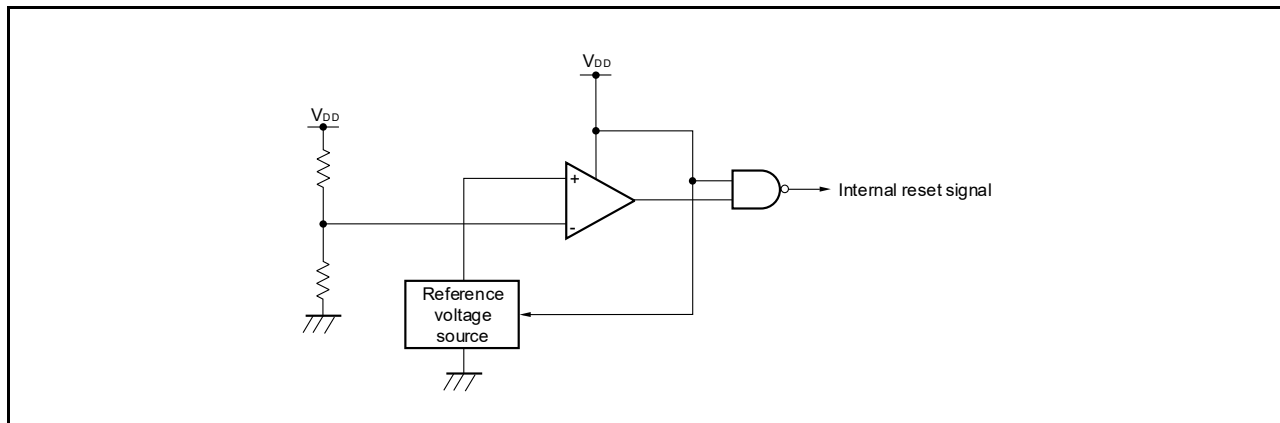
Remark 2. The power-on-reset status register (PORSR) is used to check the occurrence of an internal reset from the power-on-reset circuit. For details of the PORSR register, see **Section 28 Reset Function**.

Remark 3. VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage
For details, see **42.6.5 POR circuit characteristics**.

29.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in **Figure 29 - 1**.

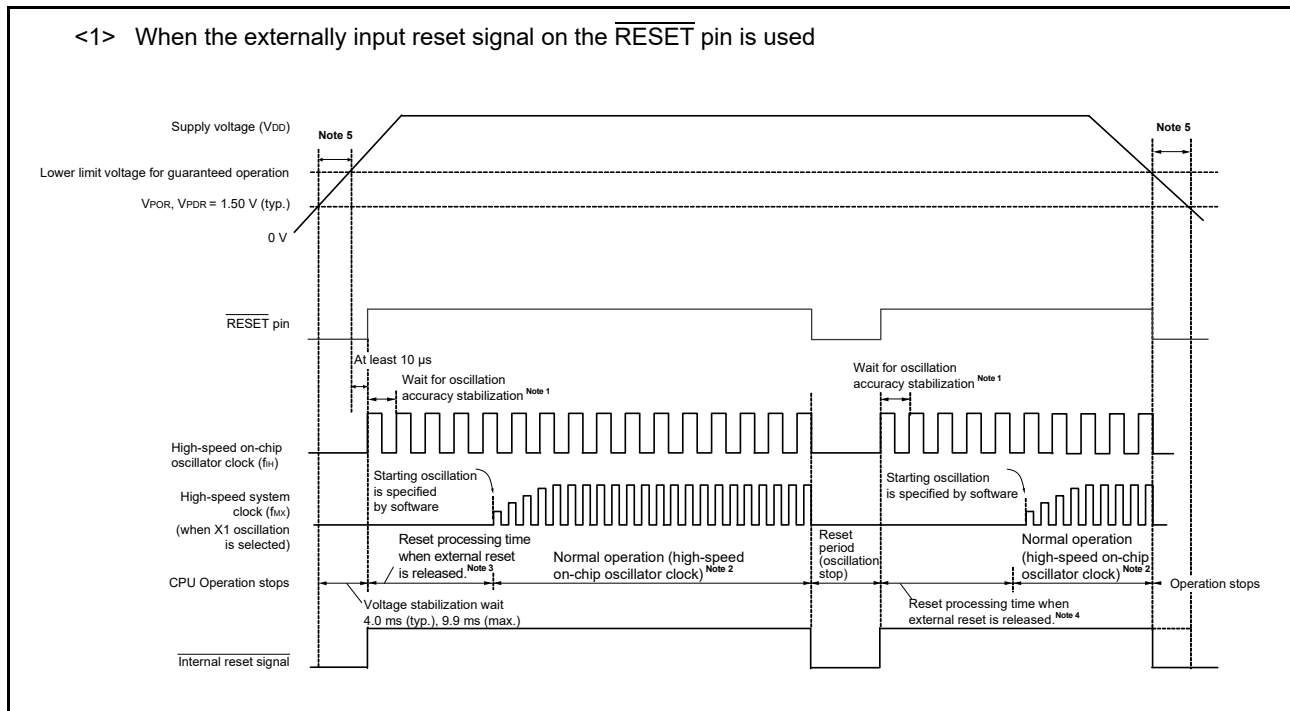
Figure 29 - 1 Block Diagram of Power-on-reset Circuit



29.3 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 29 - 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.)

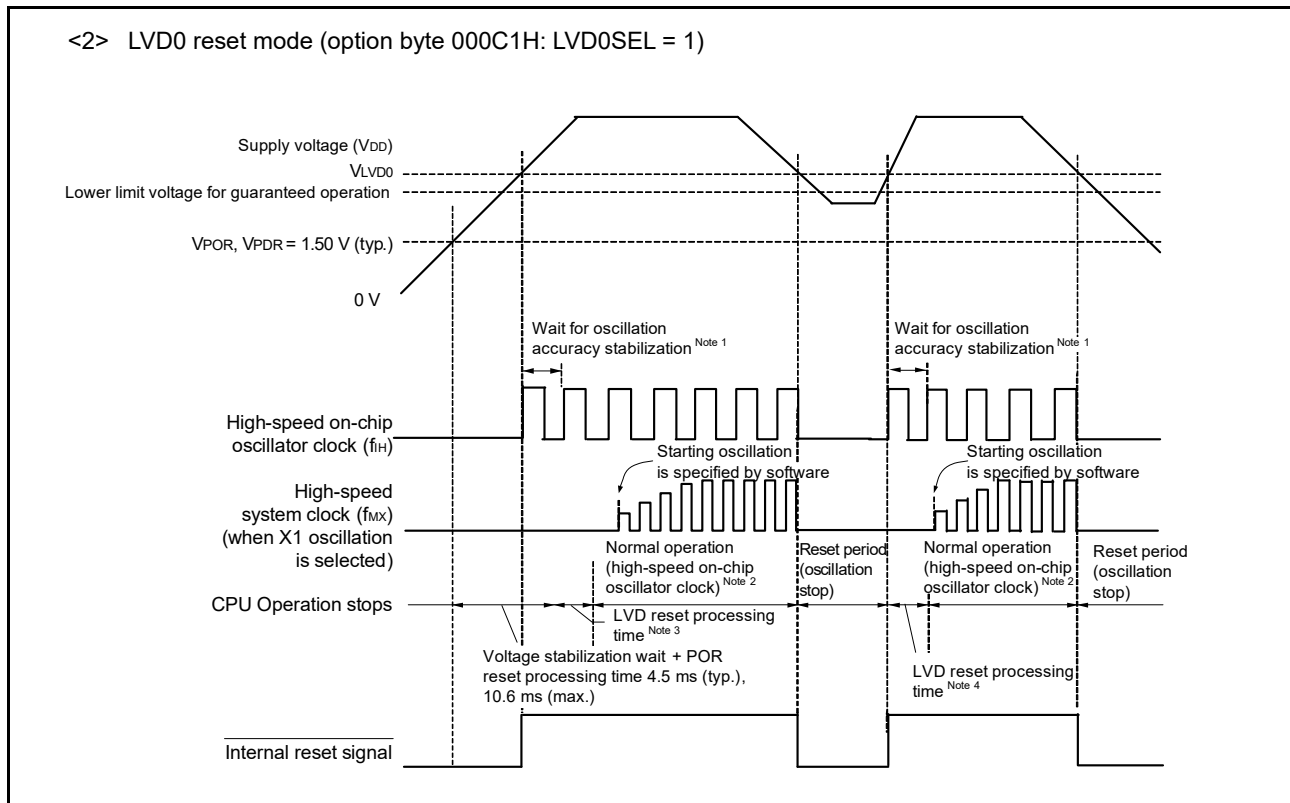
With the LVD circuit not in use: 0.170 ms (typ.), 0.257 ms (max.)

Note 5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **42.4 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 30 Voltage Detector (LVD).

Remark V_{POR} : POR power supply rise detection voltage
 V_{PDR} : POR power supply fall detection voltage

Figure 29 - 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 (VLVD0) is reached as well as the voltage stabilization wait + POR reset processing time after the VPOR (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 (VLVD0) is reached. LVD reset processing time: 0.040 ms (typ.), 0.041 ms (max.)

Remark 1. VLVD0: LVD detection voltage

VPOR: POR power supply rise detection voltage

VPDR: 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 <2> LVD0 reset mode (option byte 000C1H: LVD0SEL = 1) in Figure 29 - 2.

Remark 3. Operation of LVD1 is stopped when power is initially supplied. LVD1 is also stopped by an internal reset.

Section 30 Voltage Detector (LVD)

30.1 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 (VDD) 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 37 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 **42.4 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 $VDD \geq VLVD0$. Generates an internal reset on detecting $VDD < VLVD0$ and retains the reset state until $VDD \geq VLVD0$ is detected.	Generates an internal reset on detecting $VDD < VLVD1$ after LVD1 operation has been enabled.	Retains the state of an internal reset by the LVD immediately after a reset until $VDD \geq VLVD0$. Releases the LVD internal reset by detecting $VDD \geq VLVD0$. Generates an interrupt request signal (INTLVI) by detecting $VDD < VLVD0$ or $VDD \geq VLVD0$ after the LVD internal reset is released.	Generates an interrupt request signal (INTLVI) on detecting $VDD < VLVD1$ after LVD1 operation has been enabled. After the first detection, generates an interrupt request signal (INTLVI) on detecting $VDD < VLVD1$ or $VDD \geq VLVD1$.

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 28 Reset Function**.

30.2 Configuration of Voltage Detector

Figures 30 - 1 and 30 - 2 are block diagrams of the voltage detectors.

Figure 30 - 1 Block Diagram of LVD0

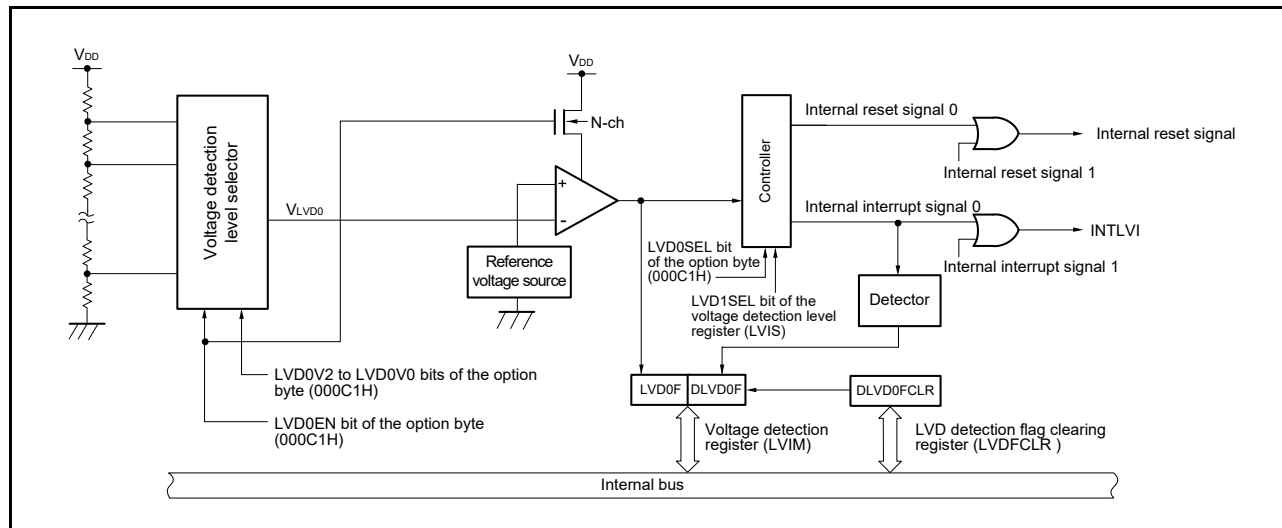
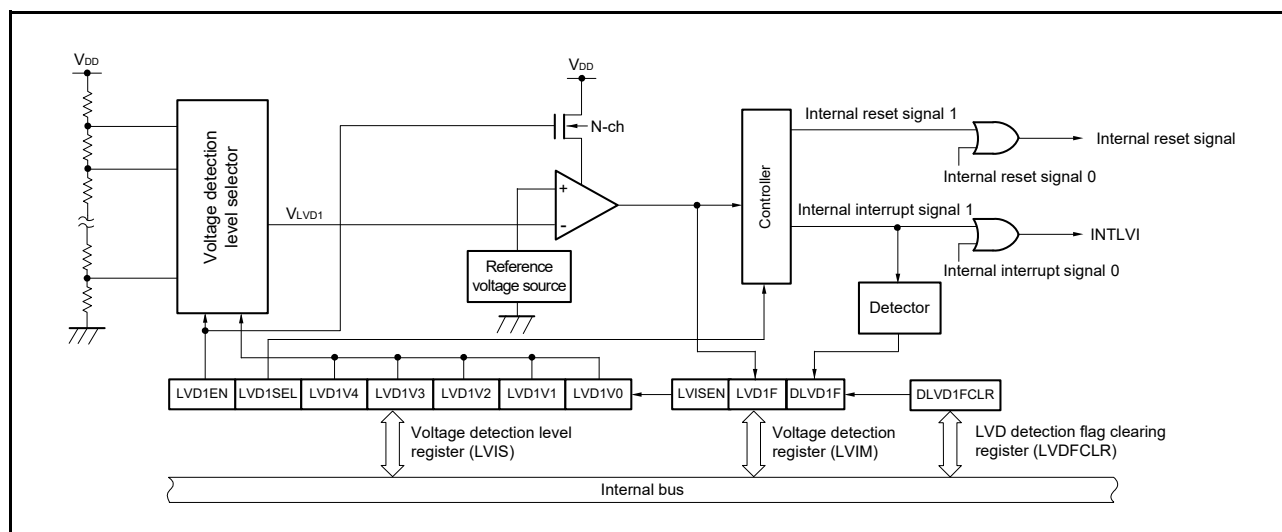


Figure 30 - 2 Block Diagram of LVD1



30.3 Registers for Controlling the Voltage Detector

The following registers are used to control the voltage detector.

- User option byte (000C1H/040C1H): See **Section 37 Option Bytes**.
- Voltage detection register (LVIM)
- LVD detection flag clearing register (LVDFCLR)
- Voltage detection level register (LVIS)

30.3.1 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 30 - 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 (VDD) ≥ detection voltage (VLVDn), or when LVD is off							
1	Supply voltage (VDD) < detection voltage (VLVDn)							

Note 1. The value after a reset is 01H when LVD0 operation is enabled and the power supply voltage (VDD) is less than the detection voltage (VLVD0).

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.

30.3.2 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 30 - 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	Clearing DLVD1F	
Note		
0	No effect	
1	Writing 1 to this bit clears the DLVD1F flag.	

DLVD0FCLR	Clearing DLVD0F	
Note		
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.

30.3.3 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 30 - 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, 2}		
						Rising edge	Falling edge
1	1	1	1	1	VLVD117	1.67 V	1.63 V
1	1	1	1	0	VLVD116	1.78 V	1.74 V
1	1	1	0	1	VLVD115	1.88 V	1.84 V
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. When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of LVD0.

Note 2. 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 is listed on the next page.)

Caution When the values in the LVIS register are to be changed, do so according to the procedure described in Figure 30 - 10 Example of Software Processing If Supply Voltage Fluctuation Is 50 ms or Less in Vicinity of the Detection Voltages for LVD0 or LVD1 and Figure 30 - 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.

30.4 Operation of Voltage Detector

30.4.1 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.

Figure 30 - 6 shows the timing of generation of the internal reset signal from LVD0 and **Figure 30 - 7** shows that of the signal from LVD1.

Figure 30 - 6 Timing of Generation of the Internal Reset Signal from LVD0

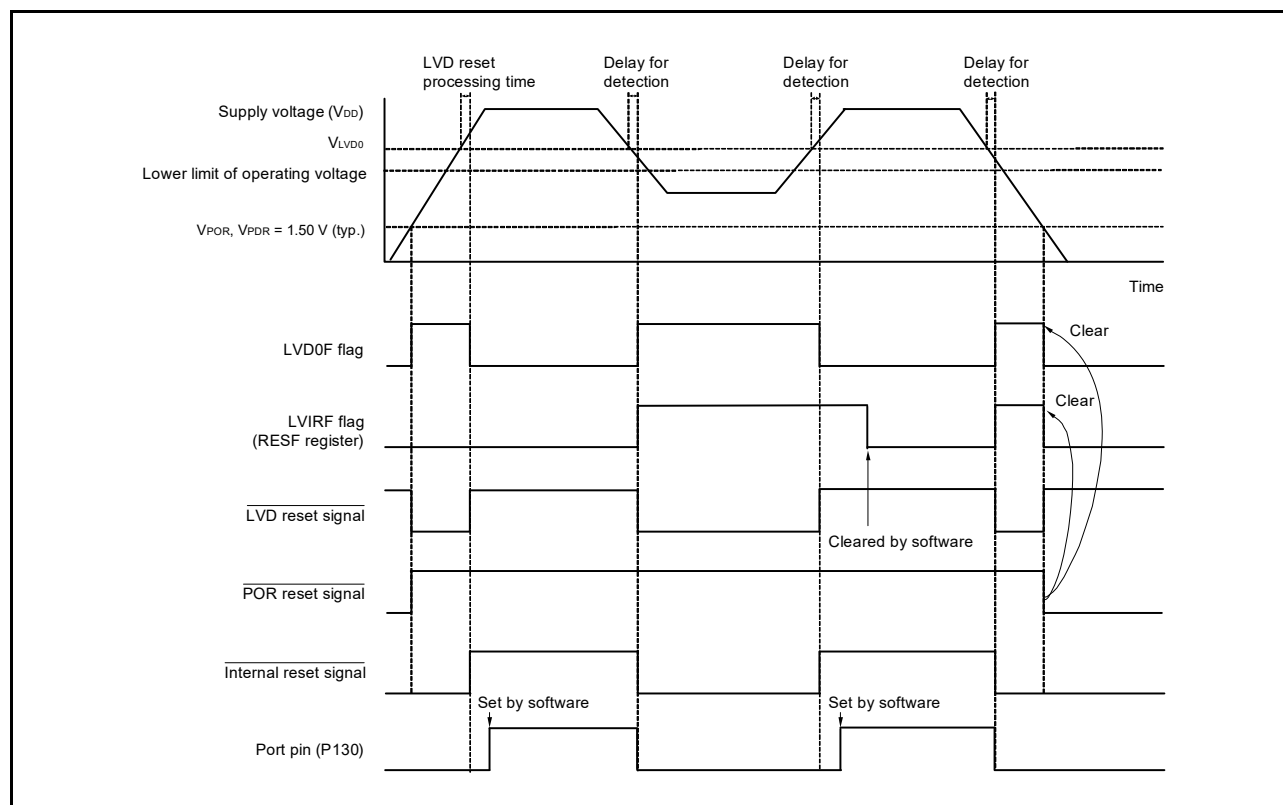
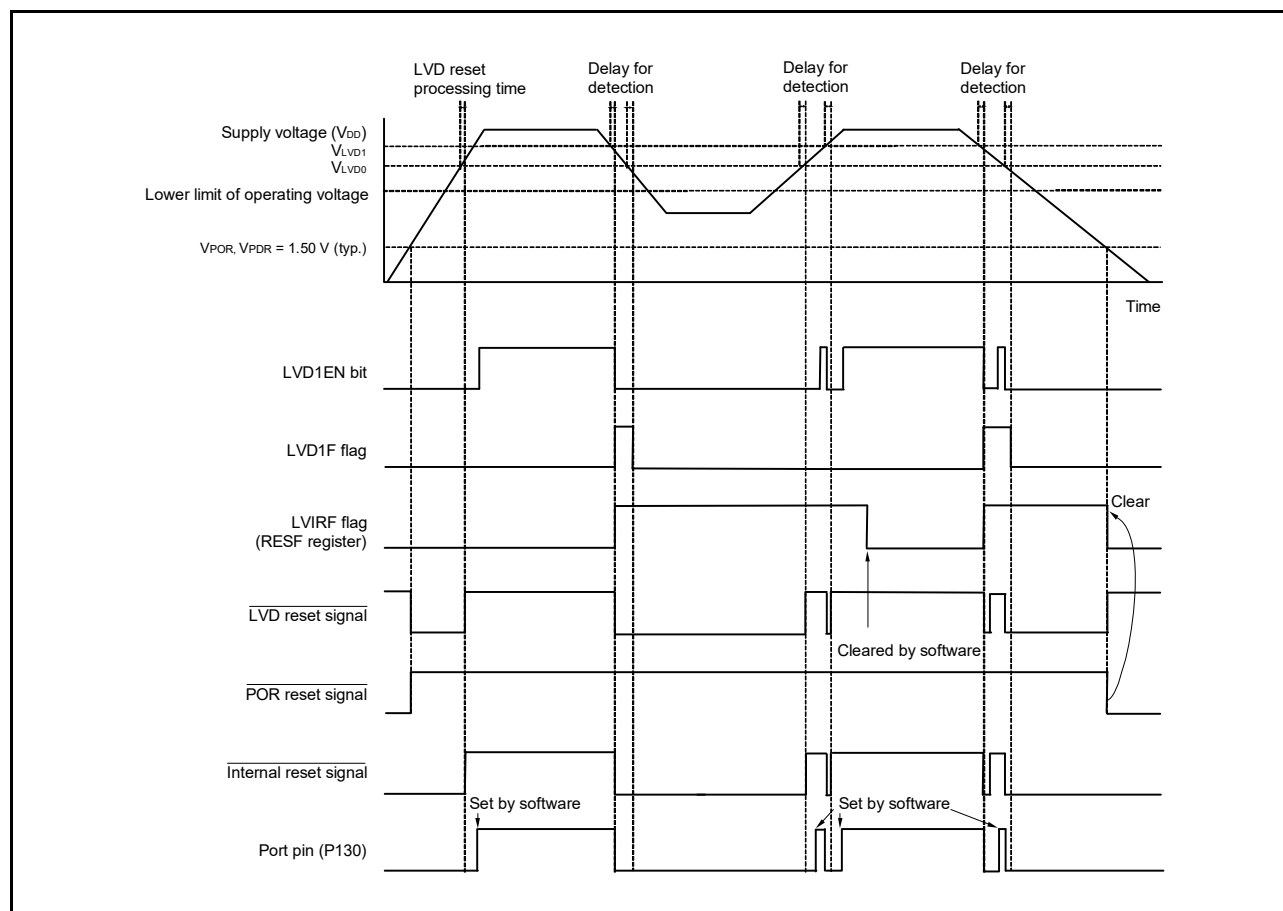


Figure 30 - 7 Timing of Generation of the Internal Reset Signal from LVD1



Remark LVD0: Reset mode

30.4.2 When used as interrupt mode

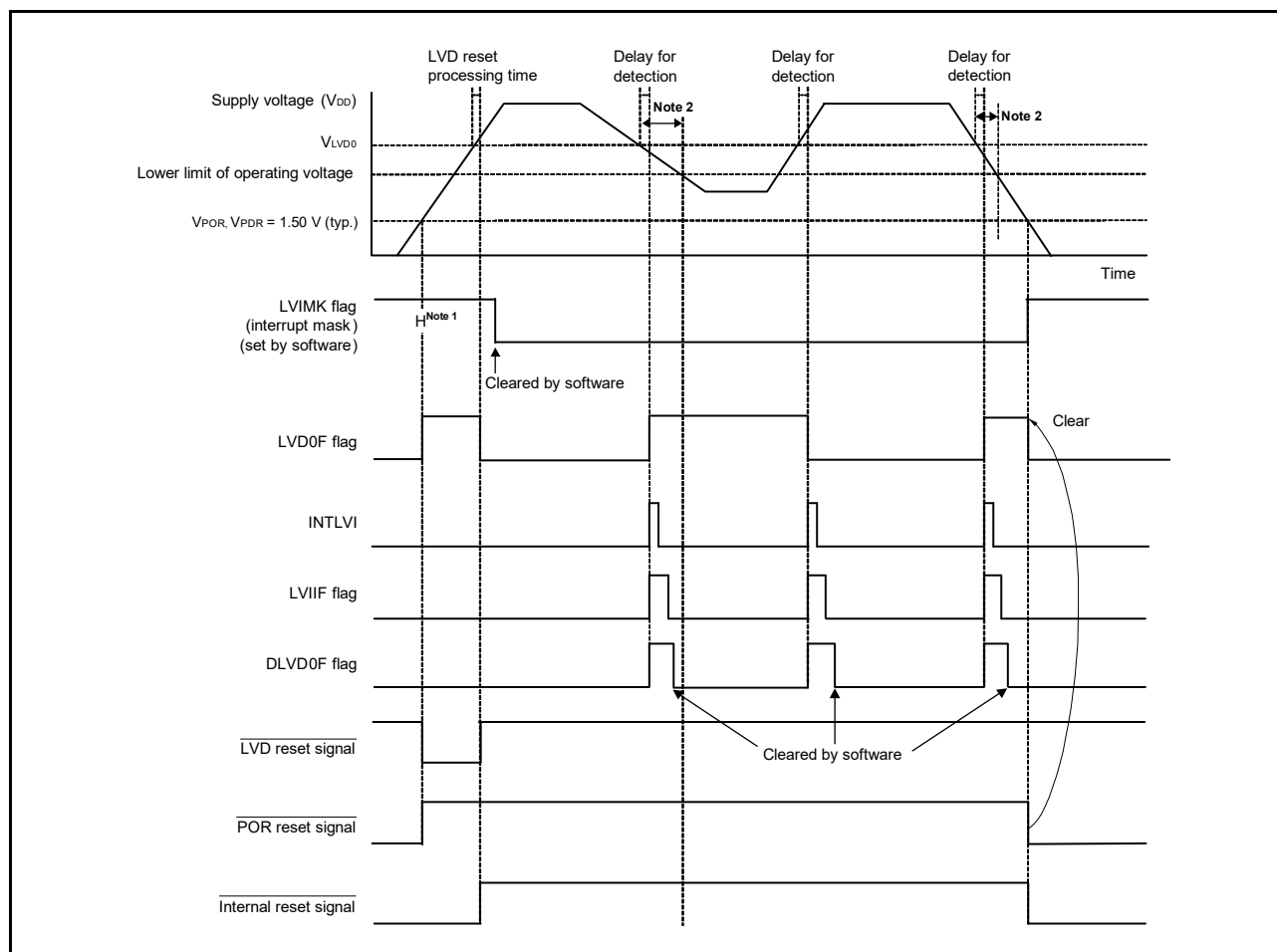
Enabling, selecting the operation mode (interrupt mode: LVD0SEL = 0), and setting the detection voltage (VLVD0) for LVD0 are done by using an option byte (000C1H). On the other hand, enabling, selecting the operation mode (interrupt mode: LVD1SEL = 0), and setting the detection voltage (VLVD1) for LVD1 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 **42.4 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.

Figure 30 - 8 shows the timing of generation of the interrupt signal from LVD0 and **Figure 30 - 9** shows that of the signal from LVD1.

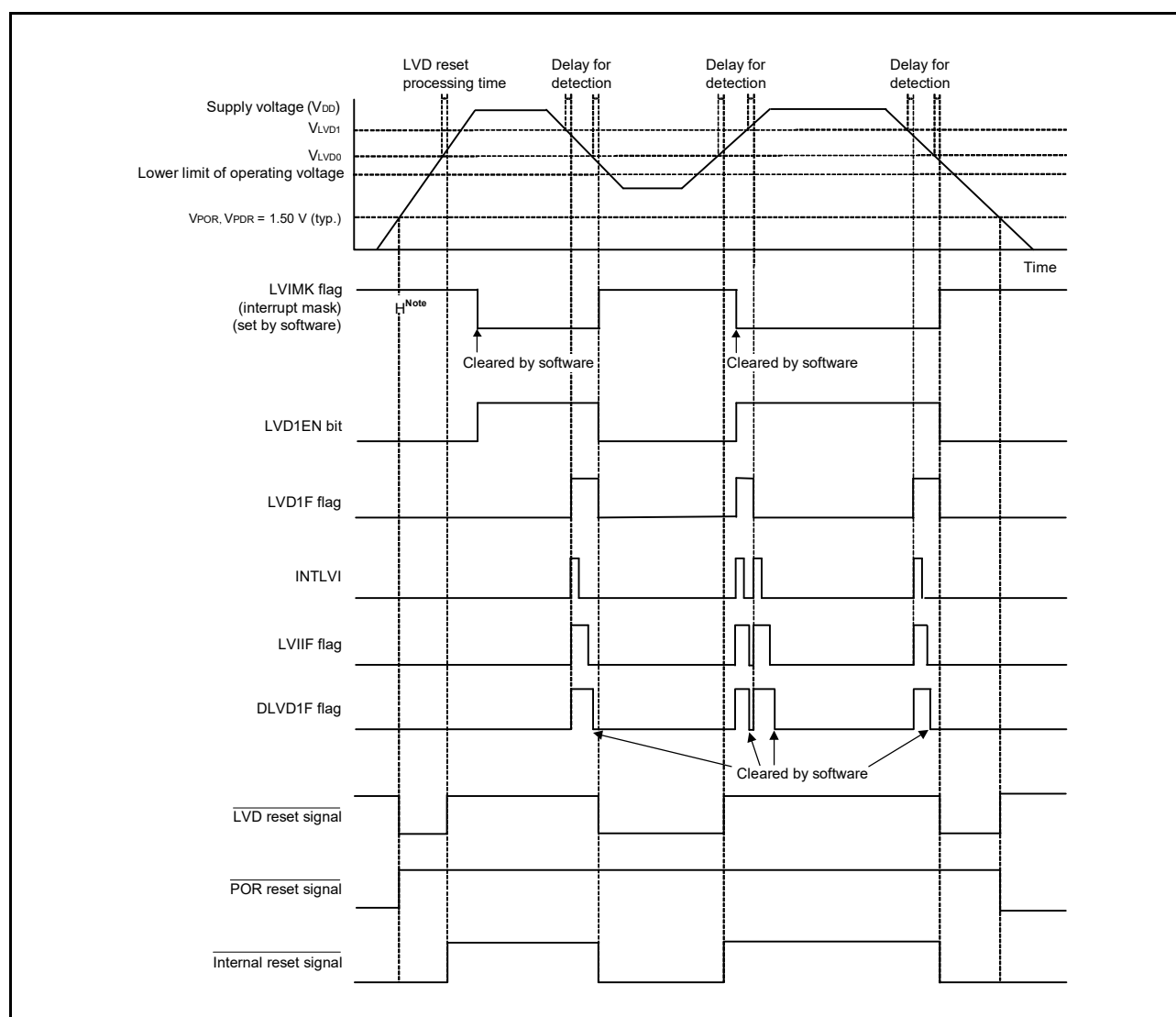
Figure 30 - 8 Timing of Generation of the Interrupt Signal from LVD0



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 **42.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

Figure 30 - 9 Timing of Generation of the Interrupt Signal from LVD1



Note The LVIMK flag is set to 1 by reset signal generation.

Remark LVD0: Reset mode

30.5 Points for Caution when the Voltage Detector Is to be Used

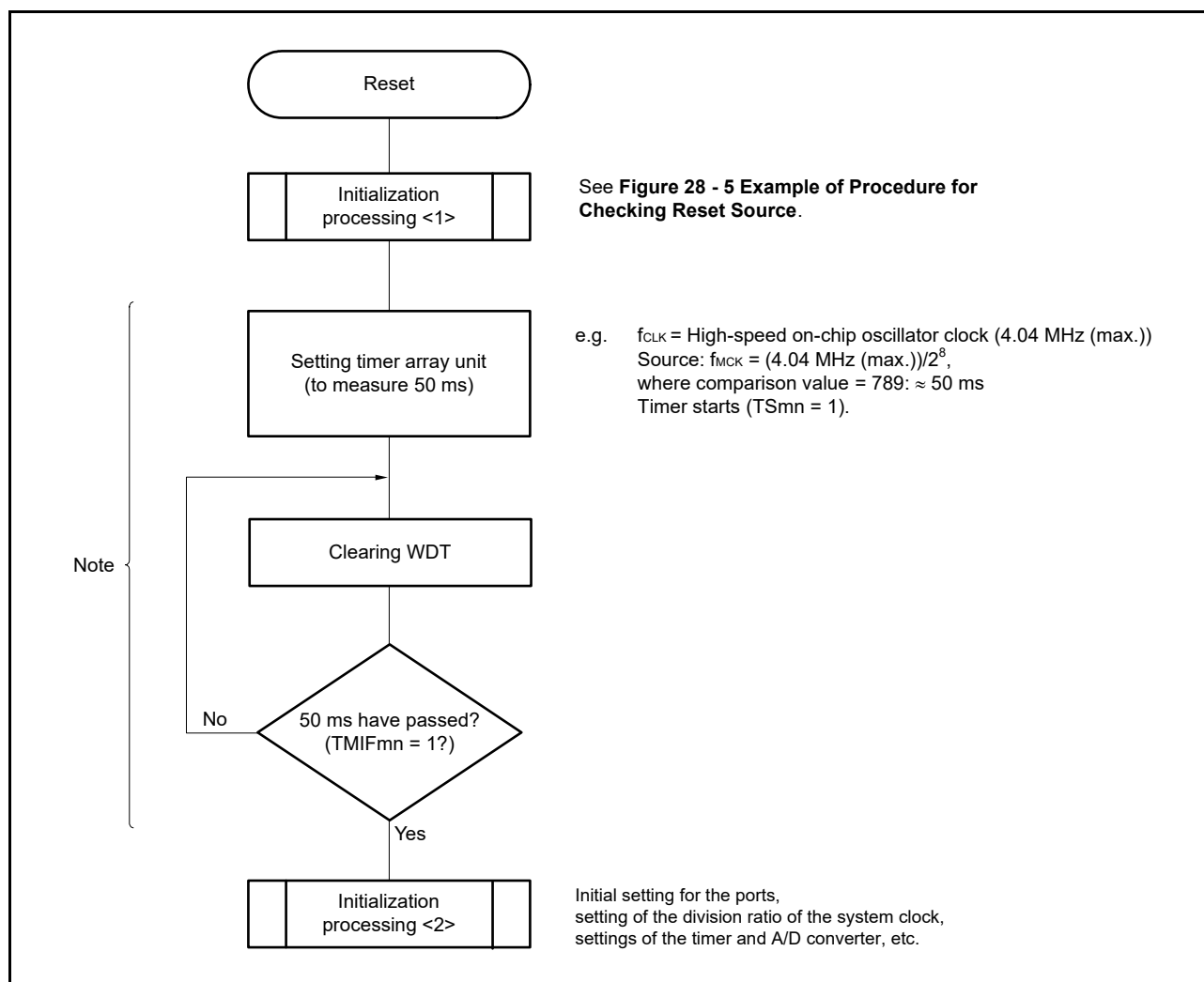
1. Voltage fluctuation when power is supplied

In a system where the supply voltage (VDD) 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 30 - 10 Example of Software Processing If Supply Voltage Fluctuation Is 50 ms or Less in Vicinity of the Detection Voltages for LVD0 or LVD1



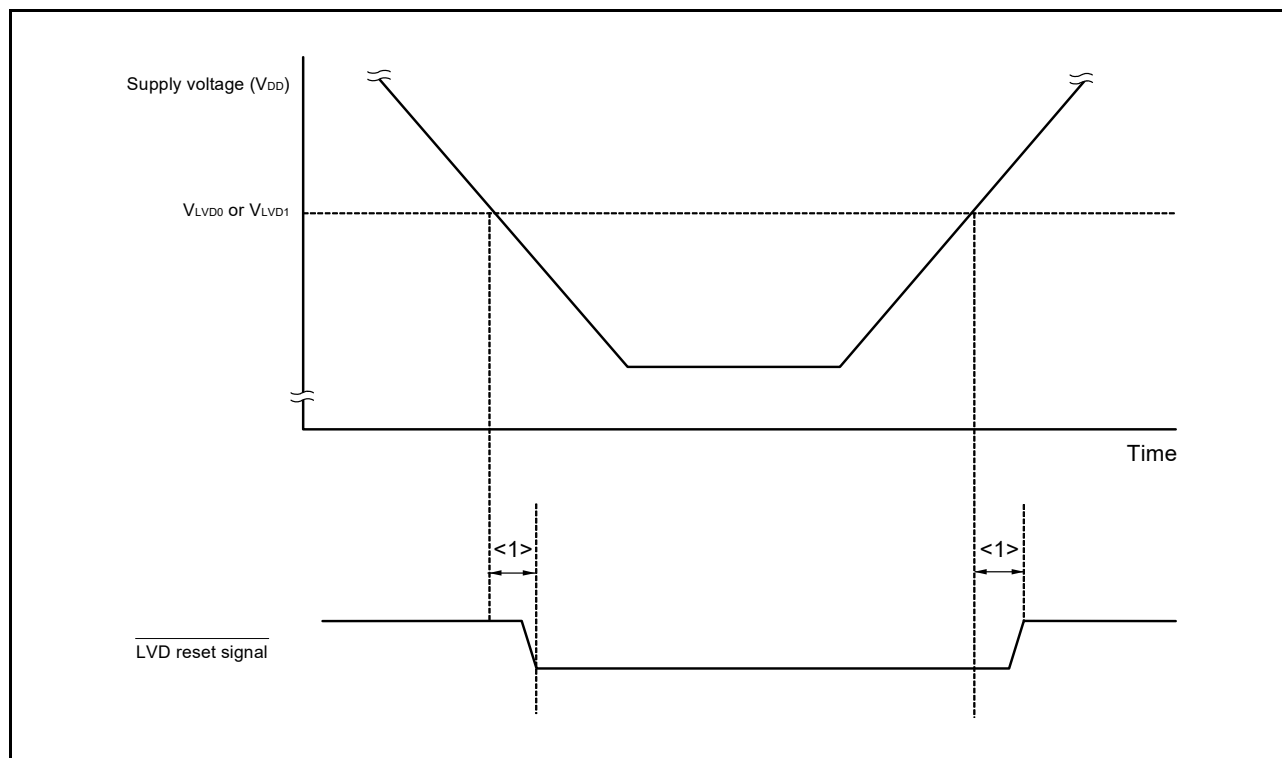
Note If reset is generated again during this period, initialization processing <2> is not started.

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 30 - 11**.

Figure 30 - 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



Remark $\langle 1 \rangle$: 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 **42.4 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 **42.4 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 1500 μs) has elapsed.

<1> Set the LVISEN 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 LVISEN bit of the LVIM register to 0.

Section 31 Oscillation Stop Detector

31.1 Functions of Oscillation Stop Detector

The oscillation stop detector monitors the operating state of the subsystem clock X (fsx) with the low-speed on-chip oscillator clock (fil). If it detects that operation has stopped longer than a predefined interval, it judges that an error has occurred in an XT1 oscillator and outputs the oscillation stop detection interrupt signal.

If the oscillation stop detector is to be used, enable it by software after release from the reset state.

Detection of oscillation having stopped by the oscillation stop detector is stopped by software, a reset from the $\overline{\text{RESET}}$ pin, or an internal reset due to the watchdog timer, power-on reset, voltage detection by the voltage detector (LVD0, 1), execution of an illegal instruction^{Note}, generation of a RAM parity error, or illegal memory access. Furthermore, oscillation of the XT1 oscillator clock is also stopped due to the generation of an internal reset by a power-on reset. Therefore, after this reset, use software to resume oscillation of XT1 oscillator clock before enabling oscillation stop detection.

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.

The period used by the oscillation stop detector to judge that oscillation has stopped (oscillation stop judgment time) can be set by using the OSDCCMP11 to OSDCCMP0 bits of the oscillation stop detection control register (OSDC).

Oscillation stop judgment time = Low-speed on-chip oscillator clock (fil) cycle/2 × ((setting of OSDCCMP11 to OSDCCMP0) + 1)

- OSDCCMP11 to OSDCCMP0 = 003H: 212 μs (min.), 244 μs (typ.), 287 μs (max.)
- OSDCCMP11 to OSDCCMP0 = FFFH: 217 ms (min.), 250 ms (typ.), 294 ms (max.)

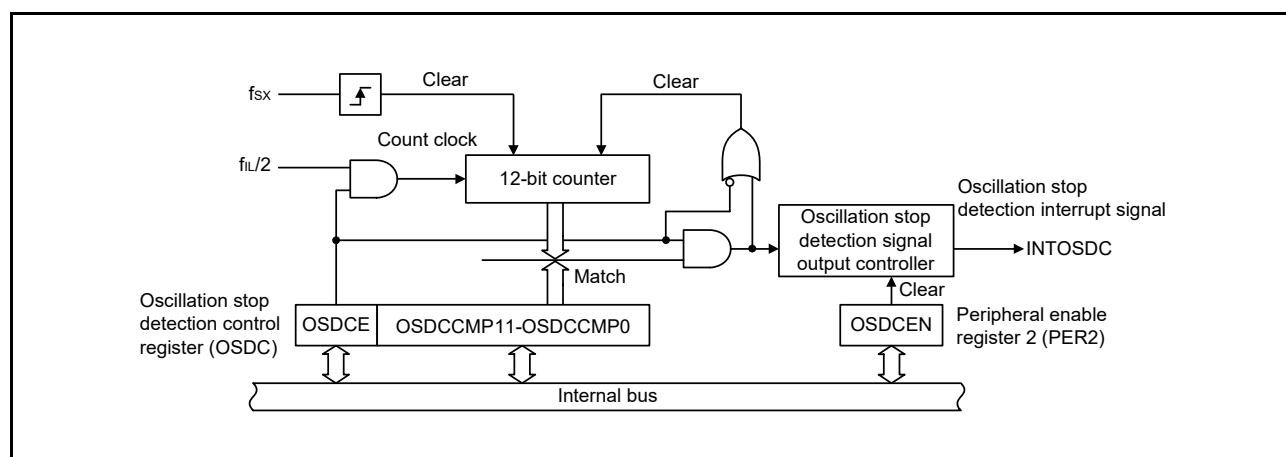
31.2 Configuration of Oscillation Stop Detector

The oscillation stop detector includes the following hardware.

Table 31 - 1 Configuration of Oscillation Stop Detector

Item	Configuration
Control registers	- Peripheral enable register 2 (PER2) - Peripheral reset control register 2 (PRR2) - Oscillation stop detection control register (OSDC)

Figure 31 - 1 Block Diagram of Oscillation Stop Detector



31.3 Registers for Controlling the Oscillation Stop Detector

The following registers are used to control the oscillation stop detector.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Oscillation stop detection control register (OSDC)

31.3.1 Peripheral enable register 2 (PER2)

The PER2 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 the oscillation stop detector, be sure to set bit 6 (OSDCEN) of this register to 1. The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PER2 register following a reset is 00H.

Figure 31 - 2 Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER2	EXSDEN	OSDCEN	TKBEN	UTA3EN	UTA2EN	UTA1EN	TRJ1EN	TRJ0EN
OSDCEN		Control of supply of an input clock to the oscillation stop detector						
0		Stops supply of an input clock. • The SFRs used by the oscillation stop detector cannot be written. When an SFR used by the oscillation stop detector is read, the value returned is 00H. However, the SFR is not actually initialized. Note						
1		Enables supply of an input clock. • The SFRs used by the oscillation stop detector can be read and written.						

Note To initialize the oscillation stop detector and the SFRs used by the oscillation stop detector, use bit 6 (OSDCRES) of PRR2.

Caution When setting the oscillation stop detector, be sure to set OSDCEN to 1 first. If OSDCEN = 0, writing to a control register of the oscillation stop detector is ignored, and all values read are the default values.

31.3.2 Peripheral reset control register 2 (PRR2)

The PRR2 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 (OSDCRES) to 1 to reset the oscillation stop detector. The PRR2 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 31 - 3 Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH

After reset: 00H

R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
PRR2	EXSDRES	OSDCRES	TKBRES	0	0	0	TRJ1RES	TRJ0RES
	Control resetting of the oscillation stop detector							
	0	The oscillation stop detector is released from the reset state.						
	1	The oscillation stop detector is in the reset state. The SFRs for use with the oscillation stop detector are initialized.						

Caution Be sure to clear bits 4 to 2 to 0.

31.3.3 Oscillation stop detection control register (OSDC)

The OSDC register is used to control the oscillation stop detector. Use this register to start and stop operation of the oscillation stop detector. This register can also be used to specify the oscillation stop judgment time. Operation of the oscillation stop detector cannot be started while the OSDCE bit is 0. The OSDC register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0FFFH.

Figure 31 - 4 Format of Oscillation Stop Detection Control Register (OSDC)

Address: F02D0H

After reset: 0FFFH

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
OSDC	OSDCE	0	0	0	OSDCCMP11	OSDCCMP10	OSDCCMP9	OSDCCMP8
	7	6	5	4	3	2	1	0
	OSDCCMP7	OSDCCMP6	OSDCCMP5	OSDCCMP4	OSDCCMP3	OSDCCMP2	OSDCCMP1	OSDCCMP0
	OSDCE	Control of oscillation stop detector operation						
	0	Stops operation of the oscillation stop detector.						
	1	Starts operation of the oscillation stop detector.						
	OSDCCMP11 to OSDCCMP0		Oscillation stop judgment time					
	000H ... 002H		Setting prohibited					
	003H ... FFFH		These bits specify the oscillation stop judgment time. The oscillation stop detector judges that oscillation has stopped when oscillation has stopped for (A - 2) to (A + 1) clock cycles, where A indicates the value set in the OSDCCMP bits. Oscillation stop judgment time = Low-speed on-chip oscillator clock (f _{IL}) cycle/2 × ((setting of OSDCCMP11 to OSDCCMP0) + 1)					

Caution 1. Be sure to set the OSDCE bit to 0 (to stop operation of the oscillation stop detector) before changing the setting of the OSDCCMP11 to OSDCCMP0 bits.

Caution 2. Detection of oscillation having stopped by the oscillation stop detector is stopped by setting the OSDCE bit to 0 (to stop operation of the oscillation stop detector) by software, a reset from the RESET pin, or an internal reset due to the watchdog timer, power-on reset, voltage detection by the voltage detector (LVD0, 1), execution of an illegal instruction^{Note}, generation of a RAM parity error, or illegal memory access.

Furthermore, oscillation of the XT1 oscillator clock is also stopped due to the generation of an internal reset by a power-on reset. Therefore, after this reset, use software to resume oscillation of XT1 oscillator clock before enabling oscillation stop detection.

Caution 3. Be sure to set bits 14 to 12 to 0.

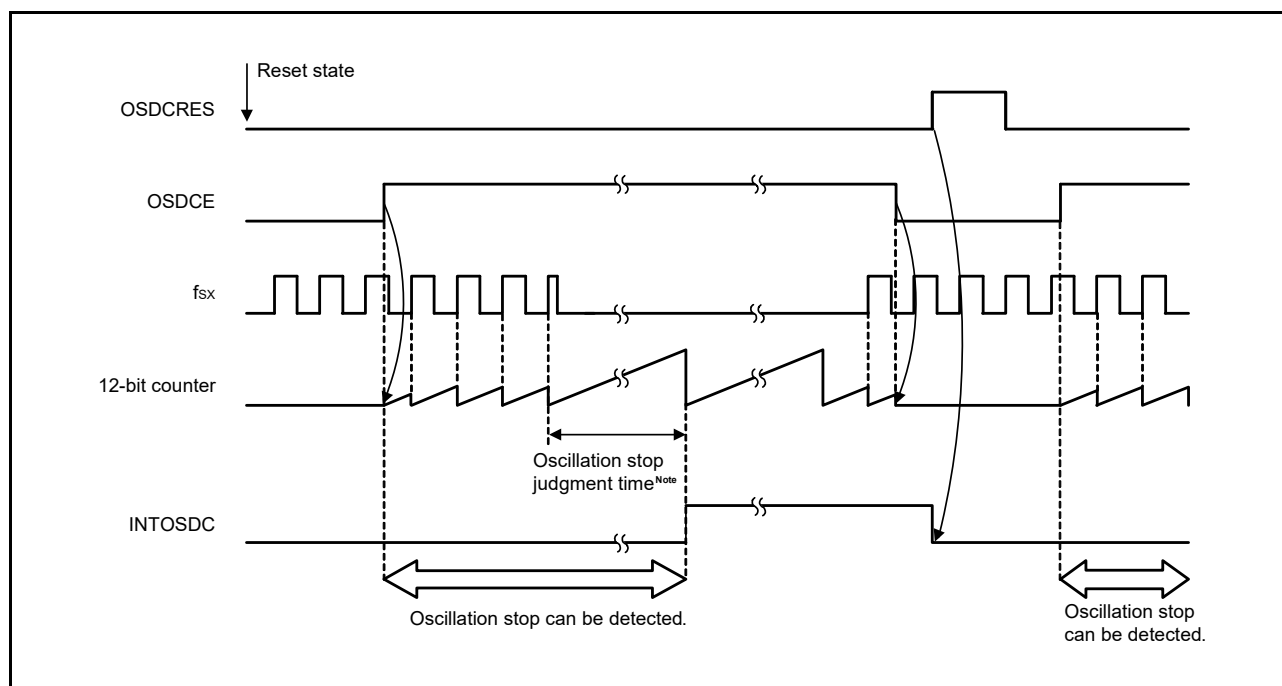
Note In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the in-circuit emulator or on-chip debugging emulator.

31.4 Operation of Oscillation Stop Detector

31.4.1 How the oscillation stop detector operates

1. The subsystem clock starts oscillating after release from the external reset state.
2. The oscillation stop detector starts operating by writing to the oscillation stop detection control register (OSDC).
3. While the oscillation stop detector is operating, if oscillation of the subsystem clock has stopped for a period equal to the oscillation stop judgment time or longer, the oscillation stop detector outputs the oscillation stop detection interrupt signal (INTOSDC).

Figure 31 - 5 Timing of Oscillation Stop Detection by Oscillation Stop Detector



Note The oscillation stop detector judges that oscillation has stopped when oscillation has stopped for $(A - 2)$ to $(A + 1)$ clock cycles, where A indicates the value set in the OSDCCMP bits.

31.5 Points for Caution when the Oscillation Stop Detector Is to be Used

Use the oscillation stop detector along with the watchdog timer.

Oscillation stop detection can be used under the following conditions:

- Bit 0 (WDSTBYON) and bit 4 (WDTON) of the option byte (000C0H) are set to 1 and bit 4 (WUTMMCK0) of the OSMC register is set to 0.

Section 32 Safety Functions

32.1 Overview of Safety Functions

The RL78/L23 provides the following safety functions to comply with the IEC60730 safety standard. 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/L23 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 3, m = 0 to 3

Remark 2. For usage examples of the safety functions complying with the IEC60730 safety standard, refer to the **application notes IEC60730/60335 Self Test Library for RL78 MCU**.

32.2 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 **32.3 Operation of Safety Functions**.

32.3 Operation of Safety Functions

32.3.1 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/L23 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.

32.3.1.1 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 32 - 1 Format of Flash Memory CRC Control Register (CRC0CTL) (1/2)

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 0FFFBH (64 Kbytes - 4 bytes)
0	0	0	1	0	0	0000H to 13FFBH (80 Kbytes - 4 bytes)
0	0	0	1	0	1	0000H to 17FFBH (96 Kbytes - 4 bytes)
0	0	0	1	1	0	0000H to 1BFFBH (112 Kbytes - 4 bytes)
0	0	0	1	1	1	0000H to 1FFFBH (128 Kbytes - 4 bytes)
0	0	1	0	0	0	0000H to 23FFBH (144 Kbytes - 4 bytes)
0	0	1	0	0	1	0000H to 27FFBH (160 Kbytes - 4 bytes)
0	0	1	0	1	0	0000H to 2BFFBH (176 Kbytes - 4 bytes)
0	0	1	0	1	1	0000H to 2FFFBH (192 Kbytes - 4 bytes)
0	0	1	1	0	0	0000H to 33FFBH (208 Kbytes - 4 bytes)
0	0	1	1	0	1	0000H to 37FFBH (224 Kbytes - 4 bytes)
0	0	1	1	1	0	0000H to 3BFFBH (240 Kbytes - 4 bytes)
0	0	1	1	1	1	0000H to 3FFFBH (256 Kbytes - 4 bytes)
0	1	0	0	0	0	0000H to 43FFBH (272 Kbytes - 4 bytes)
0	1	0	0	0	1	0000H to 47FFBH (288 Kbytes - 4 bytes)
0	1	0	0	1	0	0000H to 4BFFBH (304 Kbytes - 4 bytes)
0	1	0	0	1	1	0000H to 4FFFBH (320 Kbytes - 4 bytes)
0	1	0	1	0	0	0000H to 53FFBH (336 Kbytes - 4 bytes)
0	1	0	1	0	1	0000H to 57FFBH (352 Kbytes - 4 bytes)

Figure 32 - 1 Format of Flash Memory CRC Control Register (CRC0CTL) (2/2)

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	1	0	1	1	0	0000H to 5BFFBH (368 Kbytes - 4 bytes)
0	1	0	1	1	1	0000H to 5FFFBH (384 Kbytes - 4 bytes)
0	1	1	0	0	0	0000H to 63FFBH (400 Kbytes - 4 bytes)
0	1	1	0	0	1	0000H to 67FFBH (416 Kbytes - 4 bytes)
0	1	1	0	1	0	0000H to 6BFFBH (432 Kbytes - 4 bytes)
0	1	1	0	1	1	0000H to 6FFFBH (448 Kbytes - 4 bytes)
0	1	1	1	0	0	0000H to 73FFBH (464 Kbytes - 4 bytes)
0	1	1	1	0	1	0000H to 77FFBH (480 Kbytes - 4 bytes)
0	1	1	1	1	0	0000H to 7BFFBH (496 Kbytes - 4 bytes)
0	1	1	1	1	1	0000H to 7FFFBH (512 Kbytes - 4 bytes)
Other than the above						Setting prohibited

Caution Be sure to clear bit 6 to 0.

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.

32.3.1.2 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 32 - 2 Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H

After reset: 0000H

R/W: R/W

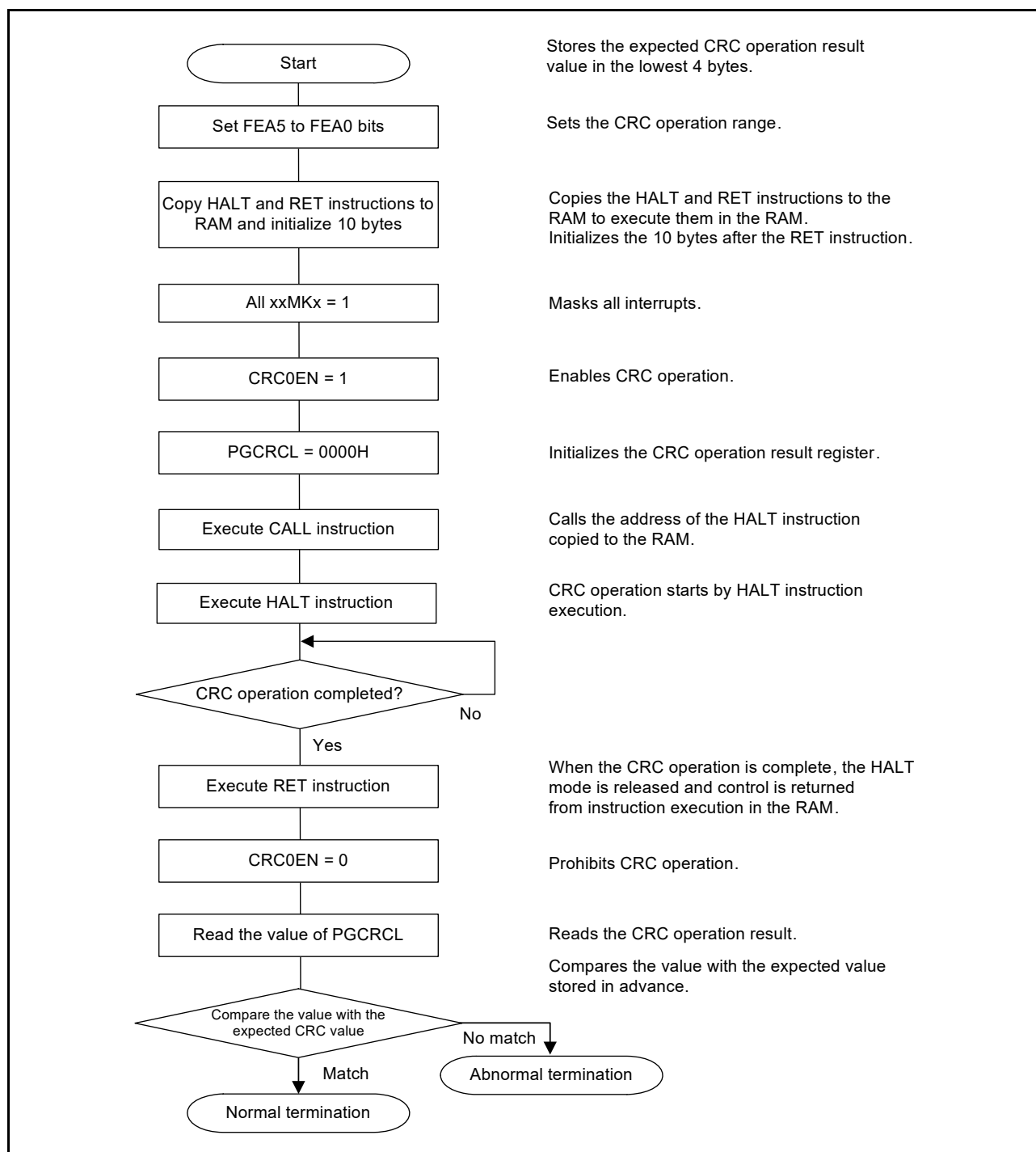
Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0
	High-speed CRC operation results							
	0000H to FFFFH							
	Holds the high-speed CRC operation results.							

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 32 - 3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 32 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.

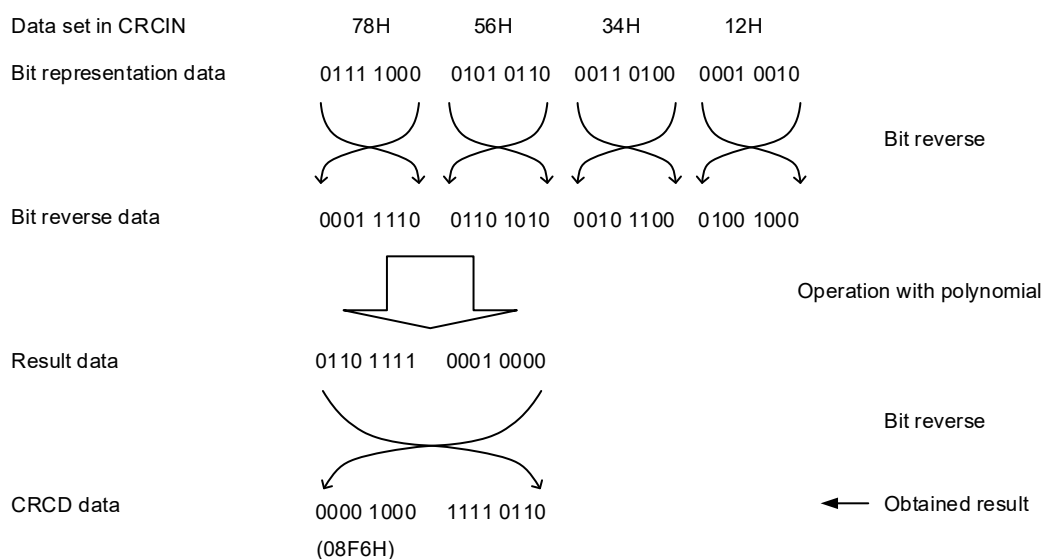
Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.

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.

32.3.2 CRC operation (general-purpose CRC)

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.

32.3.2.1 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 32 - 4 Format of CRC Input Register (CRCIN)

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

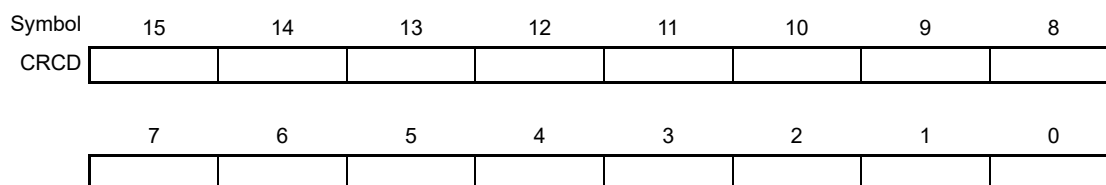
Symbol	7	6	5	4	3	2	1	0
CRCIN								
Bits 7 to 0	Function							
00H to FFH	Data input							

32.3.2.2 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 (fCLK) 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 32 - 5 Format of CRC Data Register (CRCD)

Address: F02FAH
After reset: 0000H
R/W: R/W

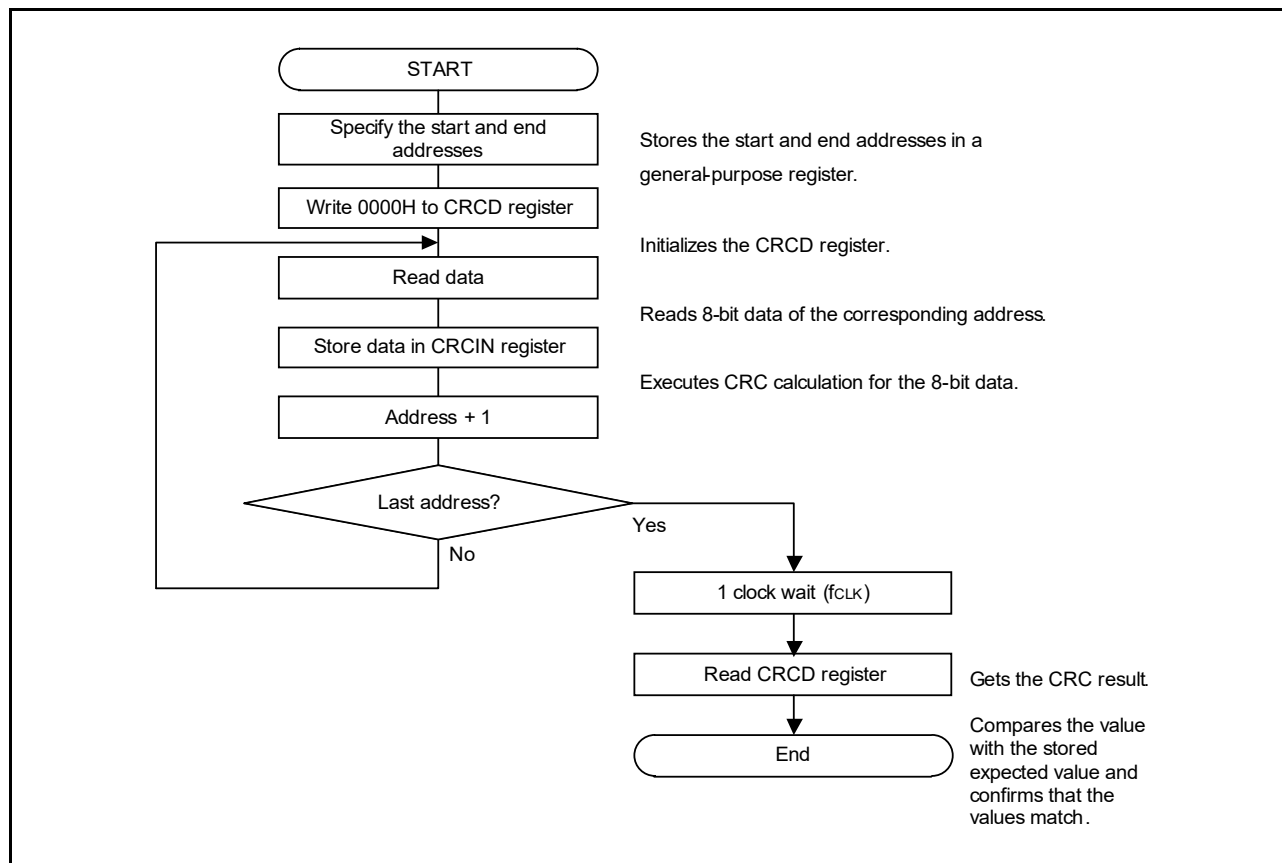


Caution 1. Read the value written to the CRCD register before writing to the CRCIN register.

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 32 - 6 CRC Operation Function (General-purpose CRC)



32.3.3 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/L23 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.

32.3.3.1 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 32 - 7 Format of Code Flash Memory Guard Register (GFLASH0)

Address: F0488H
 After reset: 0000H
 R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH0	RKEY[7:0]							
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	GFL0
	RKEY[7:0]							
	Function							
Bits 7 to 0	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. 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).							

32.3.3.2 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 32 - 8 Format of Data Flash Memory Guard Register (GFLASH1)

Address: F048AH

After reset: 0000H

R/W: R/W

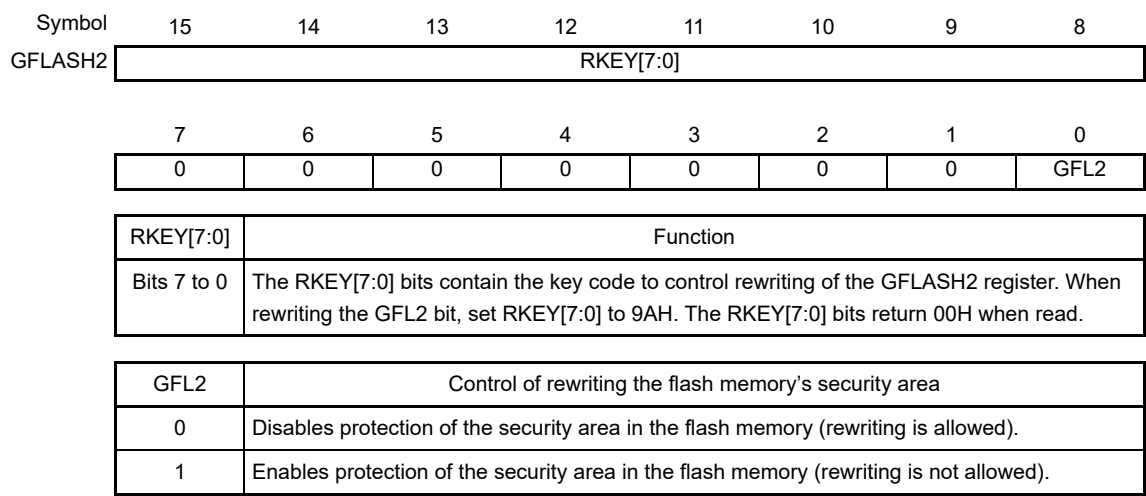
Symbol	15	14	13	12	11	10	9	8
GFLASH1	RKEY[7:0]							
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	GFL1
RKEY[7:0]	Function							
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of the GFLASH1 register. When rewriting the GFL1 bit, set RKEY[7:0] to C5H. The RKEY[7:0] bits return 00H when 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).							

32.3.3.3 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. The GFLASH2 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 32 - 9 Format of Flash Security Area Guard Register (GFLASH2)

Address: F048CH
After reset: 0000H
R/W: R/W



32.3.4 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/L23'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.

32.3.4.1 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 32 - 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.

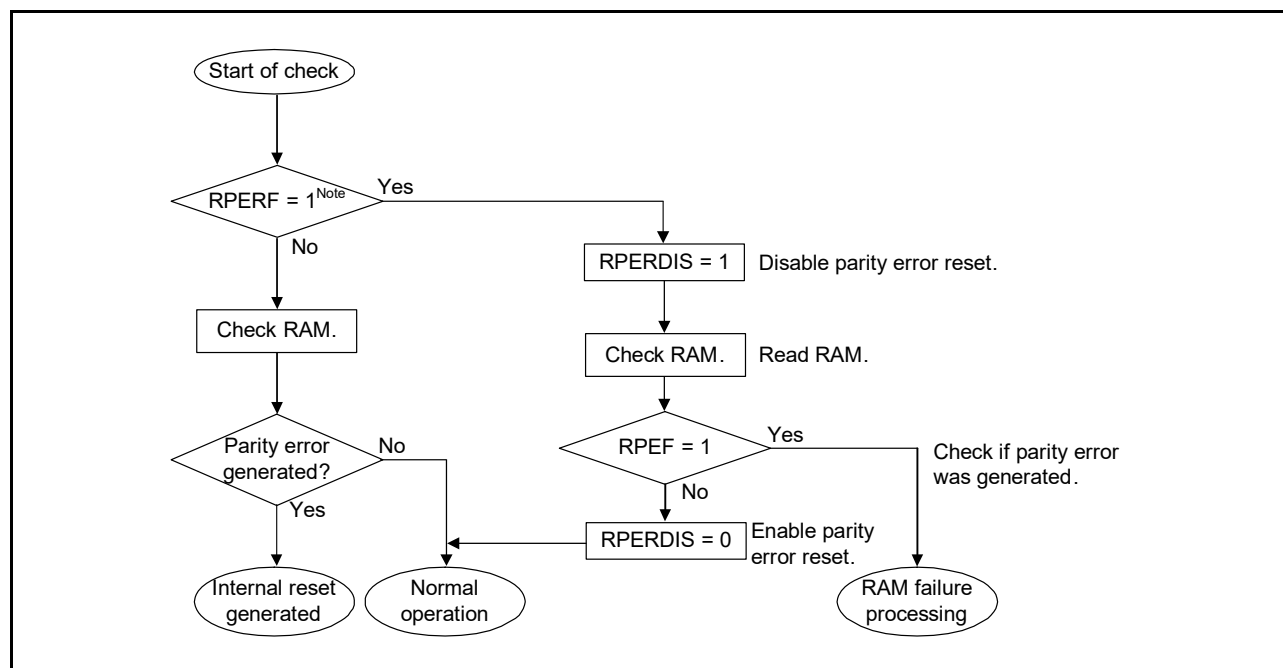
Remark 1. The parity error reset is enabled by default (RPERDIS = 0).

Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set to 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 to 1 when the RAM parity error occurs and cleared to 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 32 - 11 Flowchart of RAM Parity Check



Note To check the state following an internal reset triggered by a RAM parity error, see **Section 28 Reset Function**.

32.3.5 RAM guard function

The RL78/L23 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.

32.3.5.1 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 32 - 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 Note
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

Note The base address of the RAM differs depending on the size of the RAM in the product.

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).

32.3.6 SFR guard function

The RL78/L23 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.

32.3.6.1 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 32 - 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, PMCExx, PFOEx, PDIDISxx, CCDE, CCSm, PTDC, PFSEGx, ISCLCD Note						
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, PSMCR, MODRV, SOMRG						

Note The port registers (Pxx) are not protected.

32.3.7 Illicit memory access detection

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly. The RL78/L23 provides functionality to trigger a reset when an invalid memory area is accessed. Access to the areas indicated as “Not allowed” in **Figure 32 - 14** is detected as illicit.

Figure 32 - 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 FFFEFFH	General-purpose registers 32 bytes			Allowed
FFEE0H FFEDFH	RAM ^{Note}		Not allowed	
zzzzzH				Not allowed
Mirror				
Data flash memory				Allowed
F1000H F0FFFFH	Reserved		Allowed	
F0800H F07FFH	Special function registers (2nd SFRs) 2 Kbytes		Not allowed	Not allowed
F0000H EFFFFFH	Reserved			Allowed
EFFDEH EFFDDH	Product name in ASCII code (unique ID)			
EFFC0H EFFBFH	Reserved			Not allowed
EE000H EDFFFFH	Reserved			
yyyyyH xxxxxH	Code flash memory ^{Note}			Allowed
00000H				

(Note is listed on the next page.)

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)	RAM (zzzzzH to FFEFFH)	Lowest address of the area to be detected as illicit when accessed for reading or instruction fetching (yyyyyH)
R7F100LxE (x = F, G, J, L)	65536 × 8 bits (00000H to 0FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	10000H
R7F100LxG (x = F, G, J, L, M, P)	131072 × 8 bits (00000H to 1FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	20000H
R7F100LxJ (x = F, G, J, L, M, P)	262144 × 8 bits (00000H to 3FFFFH)	32768 × 8 bits (F7F00H to FFEFFH)	40000H
R7F100LxL (x = F, G, J, L, M, P)	524288 × 8 bits (00000H to 7FFFFH)	32768 × 8 bits (F7F00H to FFEFFH)	80000H

32.3.7.1 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 32 - 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.

32.3.8 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/L23 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.

32.3.8.1 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 32 - 16 Format of Guard Register of IAWCTL Register (GIAWCTL)

Address: F048EH
 After reset: 0000H
 R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GIAWCTL	RKEY[7:0]							
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	GIA
	RKEY[7:0]							
	Function							
Bits 7 to 0	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 A4H. 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).							

32.3.9 Frequency detection

The IEC60730 standard mandates checking that the oscillation frequency is correct. By using the CPU/peripheral hardware clock frequency (fCLK) 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>

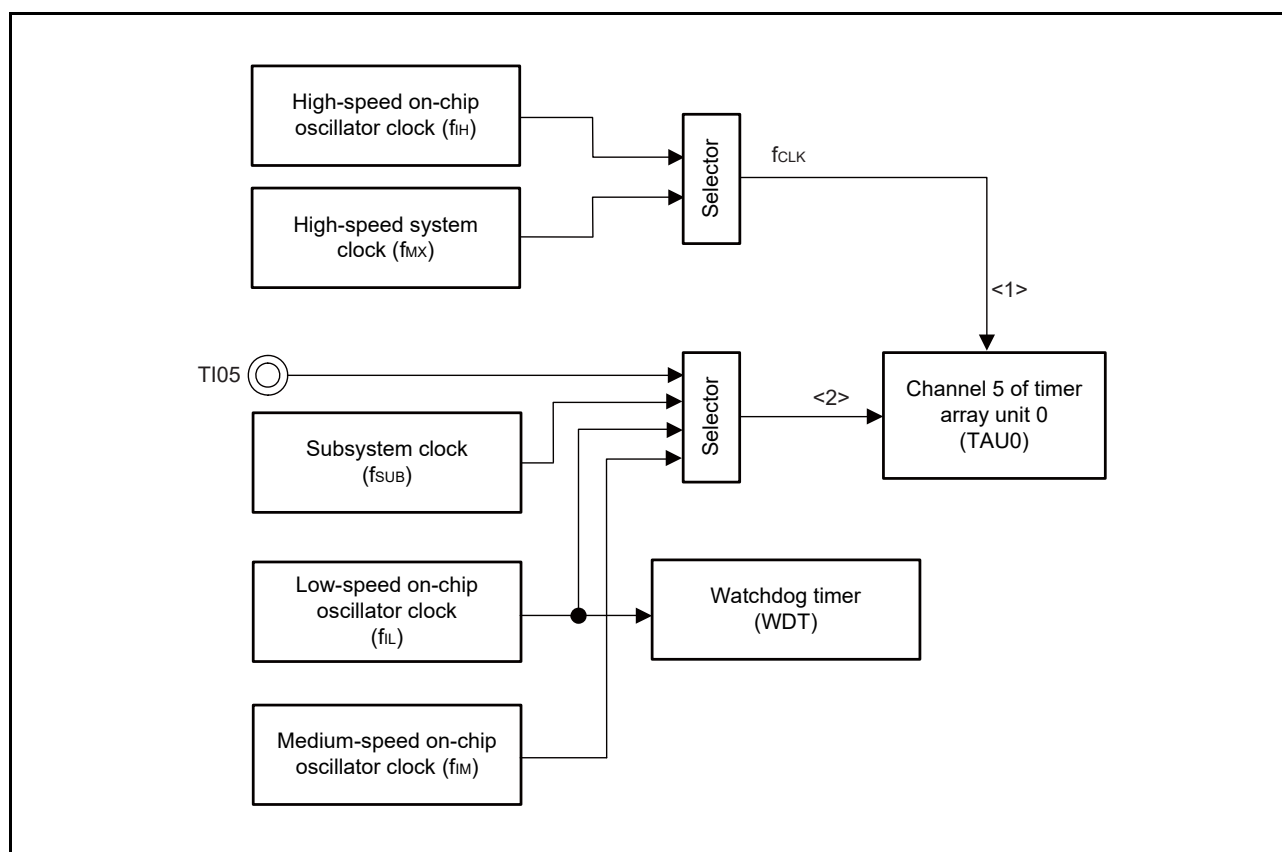
<1> CPU/peripheral hardware clock frequency (fCLK):

- High-speed on-chip oscillator clock (f_H)
- 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_L)
- Subsystem clock (f_{SUB})
- Medium-speed on-chip oscillator clock (f_M)

Figure 32 - 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**.

32.3.9.1 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 32 - 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 ELCL
0	1	1	Medium-speed on-chip oscillator clock (f _{IM})
1	0	0	Low-speed on-chip oscillator clock (f _{IL})
1	0	1	Subsystem clock (f _{SUB})
Other than the above			Setting prohibited

32.3.10 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), TSCAP voltage of the CTSU, 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.

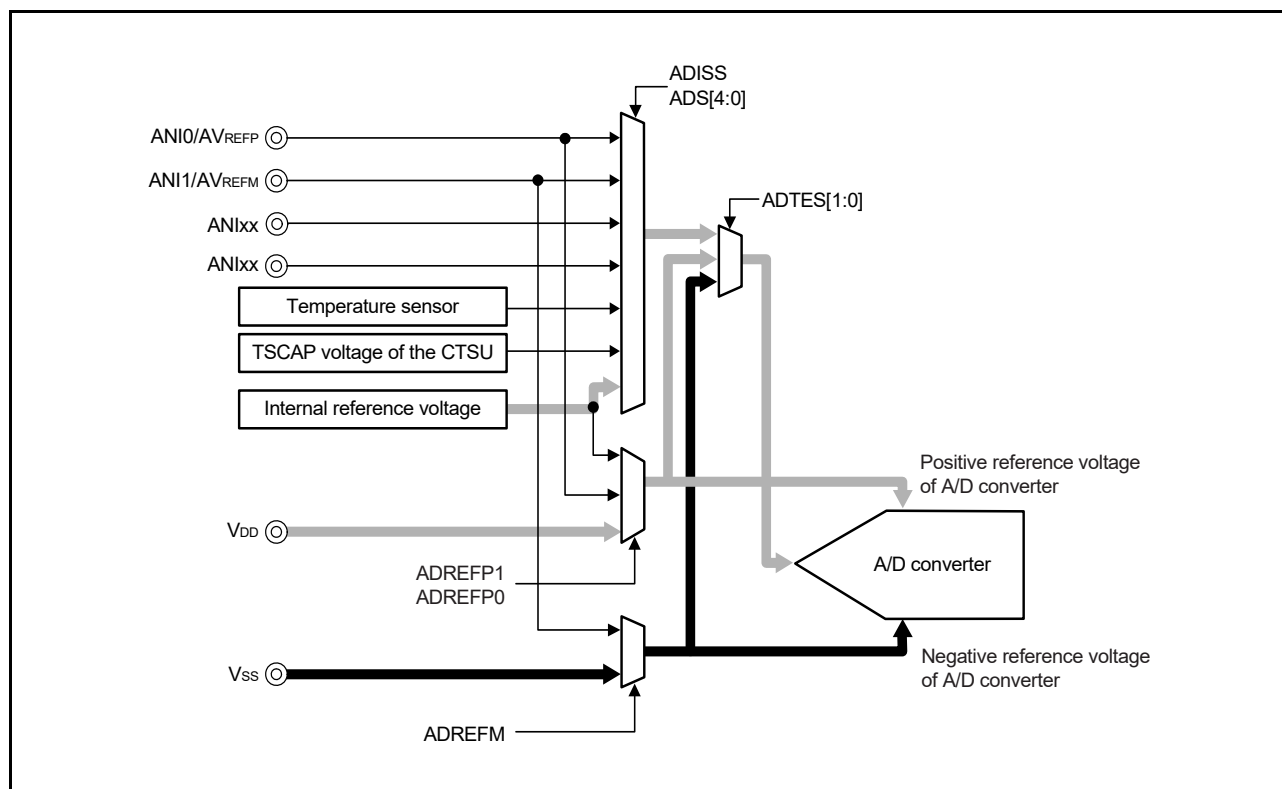
- <1> Select the ANLxx pin for A/D conversion using the ADTES register (ADTES[1:0] = 00B).
- <2> Perform A/D conversion for the ANLxx pin (conversion result 1-1).
- <3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES[1:0] = 10B)
- <4> Perform A/D 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 (ADTES[1:0] = 00B).
- <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 (ADTES[1:0] = 11B)
- <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 (ADTES[1:0] = 00B).
- <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 32 - 19 Configuration of Testing of the A/D Converter



32.3.10.1 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), TSCAP voltage of the CTSU, 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 32 - 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.

32.3.10.2 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 32 - 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/AVREFF pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	1	0	0	0	0	ANI16	P22/ANI16 pin
0	1	0	0	0	1	ANI17	P23/ANI17 pin
0	1	0	0	1	0	ANI18	P24/ANI18 pin
0	1	0	0	1	1	ANI19	P25/ANI19 pin
0	1	0	1	0	0	ANI20	P26/ANI20 pin
0	1	0	1	0	1	ANI21	P27/ANI21 pin
0	1	0	1	1	0	ANI22	P10/ANI22 pin
0	1	0	1	1	1	ANI23	P11/ANI23 pin
0	1	1	0	0	0	ANI24	P12/ANI24 pin
0	1	1	0	0	1	ANI25	P13/ANI25 pin
0	1	1	0	1	0	ANI26	P14/ANI26 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	

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 1, 2, and 4 (PM1, PM2, and PM4).

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 1, 2, and 4 (PMCA1, PMCA2, and PMCA4).

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.

(Cautions are listed on the next page.)

- 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 16.7.5 Example of using the ADC when selecting the temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU, and software trigger no-wait 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 42.3.2 Supply current characteristics is added.

32.3.11 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/L23 provides functionality to read the output level of an I/O pin when the pin is in the output mode.

32.3.11.1 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 32 - 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. 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 2. The setting of this register has no effect on the input port pins (P123, P124, and P137) and output port pin (P130).

Caution 3. Be sure to clear bits 7 to 1 to 0.

Remark m = 0 to 9, 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 5, 7, 9, 12
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 to 9, 12 to 14
n = 0 to 7

32.3.12 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 RxD 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 TxD pin is controlled by the port functions so that it does not affect its communication partner.

- Communication with negative logic

The port pin multiplexed with the TxD pin is set to the input mode ($PM_{xx} = 1$) and an on-chip pull-up resistor is connected ($PU_{xx} = 1$) to retain the setting of 1. Retaining of 1 on the TxD_{A0} pin is possible by setting the PFOE14 bit in the PFOE1 register to 0 while 1 is being output from the TxD_{A0} pin ($PM_{xx} = 0$, $P_{xx} = 1$). Retaining of 1 on the TxD₀ pin is possible by setting the PFOE10 bit in the PFOE1 register to 0 while 1 is being output from the TxD₀ pin ($PM_{xx} = 0$, $P_{xx} = 1$).

- Communication with positive logic

0 ($PM_{xx} = 0$, $P_{xx} = 0$) is output from the port that also uses the TxD pin.

32.3.12.1 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 32 - 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	ULBS7	ULBS6	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0
	ULBS7	UART loopback selection						
	0	Inputs the state of the RxDA3 pin of serial interface UARTA3 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						
	ULBS6	UART loopback selection						
	0	Inputs the state of the RxDA2 pin of serial interface UARTA2 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						
	ULBS5	UART loopback selection						
	0	Inputs the state of the RxDA1 pin of serial interface UARTA1 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						
	ULBS4	UART loopback selection						
	0	Inputs the state of the RxDA0 pin of serial interface UARTA0 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						
	ULBS3	UART loopback selection						
	0	Inputs the state of the RxD3 pin of serial array unit UART3 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 RxD2 pin of serial array unit UART2 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 RxD1 pin of serial array unit UART1 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						
	ULBS0	UART loopback selection						
	0	Inputs the state of the RxD0 pin of serial array unit UART0 to the reception shift register.						
	1	Loops back the output of the transmission shift register to the reception shift register.						

(Remark is listed on the next page.)

- Remark** The RxDA2 pin is only available in the following products.
- 48- to 64-pin products with 256 or 512 Kbytes of flash memory
 - 80- and 100-pin products
- The RxDA3 pin is only available in the 80- to 100-pin products.

Section 33 Security Functions

33.1 True Random Number Generator

33.1.1 Function of the true random number generator

The true random number generator generates 32-bit random number seeds (which are true random numbers).

33.1.2 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)

33.1.2.1 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 33 - 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.

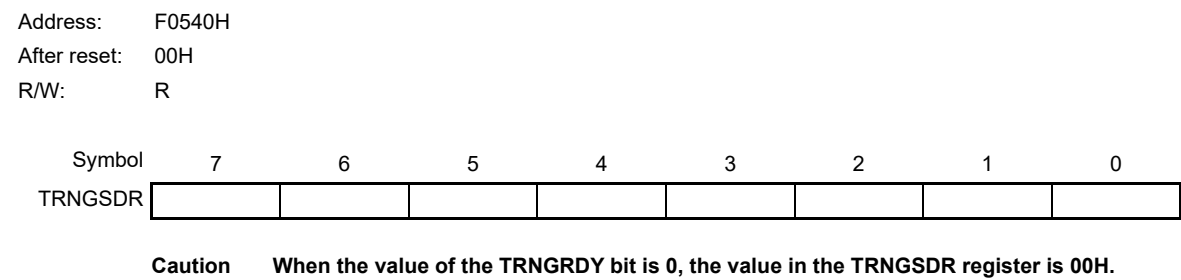
TRNGEN	Control over operation of the true random number generator
0	Stops the true random number generator.
1	Enables the true random number generator.

TRNGST	Trigger to start generating a random number seed
0	The trigger is inactive.
1	Starts generation of a random number seed.

33.1.2.2 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.

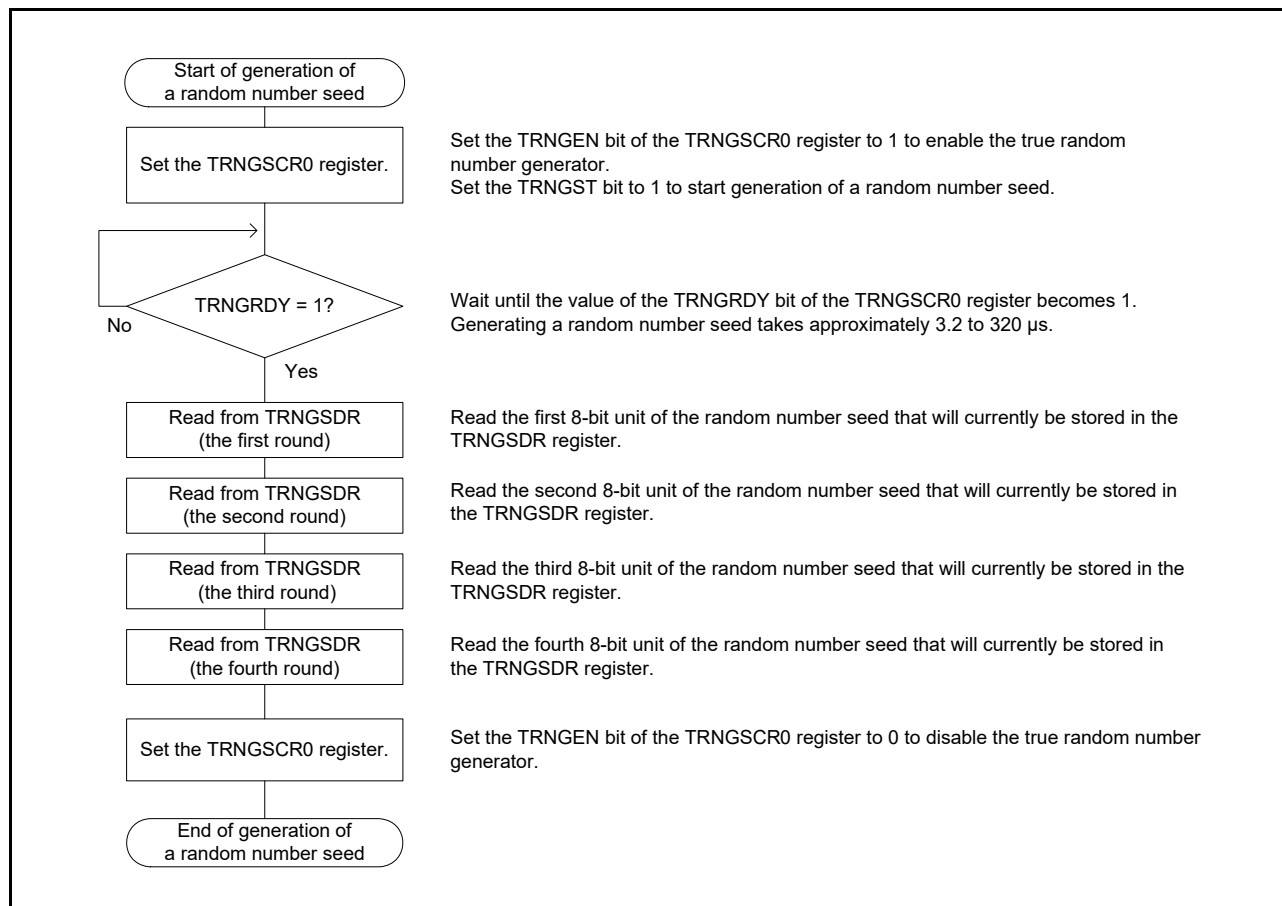
Figure 33 - 2 Format of Random Number Seed Data Register (TRNGSDR)



33.1.3 Operations of the true random number generator

Figure 33 - 3 shows the procedure for using the true random number generator to generate a random number seed.

Figure 33 - 3 Procedure for Using the True Random Number Generator to Generate a Random Number Seed



33.2 Flash Read Protection

33.2.1 Function of flash read protection

The flash read protection can be used to protect a specified range of the code flash memory area against read access by the CPU, DTC, or SMS. Fetching of instructions in the specified range by the CPU is still possible.

Note that verifying the code flash memory area protected against read access by the flash read protection is possible in the serial programming mode.

33.2.2 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. Read access to the whole range of the code flash memory is enabled by the default setting at the time of shipment. Set a range in the code flash memory to be protected against read access by specifying the flash read protection start block and end block. This makes read access to addresses in the specified range of blocks in the code flash memory impossible. In addition, changing of the flash read protection settings can be disabled, which makes the settings for the start and end blocks fixed, thus making 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. Note that release from disabling of changes to the flash read protection settings requires deletion of the contents of the full range of the code flash memory. Accordingly, reading the data written in the range protected against read access is impossible after release from disabling of changes to the flash read protection settings. **Table 33 - 1** describes the settings for flash read protection and their functions. **Table 33 - 2** describes the method of setting flash read protection.

Table 33 - 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 1FFH.
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 1FFH.
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, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Table 33 - 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.

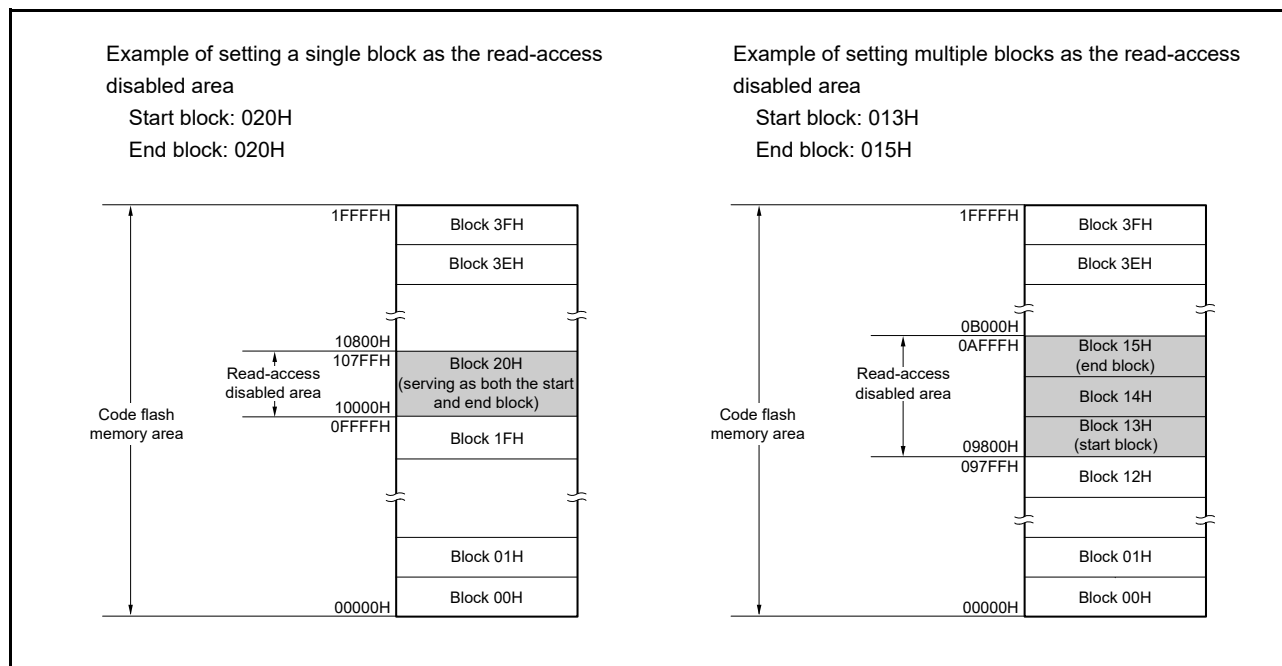
Note Release from the fixed setting is only possible when erasure of blocks is not prohibited, rewriting of boot area is not prohibited, and the code and data flash memory areas are blank.

- 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.
- Caution 5.** When settings for flash read protection have been made through self-programming, the settings become enabled after the MCU is reset and then released from the reset state.
- Caution 6.** To use the flash read protection, set some specific bits in the extra area as follows.
 BTBLS bits: A value within the range from 00H to 06H
 BAPR bit: 0

33.2.3 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 33 - 4** shows examples of setting read-access disabled areas for the application of flash read protection.

Figure 33 - 4 Examples of Setting Read-Access Disabled Areas for the Application of Flash Read Protection



To enable read access to the full range of the code flash memory again after having set a range to be protected against read access, set the flash read protection start block and end block as 1FFH while prohibition of changing the flash read protection settings is disabled.

33.3 Unique ID

33.3.1 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.**

33.3.2 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 33 - 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 33 - 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		30H: "0"
EFFDBH		4CH: "L"
EFFDCH		46H: "F" (44 pins) 47H: "G" (48 pins) 4AH: "J" (52 pins) 4CH: "L" (64 pins) 4DH: "M" (80 pins) 50H: "P" (100 pins)
EFFDDH		45H: "E" (64-Kbyte flash memory) 47H: "G" (128-Kbyte flash memory) 4AH: "J" (256-Kbyte flash memory) 4CH: "L" (512-Kbyte flash memory)

Section 34 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.

34.1 Functions of the SNOOZE Mode Sequencer

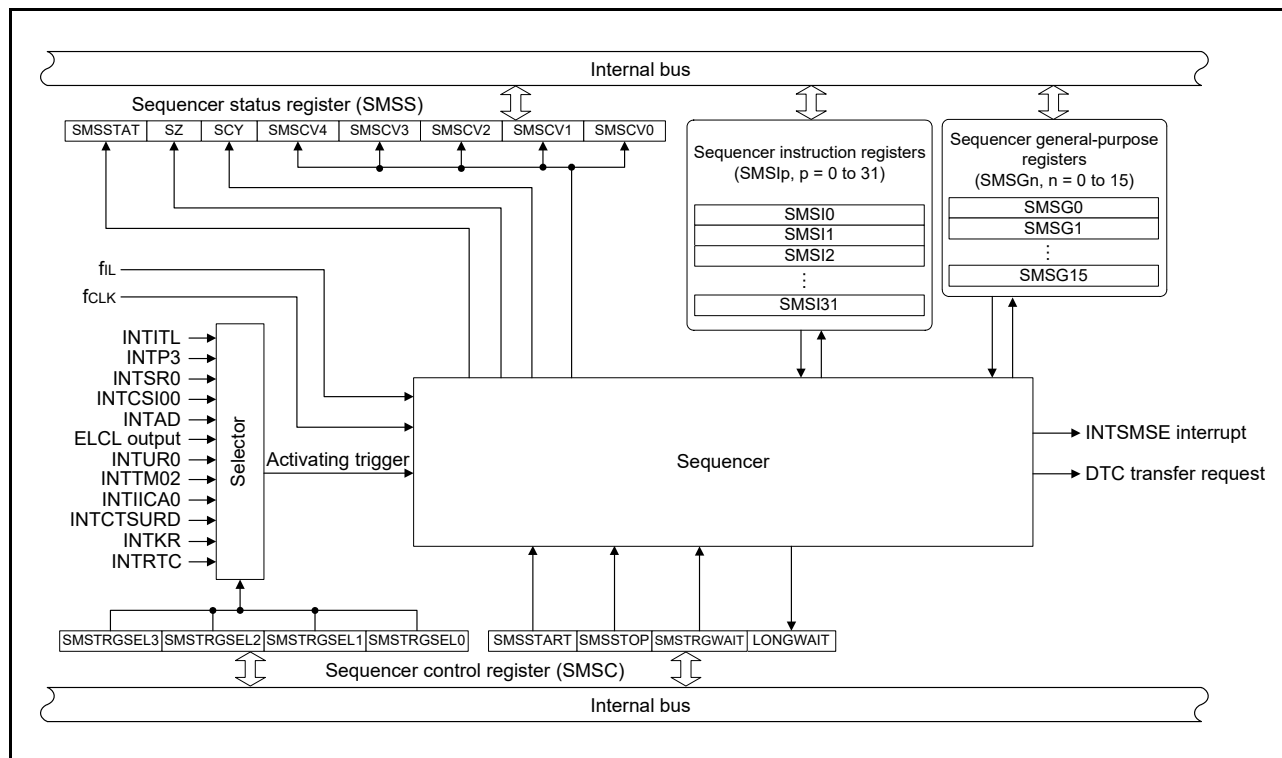
The SNOOZE mode sequencer can handle processing independently of the CPU and has the following functions.

- 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

34.2 Configuration of the SNOOZE Mode Sequencer

Figure 34 - 1 is a block diagram of the SNOOZE mode sequencer.

Figure 34 - 1 Block Diagram of the SNOOZE Mode Sequencer



34.3 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 (SMSGn) (n = 0 to 15)
- Sequencer control register (SMSC)
- Sequencer status register (SMSS)

34.3.1 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 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. For a precautionary note on the dependence on the number of pins of RL78/L23 products of the presence or absence of the PER bit, see **Section 6 Clock Generator**.

Figure 34 - 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	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTSUN
	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. Be sure to set bit 1 to 0.

34.3.2 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) 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 34 - 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	DACRES	SMSRES	CMPRES	TML32RES	0	0	0	CTSURES
	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.						

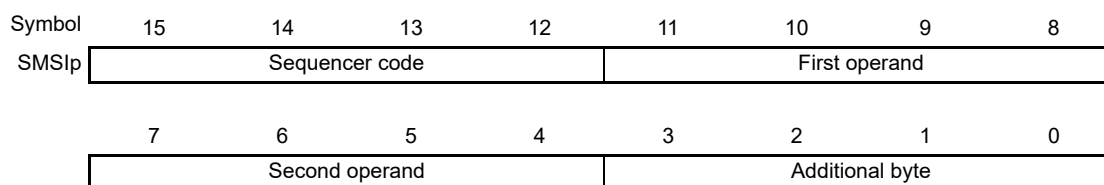
Caution Be sure to set bits 3 and 2 to 0.

34.3.3 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 **34.5 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 34 - 1** shows the correspondences between the memory addresses of the SMSIp registers and values of the SMSCV0 to SMSCV4 bits. **Table 34 - 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 34 - 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 34 - 1 Correspondences between the Memory Addresses of the SMSIp Registers and Values of the SMSCV0 to SMSCV4 Bits

SMSIp	Address	SMSCV[4:0]	SMSIp	Address	SMSCV[4:0]
SMSI15	F039EH, F039FH	01111B	SMSI31	F03BEH, F03BFH	11111B
SMSI14	F039CH, F039DH	01110B	SMSI30	F03BCH, F03BDH	11110B
SMSI13	F039AH, F039BH	01101B	SMSI29	F03BAH, F03BBH	11101B
SMSI12	F0398H, F0399H	01100B	SMSI28	F03B8H, F03B9H	11100B
SMSI11	F0396H, F0397H	01011B	SMSI27	F03B6H, F03B7H	11011B
SMSI10	F0394H, F0395H	01010B	SMSI26	F03B4H, F03B5H	11010B
SMSI9	F0392H, F0393H	01001B	SMSI25	F03B2H, F03B3H	11001B
SMSI8	F0390H, F0391H	01000B	SMSI24	F03B0H, F03B1H	11000B
SMSI7	F038EH, F038FH	00111B	SMSI23	F03AEH, F03AFH	10111B
SMSI6	F038CH, F038DH	00110B	SMSI22	F03ACH, F03ADH	10110B
SMSI5	F038AH, F038BH	00101B	SMSI21	F03AAH, F03ABH	10101B
SMSI4	F0388H, F0389H	00100B	SMSI20	F03A8H, F03A9H	10100B
SMSI3	F0386H, F0387H	00011B	SMSI19	F03A6H, F03A7H	10011B
SMSI2	F0384H, F0385H	00010B	SMSI18	F03A4H, F03A5H	10010B
SMSI1	F0382H, F0383H	00001B	SMSI17	F03A2H, F03A3H	10001B
SMSI0	F0380H, F0381H	00000B	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 in the sequence of registers. Accordingly, in the SMSI31 register, write the code for the termination command or interrupt plus termination command to stop the sequencer or the code for a branch command to cause a branch to other processing.

Table 34 - 2 Types of Processing Specified by the SMSIp Registers

Name of Processing	Operation ^{Note 1}	Sequencer Code	First Operand (4 Bits)	Second Operand (4 Bits)	Additional Byte (4 Bits)
8-bit data transfer 1	[MSGn + Byte] ← MSGm	0000	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	Byte ^{Note 3}
8-bit data transfer 2	MSGm ← [MSGn + Byte]	0001	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	Byte ^{Note 3}
16-bit data transfer 1	[MSGn + Byte] ← MSGm	0010	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	Byte ^{Note 3}
16-bit data transfer2	MSGm ← [MSGn + Byte]	0011	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	Byte ^{Note 3}
1-bit data setting	[MSGn + Byte].bit ← 1	0100	n of MSGn ^{Note 2}	bit ^{Note 3}	Byte ^{Note 3}
1-bit data clearing	[MSGn + Byte].bit ← 0	0101	n of MSGn ^{Note 2}	bit ^{Note 3}	Byte ^{Note 3}
1-bit data transfer	SCY ← [MSGn + Byte].bit	0110	n of MSGn ^{Note 2}	bit ^{Note 3}	Byte ^{Note 3}
Word addition	MSGn, SCY ← MSGn + MSGm	0111	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	0000
Word subtraction	MSGn, SCY ← MSGn - MSGm	0111	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	0001
Word comparison	MSGn - MSGm	0111	n of MSGn ^{Note 2}	m of MSGm ^{Note 2}	0010
Logical shift right	SCY ← MSGn.0, MSGm.15 ← 0, MSGn.m-1 ← MSGn.m	0111	n of MSGn ^{Note 2}	0000	0011
Branch 1 (SCY = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 1	1000	\$addr5 ^{Note 4}		0000
Branch 2 (SCY = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 0	1000	\$addr5 ^{Note 4}		0001
Branch 3 (SZ = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 1	1000	\$addr5 ^{Note 4}		0010
Branch 4 (SZ = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 0	1000	\$addr5 ^{Note 4}		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	n of MSGn ^{Note 2}	Bit ^{Note 3}	Byte ^{Note 3}
Conditional wait 2 (bit = 0)	SMSS[4:0] ← SMSS[4:0] if [MSGn + Byte].bit = 0	1011	n of MSGn ^{Note 2}	Bit ^{Note 3}	Byte ^{Note 3}
Termination	SMSSTAT ← 0, SMSS[4:0] ← 0, Stopping the sequencer	1111	0000	0000	0000
Interrupt plus termination	SMSS[4:0] ← 0, Issue an interrupt and then stop the sequencer.	1111	0000	0000	0001
DTC activation	Output of a DTC activating source signal	1111	0000	0000	0010

Note 1. For details on the terms, see **34.5 Commands for Use in Processing by the Sequencer**.

Note 2. Specify values in the range from 0 to 15 (from 0000B to 1111B) for n and m.

Note 3. Specify values in the range from 0 to 7 (from 0000B to 0111B) for the bytes.

Note 4. This is an 8-bit displacement value. Specify a relative address in the ranges from -31 to -1 and 1 to 31 (0000 0001B to 0001 1111B, 1111 1111B to 1110 0001B).

34.3.4 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 34 - 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

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

Remark n = 1 to 14

Table 34 - 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

34.3.5 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 34 - 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	SMSTRG WAIT	LONGWAIT	SMSTRGSE L3	SMSTRGSE L2	SMSTRGSE L1	SMSTRGSE L0

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.

SMSTRG WAIT Note 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.

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.

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 34 - 6 Format of Sequencer Control Register (SMSC) (2/2)

SMSTRGSEL3	SMSTRGSEL2	SMSTRGSEL1	SMSTRGSEL0	Selected Activating Trigger for the SNOOZE Mode Sequencer ^{Note}
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	ELCL output signal
0	1	1	0	UARTA0 reception transfer end interrupt (INTUR0)
0	1	1	1	Timer channel 02 count or capture end interrupt (INTTM02)
1	0	0	0	IICA0 communication end interrupt (INTIICA0)
1	0	0	1	Fixed-cycle interrupt from the RTC (INTRTC)
1	0	1	0	Key return signal detection interrupt (INTKR)
1	0	1	1	CTSU measured data transfer request interrupt (INTCTSURD)
Other than above				Setting prohibited

Note 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.

34.3.6 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 34 - 7 Format of Sequencer Status Register (SMSS)

Address: F03E1H

After reset: 00H

R/W: R

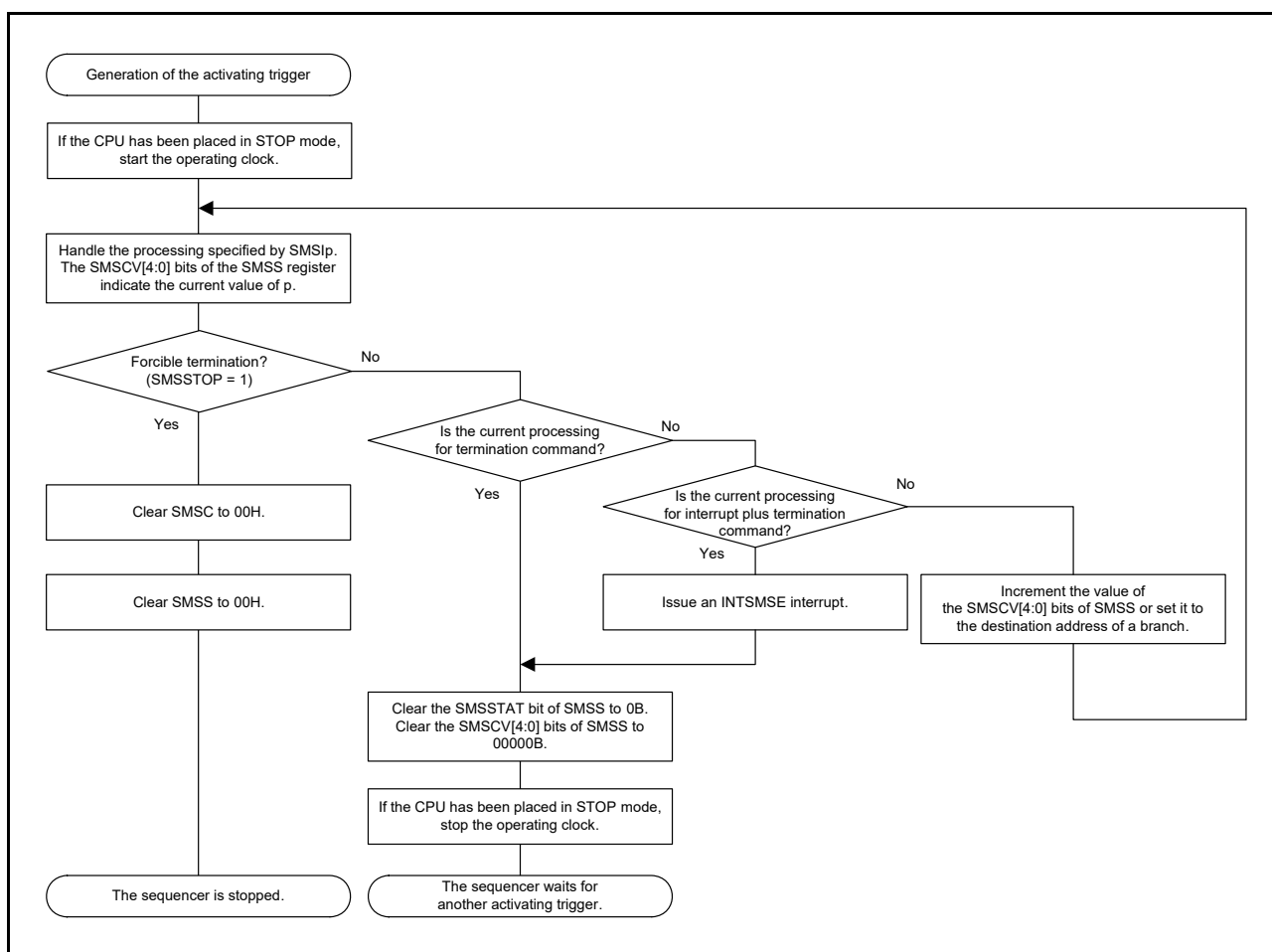
Symbol	<7>	6	5	4	3	2	1	0
SMSS	SMSSTAT	SZ	SCY	SMSCV4	SMSCV3	SMSCV2	SMSCV1	SMSCV0
	SMSSTAT	State of operation of the SNOOZE mode sequencer						
	0	The SNOOZE mode sequencer is stopped.						
	1	The SNOOZE mode sequencer is operating.						
	SZ	Zero flag of the SNOOZE mode sequencer						
	0	The result 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.		

34.4 Operations of the SNOOZE Mode Sequencer

34.4.1 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 SMSI0 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, 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 34 - 8** shows the flow of internal operations of the SNOOZE mode sequencer.

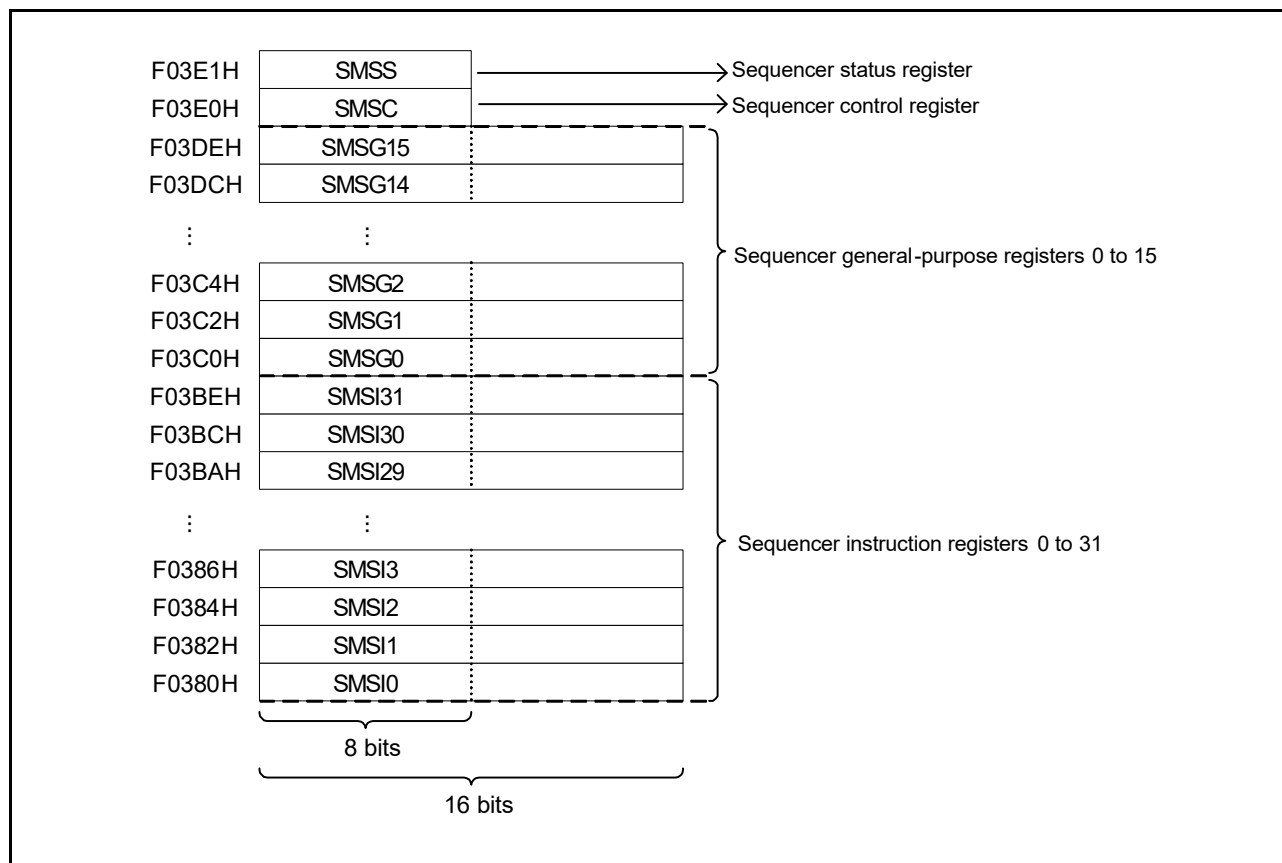
Figure 34 - 8 Flow of Internal Operations of the SNOOZE Mode Sequencer



34.4.2 Memory space allocated to the sequencer

The registers of the SNOOZE mode sequencer are in the second SFR space as shown in **Figure 34 - 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 34 - 9 Memory Space Allocated to the SNOOZE Mode Sequencer

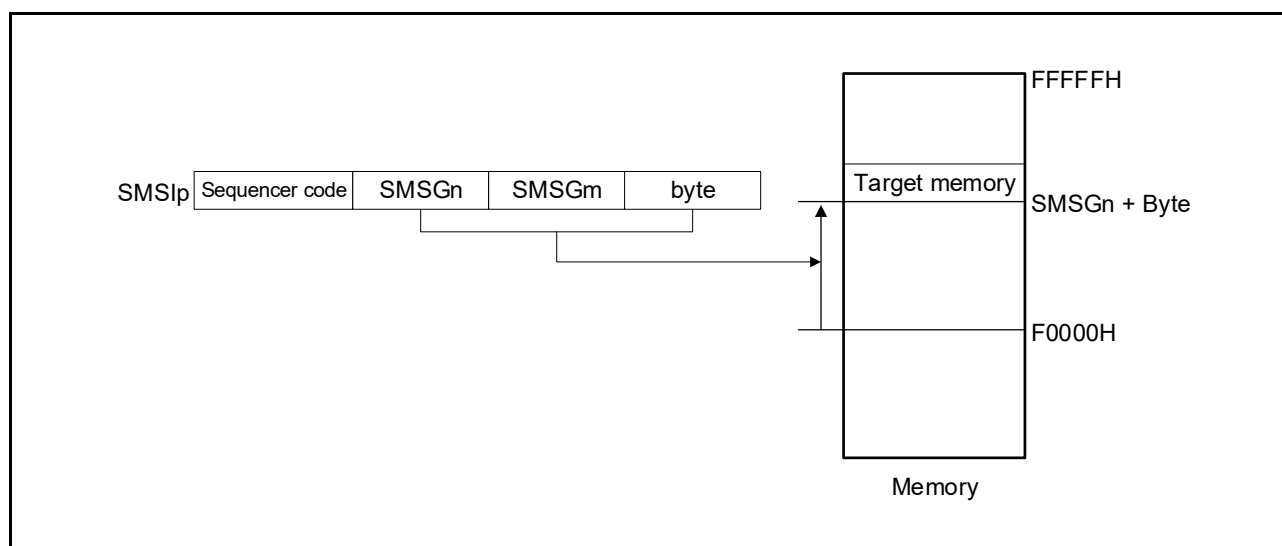


Moreover, the sequencer also has indirect access with some limitations^{Note} to the memory space from F0000H to FFFFFH through use of the SMSGn registers.

Note Access by the SNOOZE mode sequencer to the areas listed below is prohibited.

- General-purpose register area
- Mirror area
- Data flash memory area
- An internal RAM area that has been placed in shutdown mode by the setting of the memory power reduction control register (PSMCR)
- Sequencer instruction registers 0 to 31 (SMSI0 to SMSI31)
- Sequencer control register (SMSC)
- Sequencer status register (SMSS)
- SMSEN bit of peripheral enable register 1 (PER1)
- SMSRES bit of peripheral reset control register 1 (PRR1)

Figure 34 - 10 Memory Space to which the Sequencer Has Access



34.4.3 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 **34.5 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 **34.5 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.

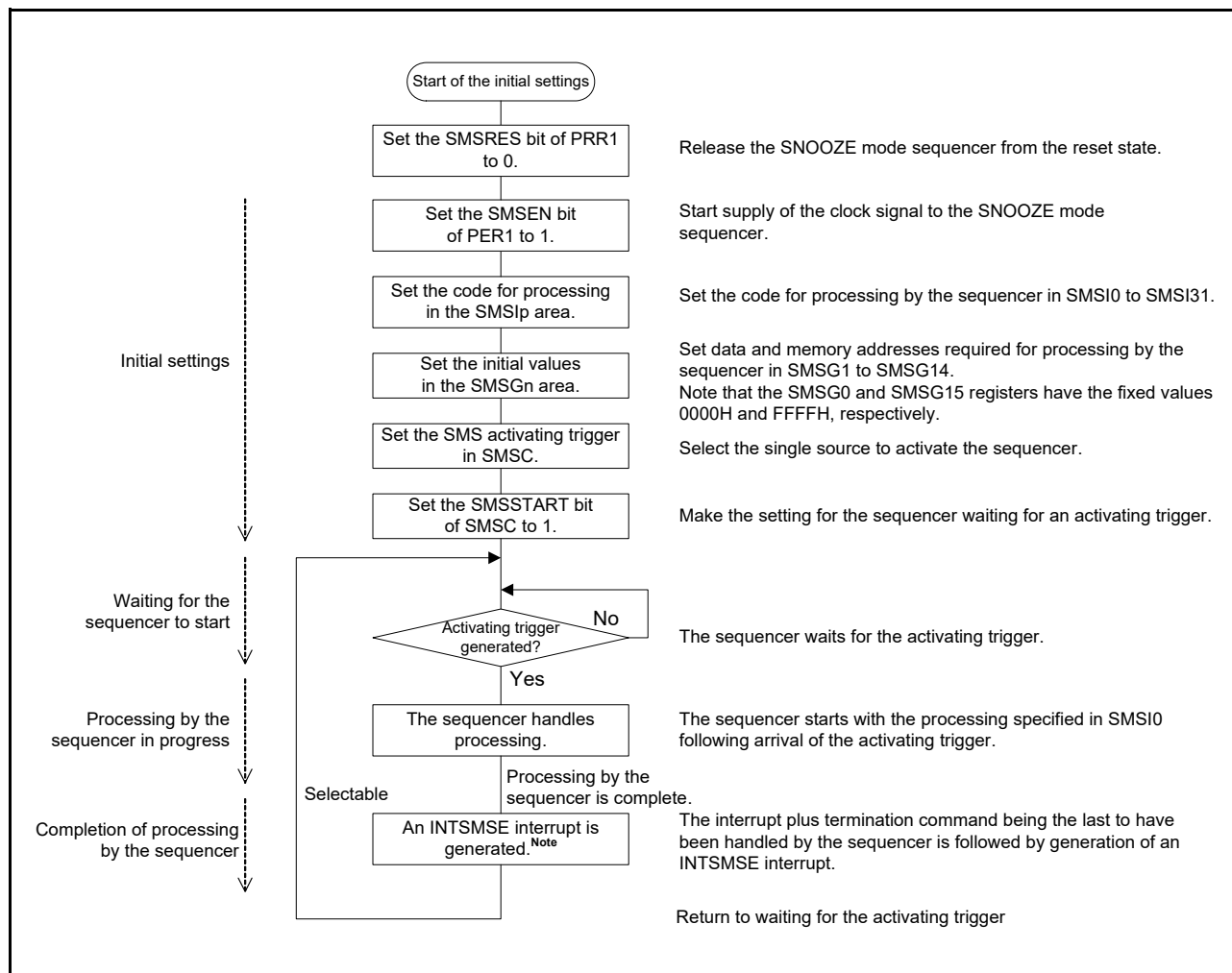
See **34.3.6 Sequencer status register (SMSS)**.

34.4.4 Procedures for running the SNOOZE mode sequencer

1. Example of the initial settings, activation, and operation of the SNOOZE mode sequencer

Figure 34 - 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 34 - 11 Flow of Activating and Running the SNOOZE Mode Sequencer

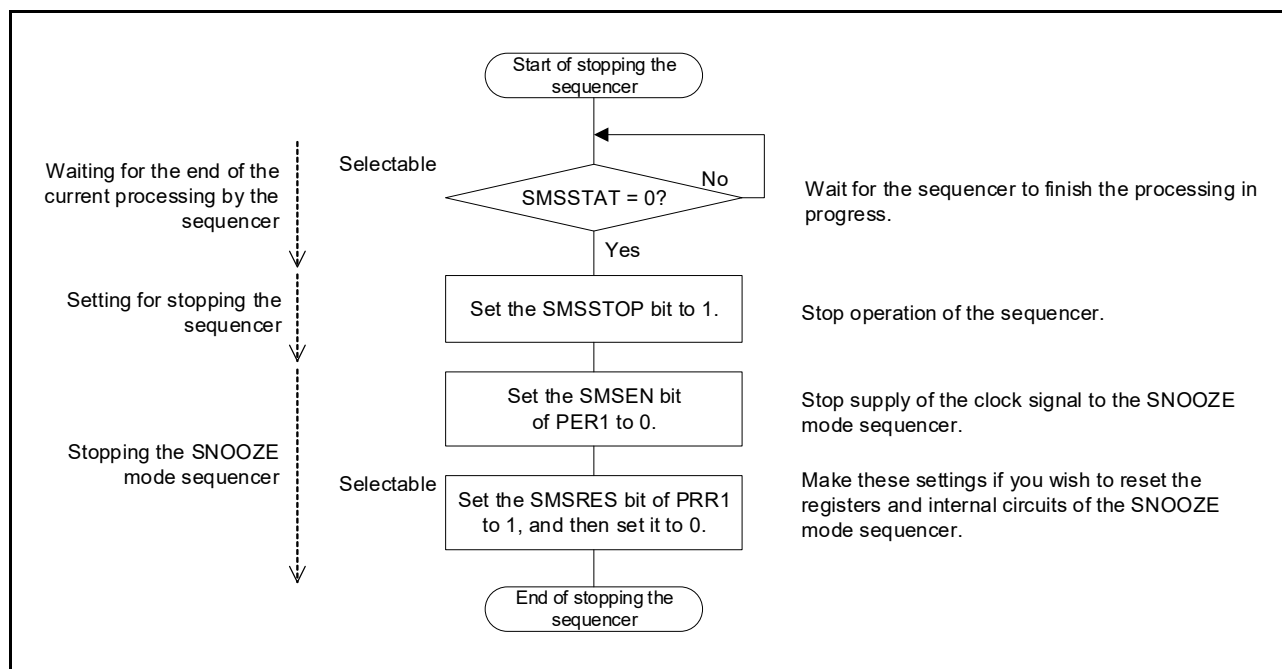


Note If processing by the sequencer ends following the processing for termination command or setting of the trigger bit for forcible termination (the SMSSTOP bit in the SMSC register), an INTSMSE interrupt will not be generated. If processing by the sequencer ends for the latter reason (setting of the 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 34 - 12 shows an example of a flow of stopping the SNOOZE mode sequencer while it is operating.

Figure 34 - 12 Flow of Stopping the SNOOZE Mode Sequencer



34.4.5 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 command 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 34 - 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.

34.5 Commands for Use in Processing by the Sequencer

The sequencer can handle 21 types of processing. For details, see sections **34.5.1** to **34.5.21**. Terms used in the descriptions of the commands (**34.5.1** to **34.5.21**) 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 -1, 1 to 31)
- jdisp8: 8-bit signed displacement value (0000 0001B 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

34.5.1 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).

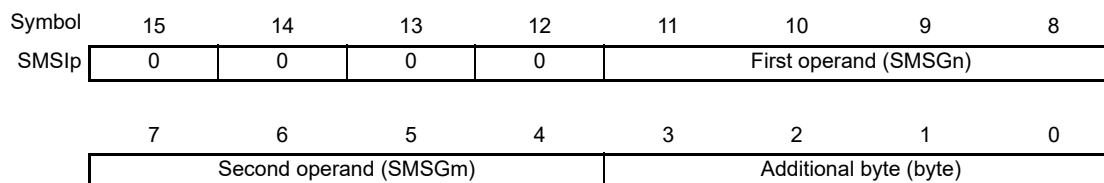
Sequencer code: 0000B

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.

34.5.2 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).

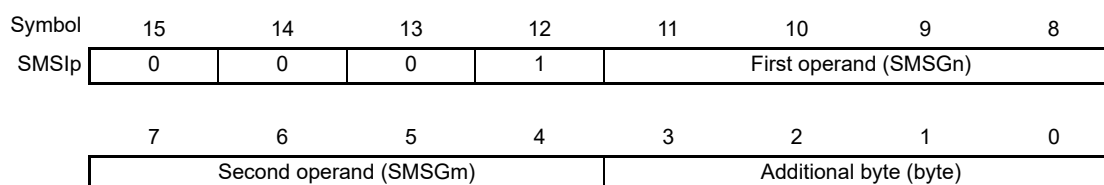
Sequencer code: 0001B

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 8 lower-order bits of the SMSG7 register.

34.5.3 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.

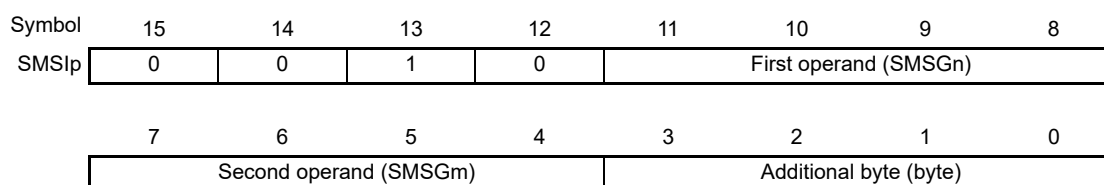
Sequencer code: 0010B

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



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.

34.5.4 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.

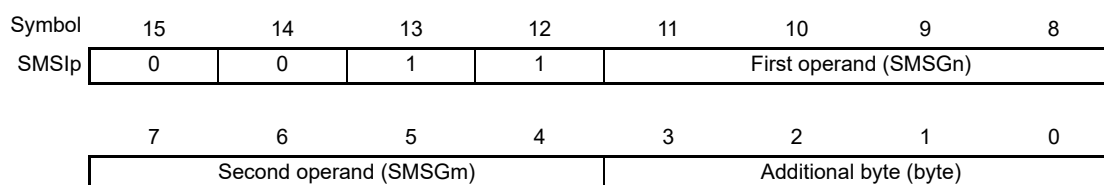
Sequencer code: 0011B

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: $\text{SMSGm} \leftarrow [\text{SMSGn} + \text{Byte}]$



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.

34.5.5 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.

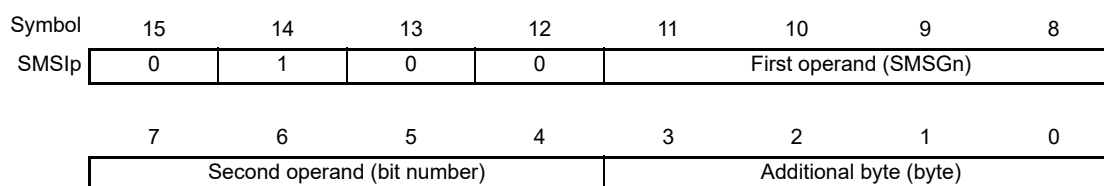
Sequencer code: 0100B

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 $\leftarrow$ 1



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.

34.5.6 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.

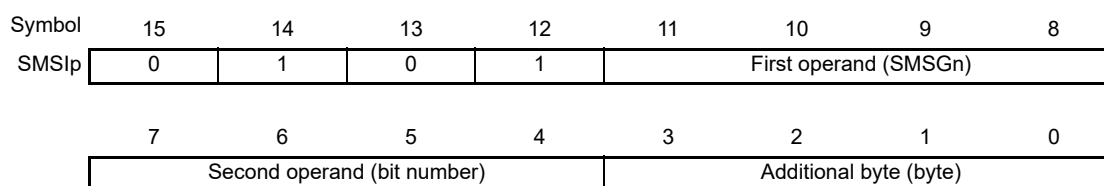
Sequencer code: 0101B

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 [MSG1+2H].3. In the case where MSG1 = 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.

34.5.7 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.

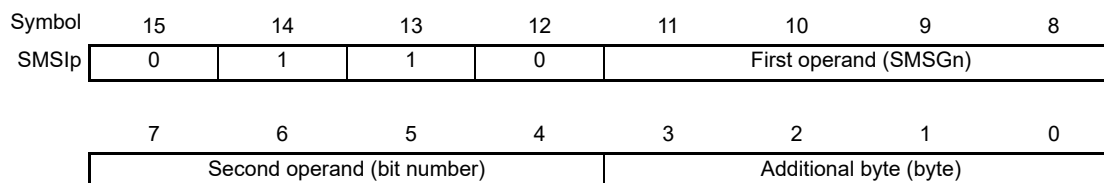
Sequencer code: 0110B

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, [MSGn+Byte].bit

Equivalent CPU operation: $SCY \leftarrow [MSGn+Byte].bit$



Example of a statement: 0110 0001 0011 0010 B

The equivalent CPU command in this case is MOV1 SCY, [MSG1+2H].3. In the case where MSG1 = 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.

34.5.8 Word addition

Word addition is for adding two 16-bit values. The values of the SMSG_n and SMSG_m registers that are specified as the first and second operands are added, and the result is stored in the SMSG_n register specified as the first operand. If the result of addition 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 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 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	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.

34.5.9 Word subtraction

Word subtraction is for subtracting 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, and then the resulting value is stored in the SMSGn register specified in the first operand. If the result of subtraction 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 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 SMSGn, SMSGm

Equivalent CPU operation: $\text{SMSGn}, \text{SCY} \leftarrow \text{SMSGn} - \text{SMSGm}$

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	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.

34.5.10 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

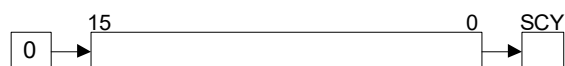
Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			
	7	6	5	4	3	2	1	0
	Second operand (SMSGm)				0	0	1	0

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.

34.5.11 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.

34.5.12 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 -1 and 1 to 31 as the first and second operands, respectively.

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: $SMSCV[4:0] \leftarrow SMSCV[4:0] + jdisp8$ 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.

34.5.13 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 -1 and 1 to 31 as the first and second operands, respectively.

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: $SMSCV[4:0] \leftarrow SMSCV[4:0] + jdisp8$ 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.

34.5.14 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 -1 and 1 to 31 as the first and second operands, respectively.

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: $SMSCV[4:0] \leftarrow SMSCV[4:0] + jdisp8$ 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.

34.5.15 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 -1 and 1 to 31 as the first and second operands, respectively.

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: $SMSCV[4:0] \leftarrow SMSCV[4:0] + jdisp8$ 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.

34.5.16 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.

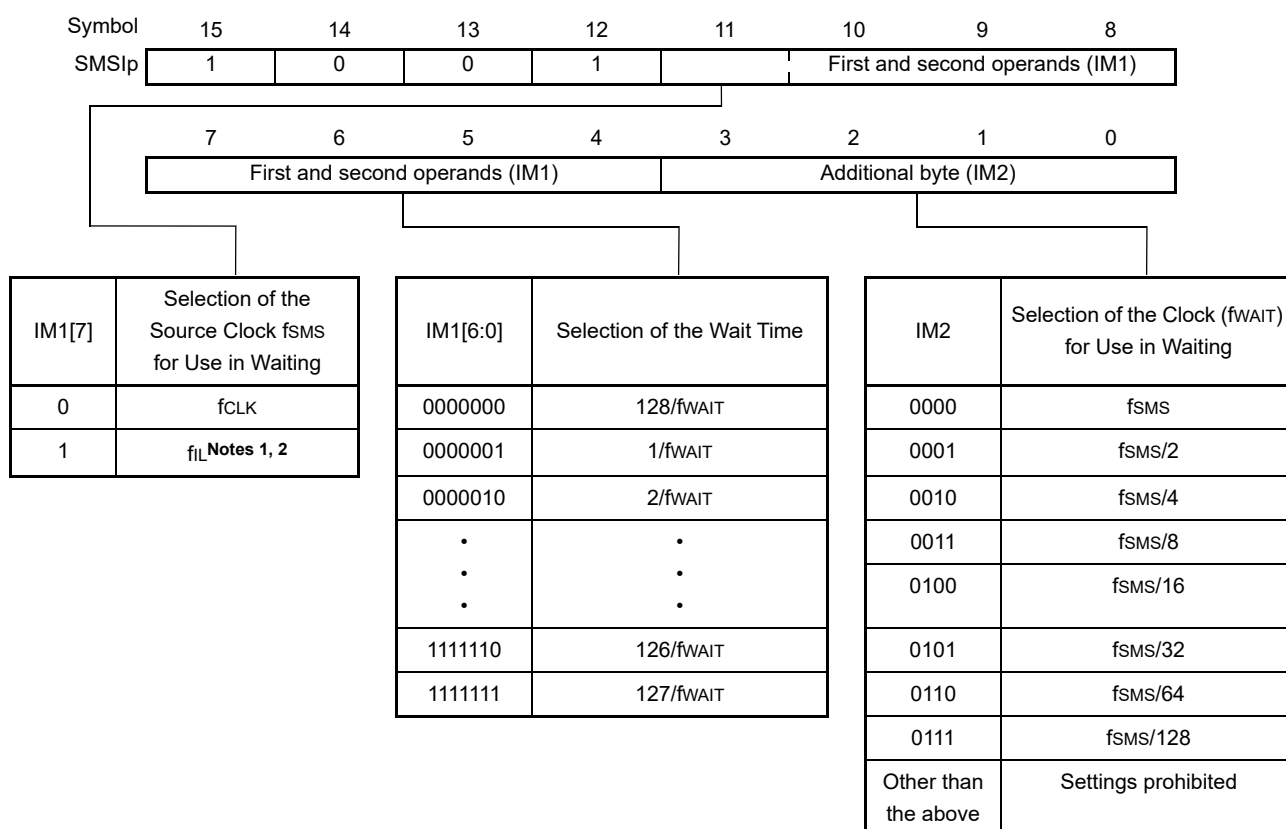
Sequencer code: 1001B

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.

34.5.17 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
SMSIp	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.

34.5.18 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
SMSIp	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 **34.5.17 Conditional wait 1 (bit = 1)**, **1. Number of clock cycles required for conditional wait 1**.

34.5.19 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: SMSSTAT $\leftarrow$ 0, SMSCV[4:0] $\leftarrow$ 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.

34.5.20 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.

34.5.21 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 6) has been set to 0.

34.6 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 2
SNOOZE mode	Operation continues. Notes 3, 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. 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 **34.4.2 Memory space allocated to the sequencer.**

Note 3. 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 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.

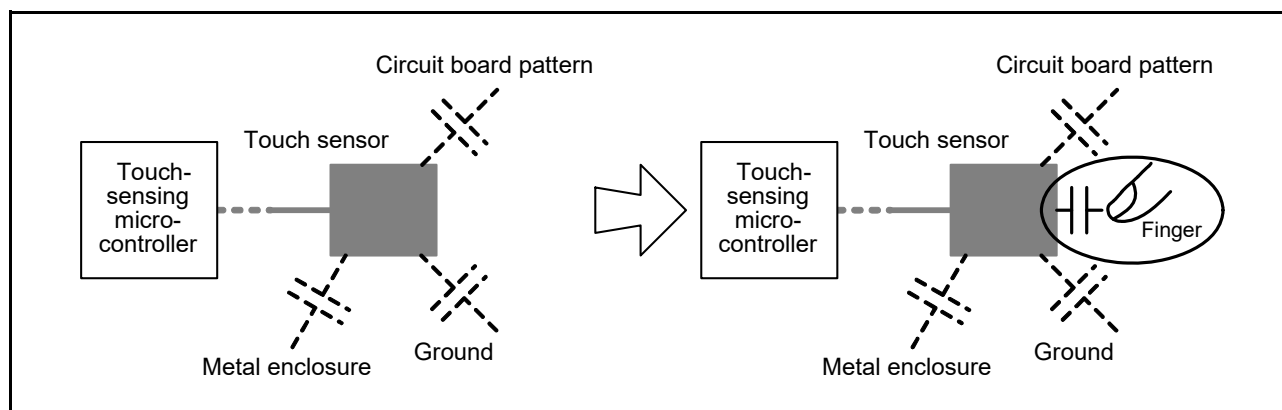
Section 35 Capacitive Sensing Unit (CTS2La)

The number of measurement channels of the capacitive sensing unit depends on the product.

Number of Pins	44	48	52	64	80	100
Number of CTS2La output channels	16: TS00 to TS02, TS09, TS10, and TS24 to TS34	18: TS00 to TS02, TS09 to TS11, TS19, and TS24 to TS34	20: TS00 to TS02, TS09 to TS12, TS19, TS20, and TS24 to TS34	25: TS00 to TS02, TS07 to TS12, TS16, TS19 to TS21, and TS24 to TS35	34: TS00 to TS04 and TS07 to TS35	36: TS00 to TS35

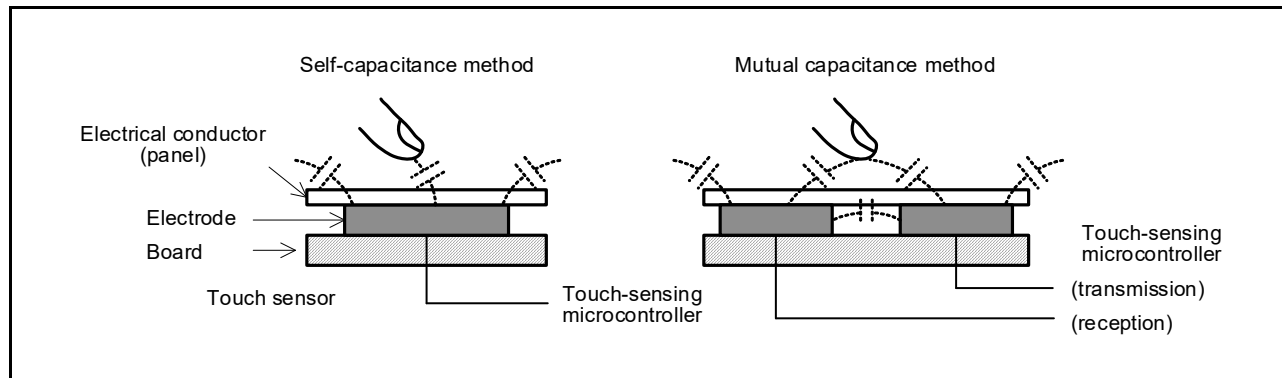
The capacitive sensing unit (CTS2La) measures the electrostatic capacitance of the capacitive sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTS2La to detect whether a finger is in contact with the capacitive sensor. The electrode surface of the capacitive 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 35 - 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 35 - 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 35 - 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, see **35.3.1 Principles of measurement**. For details on the development of touch applications, see the application note, **Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)**.

35.1 Overview

35.1.1 Functional overview of the CTSU

Table 35 - 1 lists the CTSU functions and **Figure 35 - 3** shows a block diagram of the CTSU. **Figure 35 - 4** shows the configuration of clock signals for sensor driving pulse output.

Table 35 - 1 Functional Overview of the CTSU

Item		Configuration
Module type name		CTS2La
CTSU operating voltage condition		VDD = 1.8 to 5.5 V
Operating clock ^{Note}		fCLK (at least 1 MHz), fCLK/2, fCLK/4, or fCLK/8
Pins	Electrostatic capacitance measurement	TSM (m = 00 to 35) up to 36 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		• Spread spectrum for sensor driving pulses • Random phase-shifting for sensor driving pulses • Majority decision from multi-frequency measurement
Adjustment for each pin		• Offset current adjustment • Sensor driving pulse frequency specification • Measurement time specification
Measurement start conditions		• Software trigger • External trigger (event input from the ELCL)
Low-power function		SNOOZE mode supported
Interrupt requests	DTC activation sources	• Request to write to a configuration register of an individual CTSU channel • Request to transfer data measured by the CTSU
	Interrupt source	• Write request interrupt for a configuration register of an individual CTSU channel (INTCTSUWR) • Request to transfer data measured by the CTSU (INTCTSURD) • CTSU measurement end (INTCTSUFN)
Transmission power switching of the mutual capacitance method		The source of power to the pins can be switched at the time of transmission.

Note The accuracy of measurement by the capacitive sensing unit (CTS2La) depends on the accuracy of the operating clock. When the CTS2La is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IN}) or high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}). Do not select the subsystem clock (f_{SUB}) as the CPU/peripheral hardware clock (f_{CLK}).

As shown in **Figure 35 - 3**, the CTSU consists of a status control block, a trigger control block, a clock control block, a channel control block, a port control block, a sensor driving pulse generator, a measurement block, an interrupt control block, a SNOOZE control block, and control registers.

Figure 35 - 3 Block Diagram of the CTSU

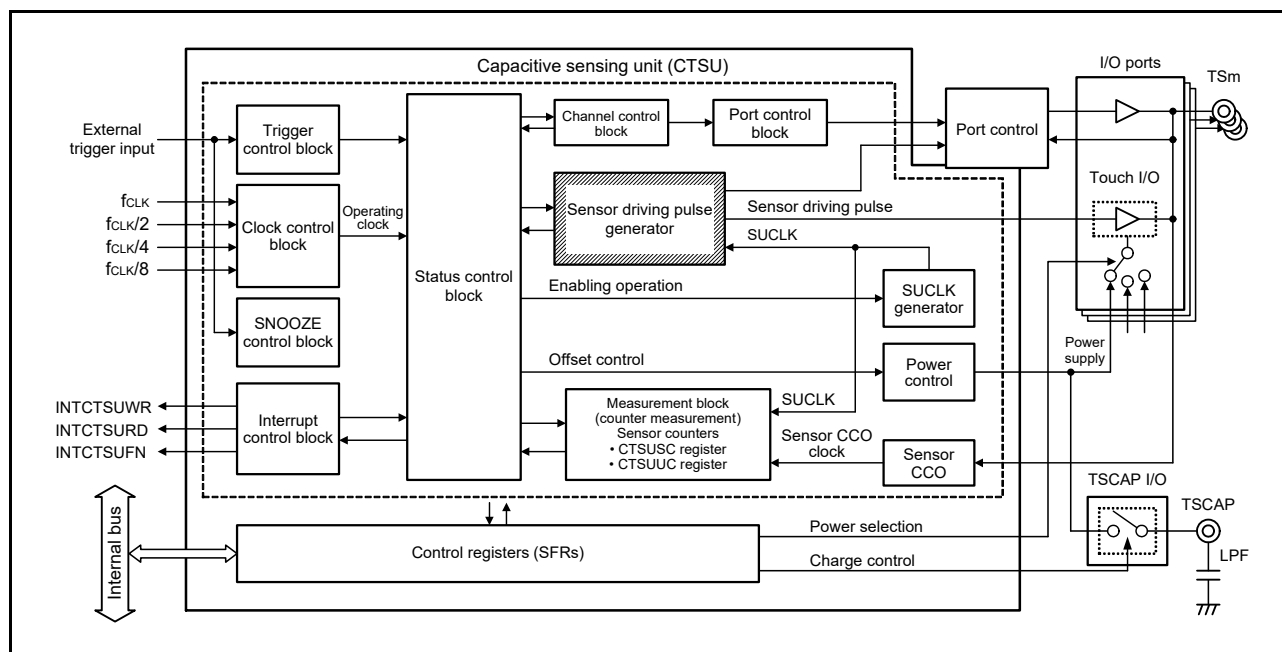


Figure 35 - 4 Configuration of Clock Signals for Sensor Driving Pulse Output

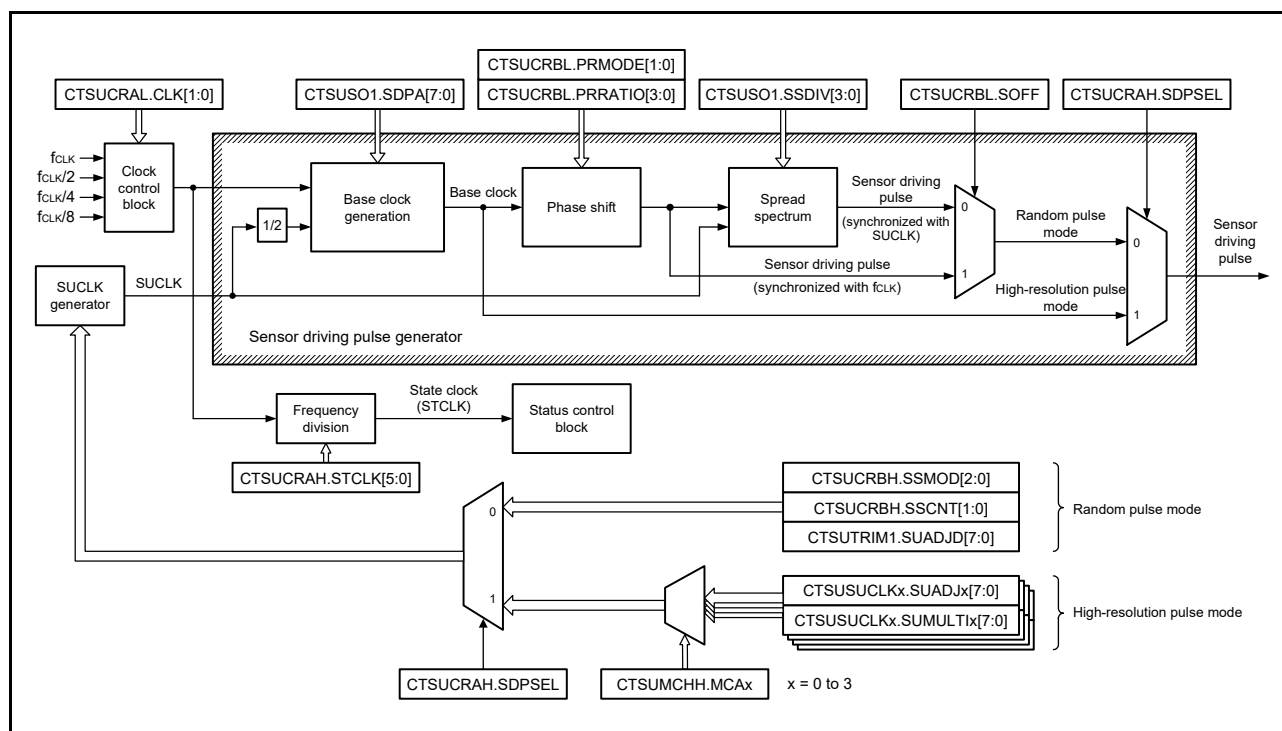


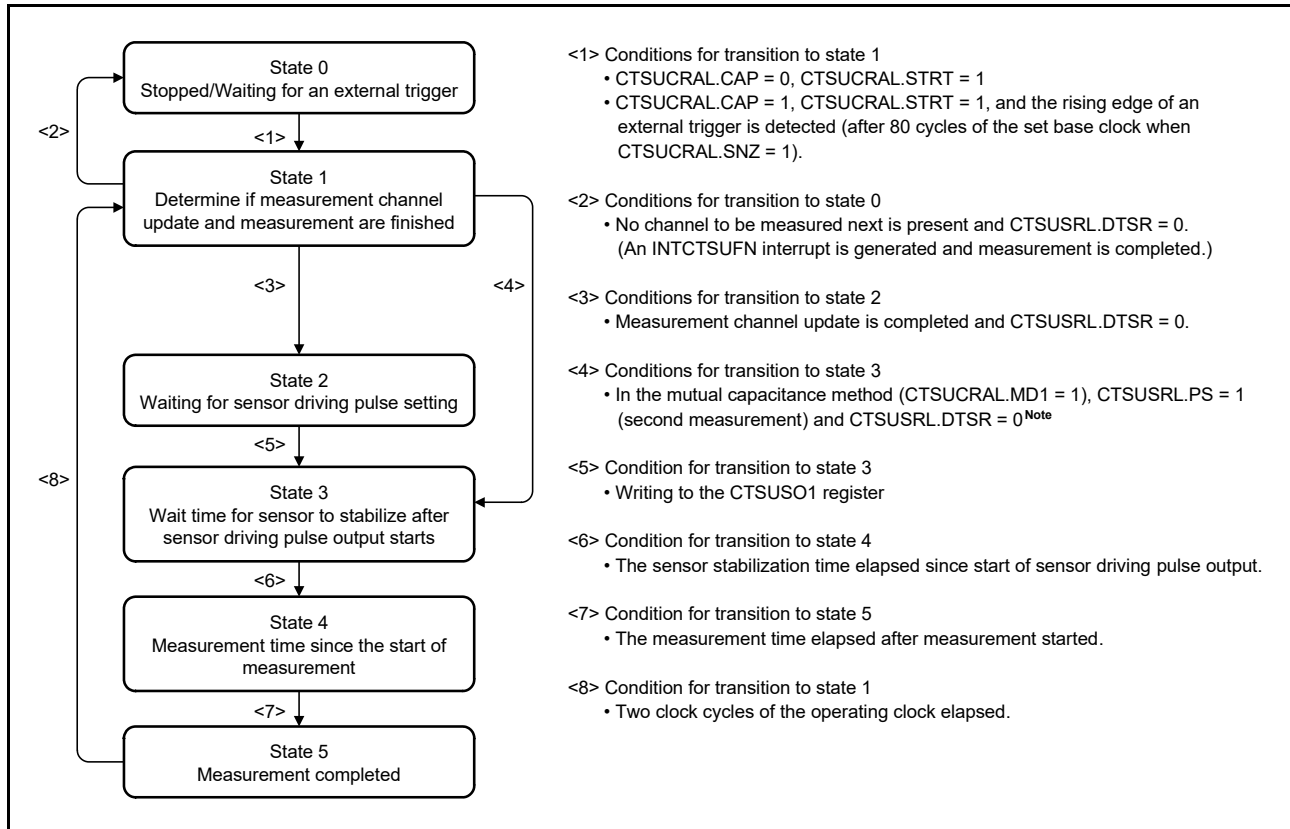
Table 35 - 2 Input and Output Pins of the CTSU

Pin name	Input/output	Function
TSM (m = 00 to 35)	Input/output	Electrostatic capacitance measurement pins (touch pins)
TSCAP	—	Connection pin to capacitor for measurement secondary power

35.1.2 Measurement states

The CTSU has a state controller. The measurement states are common to both modes of measurement. The measurement status counter indicates the current state of measurement, and its value can be read from the CTSUSRL.STC[2:0] bits. **Figure 35 - 5** shows the transitions between the measurement states.

Figure 35 - 5 Transitions between the Measurement States



Note When CTSUSRL.DTSR = 1, waiting until the reading of the previous measurement result is required.

The status counter makes the transition to state 0 when measurement has been completed on all of the specified measurement channels. The CTSUCRAL.STRT bit is set to 0 by hardware when the software trigger is to be used. When an external trigger is to be used, the CTSUCRAL.STRT bit retains the value 1, and the CTSU waits for the next trigger. When operation is forcibly stopped (by writing 0 to the CTSUCRAL.STRT bit and 1 to the CTSUCRAL.INIT bit at the same time) during measurement or is in the state of waiting for a trigger, the state makes the transition to state 0 and measurement is stopped. If no channels for measurement are set in the CTSUMCHL, CTSUMCHH, CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHTRCAL, CTSUCHTRCAH, and CTSUCHTRCBL registers, a measurement end interrupt (INTCTSUFN) is generated immediately after the transition to state 1, and the state then makes the transition to state 0.

<R>

The following are the cases when there are no channels for measurement.

- <R> • No target measurement channels are specified in the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers.
- <R> • In single scan mode, no channels specified in the CTSUMCHL and CTSUMCHH registers are set as targets for measurement in the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers.
- <R> • In the self-capacitance method, there is no reception channel for measurement in the following combinations of registers: CTSUCHACAL/CTSUCHACAH/CTSUCHACBL and CTSUCHTRCAL/CTSUCHTRCAH/CTSUCHTRCBL.
- <R> • In the mutual capacitance method, there is no transmission channel or reception channel for measurement in the following combinations of registers: CTSUCHACAL/CTSUCHACAH/CTSUCHACBL and CTSUCHTRCAL/CTSUCHTRCAH/CTSUCHTRCBL.

35.2 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, and BL (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL)
- CTSU channel transmit/receive control registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL)
- CTSU status register L (CTSUSRL)
- CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)
- CTSU sensor counter registers (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)
- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

35.2.1 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. 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 35 - 6 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	DACEN	SMS2EN	CMPEN	TML32EN	DTCEN	UTA0EN	0	CTS2EN
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 Be sure to clear bit 1 to 0.

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

Figure 35 - 7 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	DACRES	SMSRES	CMPRES	TML32RES	0	0	0	CTSURES
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 3 to 1 to 0.

35.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCRAL register can respectively be set as the CTSUCR1 and CTSUCR0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCRAH register can respectively be set as the CTSUCR3 and CTSUCR2 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (1/8)

Address: F0500H, F0501H (CTSUCRAL), F0502H, F0503H (CTSUCRAH)

After reset: 0000H

R/W: R/W

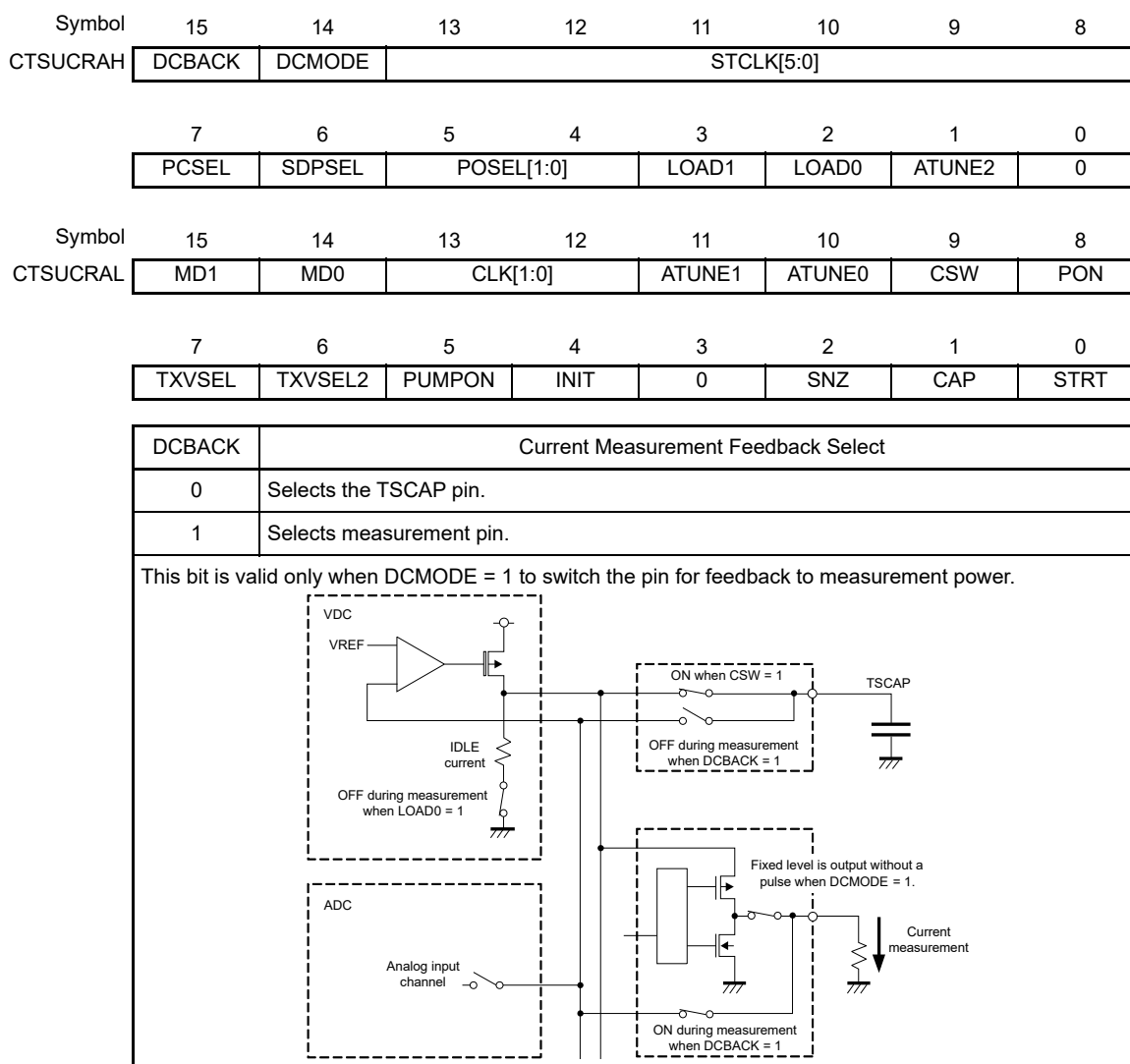


Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (2/8)

DCMODE	Current Measurement Mode Select					
0	Normal mode					
1	Current measurement mode					
Setting this bit to 1 enables measurement of the current without using sensor driving pulses. A fixed level is output through the reception pin for measuring the capacitance in accord with the value in the CTSUDBGR0.IOC bit, and the sensor driving pulses are stopped. CTSUDBGR0.IOC = 0: High-level output CTSUDBGR0.IOC = 1: Low-level output						

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 the state clock (STCLK). STCLK is related to the measurement time and the frequency adjustment cycle of SUCLK. The division value is determined by the following equation. Division value = (STCLK[5:0] + 1) × 2 (divided by 2 to 128) Setting these bits so that STCLK will be 0.5 MHz (2 μs) is recommended.						

PCSEL	Boost Clock Select					
0	Sensor driving pulse					
1	STCLK					
The PCSEL bit selects the clock for the boost circuit.						

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (3/8)

SDPSEL	Sensor Driving Pulse and SUCLK Select	
0	Random pulse mode • Sensor driving pulse Sensor driving pulses synchronized with fCLK are generated with randomly extended widths at low level by using pseudo-random numbers based on a base clock which is the operating clock divided by the value of the CTSUSO1.SDPA[7:0] bits. • SUCLK SUCLK is generated in accord with the settings of the CTSUCRBH.SSCNT[1:0], CTSUCRBH.SSMOD[2:0], and CTSUTRIM1.SUADJD[7:0] bits. SUCLK is used for spreading the spectrum of the sensor driving pulses.	
1	High-resolution pulse mode • Sensor driving pulse The clock (base clock) obtained by dividing SUCLK by the CTSUSO1.SDPA[7:0] bits is used to produce the sensor driving pulses. • SUCLK Setting the CTSUSUCLKx register (x = 0 to 3) causes clock recovery control to proceed and the CTSUSUCLKx.SUADJx[7:0] bits are updated. SUCLK is generated in accord with this setting. Set the CTSUSUCLKx register before setting this bit to 1. Do not change the CTSUSUCLKx register setting while this bit is 1.	
The SDPSEL bit is used to switch between the random pulse mode and high-resolution pulse mode. Setting this bit determines how sensor drive pulses and SUCLK are generated.		

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	Outputs a pulse in phase with the transmit channel (by the settings of the TXVSEL and TXVSEL2 bits)

The POSEL[1:0] bits select the non-measured pin output.

[Example] When measurement proceeds by the self-capacitance method using the electrode shown below to enable the active shield function:

- Shield pin: TS03
Set the transmit/receive control register to 1 to output a transmit pulse to use TS03 as a shield pin.
- Measurement pins: TS00 to TS02
Set the POSEL[1:0] bits to 11B. The same pulse as the shield pin is output from the non-measured pin.

TS03

TS00

TS01

TS02

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (4/8)

LOAD[1:0]		Measurement Load Control
0	0	2.5-μA constant current load mode (for normal measurement)
0	1	No-load mode
1	0	20-μA constant current load mode (for calibration)
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. When in resistive load mode, the amount of resistive load changes depending on the settings of the ATUNE2 and ATUNE1 bits.		

ATUNE2	Current Range Switching 2	
0	When the ATUNE1 bit is 0: 80 μA When the ATUNE1 bit is 1: 40 μA	
1	When the ATUNE1 bit is 0: 20 μA When the ATUNE1 bit is 1: 160 μA	
Setting this bit in combination with the ATUNE1 bit specifies the maximum supply current (measurement range) by changing the amount of the resistive load on the measurement power supply.		
CTSUCRAH.ATUNE2	CTSUCRAL.ATUNE1	Measurement range
0	0	80 μA
0	1	40 μA
1	0	20 μA
1	1	160 μA

MD1	Measurement Mode Select 1
0	Self-capacitance method Set the pin to be used for reception in measurement (by setting the CHTRCm bit in the CTSUCHTRCAL, CTSUCHTRCAH, or CTSUCHTRCBL register to 0). Set the pin to be used as an active shield for transmission (by setting CHTRCm to 1). If a pin set for transmission is present, the transmission pin outputs a pulse in the same phase as that for the channel being measured. When all TSm pins are set for reception, the measurement of reception pins only proceeds.
1	Mutual capacitance method The measurement proceeds with the TSm pin set for transmission (CHTRCm = 1) and the TSm pin set for reception (CHTRCm = 0). When all CHTRCm bits are set to 0, no measurement proceeds.
The MD1 bit selects the self-capacitance method or the mutual capacity 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.	

<R>

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (5/8)

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) by using the prescaler.		

ATUNE1	Current Range Switching 1
0	When the ATUNE2 bit is 0: 80 μ A When the ATUNE2 bit is 1: 20 μ A
1	When the ATUNE2 bit is 0: 40 μ A When the ATUNE2 bit is 1: 160 μ A
Setting this bit in combination with the ATUNE2 bit specifies the maximum supply current (measurement range) by changing the amount of the resistive load on the measurement power supply. For details, see the description for the ATUNE2 bit.	

ATUNE0	Power Supply Voltage Setting
0	Measurement power-supply voltage = 1.5 V
1	Measurement power-supply voltage = 1.2 V ^{Note 1}
This bit is used to control the power-supply voltage of the CTSU. Set this bit to 1 when $V_{DD} < 2.4$ V. This bit can be set to either 0 or 1 when 2.4 V $\leq V_{DD}$.	

CSW	Secondary Power Supply Capacitor for Measurement Connection
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 setting this bit to 1, use the I/O port pin assigned to TSCAP (P57) to output a low-level signal, and discharge the charge from the secondary measurement power-supply capacitor which is connected to the TSCAP pin. 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 controls the supply of the power (1.5 V) for measurement. Before setting this bit to 1, set the PUMPON bit as required and set the CSW bit to 1. Do not set this bit to 1 while the CSW bit is set to 0.	

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (6/8)

TXVSEL	TXVSEL2	Transmission Power Supply Select
0	0	Power for I/O ports
1	0	Power for the internal logic circuits Use this setting when the active shield is to be used.
x	1	Dedicated power for the CTSU Use this setting when the mutual capacitance method is to be used.

PUMPON	Boost Circuit Control
0	Disables power boosting.
1	Enables power boosting.

The PUMPON bit controls power boosting. Set this bit to 1 when $V_{DD} < 4.5$ V.

INIT	Initialization of the Control Block
0	—
1	Initializes the internal control registers.

Setting this bit to 1 initializes the following registers.

- CTSUSC, CTSUUC
- CTSUMCHL, CTSUMCHH
- All bits in CTSUSRL except bits ICOMP0 and ICOMP1

To forcibly stop operation, be sure to set the STRT bit to 0 and the INIT bit to 1 at the same time. Doing so stops the measurement and initializes the internal control registers. Do not set the STRT bit to 1 (starting measurement by the CTSU) and the INIT bit to 1 at the same time. This bit is read as 0.

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (7/8)

SNZ	SNOOZE Enable
0	Disables the SNOOZE mode.
1	Enables the SNOOZE mode.

Setting this bit to 1 places the CTSU in a suspended state, which reduces power consumption while on standby from measurement. The SNZ bit enables measurement in SNOOZE mode when an external trigger is selected (CAP = 1) as the condition for starting measurement.

- When an external trigger is selected as the condition for starting the measurement (CAP = 1)
When the SNOOZE mode is to be used, set this bit to 1, then set STRT to 1 (starting measurement by the CTSU) to place the CPU in the STOP mode. If an external trigger is detected while in STOP mode, the CTSU is placed in the SNOOZE mode and starts measurement. After a measurement end interrupt, use software to clear this bit to 0.
- When a software trigger is selected as the condition for starting the measurement (CAP = 0)
The CTSU is placed in the suspended state (low-power mode) under the conditions listed in the table below. Use an interrupt to cause the CPU to resume from the STOP mode while the CTSU is in the suspended state (low-power mode). To start measurement following resumption by the CPU, set this bit to 0 and then set the STRT bit to 1 to start measurement by the CTSU.

PON	SNZ	CAP	STRT	External trigger	State of the CTSU hardware
0	0	0	0	—	Stopped
1	0	0	0	—	State of waiting for the start of measurement
1	0	0	1	—	Measurement is proceeding in the normal mode
1	1	1	0	—	Preparing for an external trigger to start measurement
1	1	1	1	Not present (waiting)	Suspended (waiting for a trigger)
1	1	1	1	Present (operating)	Measurement in the SNOOZE mode
1	1	0	0	—	Suspended (low power mode)
Other than above					Setting prohibited

When using the SNOOZE mode function with the high-resolution pulse mode selected for the sensor driving pulses (SDPSEL = 1), the clock of the voltage boosting circuit cannot be selected for the sensor driving pulses (PCSEL = 0).

CAP	Measurement Start Trigger Select
0	Software trigger
1	External trigger (event input from the ELCL)
The CAP bit specifies the measurement start condition. For details, see the description for the STRT bit.	

Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (8/8)

STRT	Measurement Start
0	Stops measurement
1	Starts measurement (measurement in progress or waiting for an external trigger)

The STRT bit specifies whether to start or stop CTSU measurement.

- When a software trigger is selected as the condition for starting the measurement (CAP = 0)
Writing 1 to the STRT bit starts measurement. At the end of measurement, this bit is cleared to 0 by hardware.
- When an external trigger is selected as the condition for starting the measurement (CAP = 1)
Writing 1 to this bit drives the CTSU into the state of waiting for an external trigger. Measurement starts on detection of an external trigger. After the measurement is completed, the CTSU waits for the next external trigger. To release the CTSU from the state of waiting for an external trigger, set this bit to 0 and the INIT bit to 1 at the same time.

<CTSU state>

STRT	CAP	CTSU state
0	0	Stopped
0	1	Stopped
1	0	Measurement in progress
1	1	Measurement in progress or waiting for an external trigger ^{Note 2}

If 1 is written to the STRT bit by software when the STRT bit is already set to 1, the writing is ignored and operation continues. To have software forcibly stop operation while the STRT bit is set to 1, be sure to set the STRT bit to 0 and the INIT bit to 1 at the same time.

Note 1. The active shield cannot be used when ATUNE0 = 1.

Note 2. The state in terms of the two alternatives can be read from the CTSUSRL.STC[2:0] bits.

- CTSUSRL.STC[2:0] ≠ 000B: Measurement in progress
- CTSUSRL.STC[2:0] = 000B: Waiting for an external trigger

Caution Set the bits of this register other than the STRT and INIT bits while STRT = 0 (measurement stopped).

Remark 1. x = 00 to 35

Remark 2. x: Don't care

35.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher-order bits of the CTSUCRBL register can be set as the CTSUSST register by an 8-bit memory manipulation instruction, whereas the lower-order bits can be set as the CTSUSDPRS register by a 1-bit or 8-bit memory manipulation instruction.
- The higher-order bits of the CTSUCRBH register can be set as the CTSUDCLKC register by an 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 35 - 9 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 Spread Spectrum Control	
0	0	CTSUTRIM1.SUADJD[7:0] + 00H	
0	1	CTSUTRIM1.SUADJD[7:0] + 20H	
1	0	CTSUTRIM1.SUADJD[7:0] + 40H	
1	1	CTSUTRIM1.SUADJD[7:0] + 60H	
The SSCNT[1:0] bits adjust the frequency of SUCLK specified in the CTSUTRIM1.SUADJD[7:0] bits.			

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 SSMOD[2:0] bits set the number of spread spectrum cycles of SUCLK.			
The oscillation frequency of SUCLK is determined by the frequency specified by the CTSUTRIM1.SUADJD[7:0] bits and SSCNT[1:0] bits, as well as the setting of this bit.			

Figure 35 - 9 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 driving pulse supply until the TSCAP pin voltage becomes stable. The TSCAP pin voltage is stabilized by supplying sensor driving pulses. The following shows the correspondence between the setting of the SST[7:0] bits and the number of cycles.								
• In random pulse mode (CTSUCRAH.SDPSEL = 0)								
The stabilization wait time is specified in units of the number of cycles of the sensor driving pulse.								
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
Number of cycles = 2 × (value of these bits + 1)								
Set the stabilization wait time so that the following relation is satisfied.								
Number of cycles set by CTSUCRBL.SST[7:0] ≥ (PRRATIO[3:0] + 1)								
• In high-resolution pulse mode (CTSUCRAH.SDPSEL = 1)								
The stabilization wait time is specified in units of the number of the STCLK cycles.								
0	0	0	0	0	0	0	0	Setting prohibited ^{Note 1}
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
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 pseudo-random number is generated per cycle to output a value of 1 or 0 (1-bit random number generation).

SOFF ^{Note 2}		Spread Spectrum Disable Control
0	Enables the spread spectrum.	
1	Disables the spread spectrum.	

The SOFF bit enables or disables the spread spectrum to prevent synchronous noise. Set this bit to 1 when sensor driving pulses (synchronized with fCLK) are to be used. This bit is valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).

Figure 35 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (3/3)

PRMODE[1:0] ^{Note 2}		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)
The PRMODE[1:0] bits set the number of cycles for updating the pseudo-random number generation. The number of basic pulses is twice the number of cycles selected by these bits. These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).		

PRRATIO[3:0] ^{Note 2}	Phase Shift Frequency
The PRMODE[1:0] bits set the phase shift cycle for pseudo-random number generation. The number of measurement pulses and measurement time are specified as follows depending on the setting of these bits. Number of measurement pulses = number of basic pulses (set by the PRMODE[1:0] bits) × (value of the PRRATIO[3:0] bits + 1) Measurement time = (number of measurement pulses) + (number of basic pulses – 2) × 0.25 × base clock cycle These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).	

Note 1. Do not set the SST[7:0] bits to 00H when CTSUCRAH.SDPSEL = 1.

Note 2. The SOFF, PRMODE[1:0], and PRRATIO[3:0] bits are only valid when the CTSUCRAH.SDPSEL bit is 0 (random pulse mode).

Caution Set the CTSUCRBL and CTSUCRBH registers when the CTSUCRAL.STRT bit is 0.

35.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUMCHL register can respectively be set as the CTSUMCH1 and CTSUMCH0 registers by an 8-bit memory manipulation instruction.
- The lower-order bits of the CTSUMCHH register can be set as the CTSUMFAF register by a 1-bit or 8-bit memory manipulation instruction.

The value of the CTSUMCHL register following a reset is 3F3FH, and that of the CTSUMCHH register is 0000H.

Figure 35 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH) (1/2)

Address: F0508H, F0509H (CTSUMCHL), F050AH, F050BH (CTSUMCHH)

After reset: 3F3FH (CTSUMCHL), 0000H (CTSUMCHH)

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]					
MCAx	Multi-clock x Enable							
0	Disabled							
1	Enabled							
The MCAx bit sets the clocks to be used when the measurement is in progress by using SUCLKs with two or more frequencies (multi-clock measurement). For measurement using three clocks, set three bits out of the four bits (MCA3 to MCA0) to 1. When measurement starts with two or more MCAx bits enabled, valid channels are scanned in ascending order from the MCA0 bit (SUCLK0). After a channel has been measured, if the next valid clock (SUCLKx) is present, the same channel is measured using that clock. If no valid clock is present, the next channel is measured. The SUCLK frequency is set by the CTSUSUCLKx register. The following shows the relationship between the MCAx bits and the CTSUSUCLKx register. • MCA0: SUCLK0 set by the CTSUSUCLK0 register • MCA1: SUCLK1 set by the CTSUSUCLK1 register • MCA2: SUCLK2 set by the CTSUSUCLK2 register • MCA3: SUCLK3 set by the CTSUSUCLK3 register								

Figure 35 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUCHL, CTSUCHH) (2/2)

MCH1[5:0] ^{Note}						Measurement Channel 1
- In single scan mode (CTSUCRAL.MD0 = 0) The MCH1[5:0] bits set the transmit channel to be measured. Do not set the channels for which the measurement is disabled by the CTSUCHACAH, CTSUCHACAL, or CTSUCHACBL register. If such channels are set, measurement is completed immediately after it starts. - In multi-scan mode (CTSUCRAL.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.						
0	0	0	0	0	0	TS00
0	0	0	0	0	1	TS01
0	0	0	0	1	0	TS02
0	0	0	0	1	1	TS03
:	:	:	:	:	:	:
1	0	0	0	1	0	TS34
1	0	0	0	1	1	TS35
Other than above						Setting prohibited

MCH0[5:0] ^{Note}						Measurement Channel 0
- In single scan mode (CTSUCRAL.MD0 = 0) The MCH0[5:0] bits set the reception channel to be measured. Do not set the channels for which the measurement is disabled by the CTSUCHACAH, CTSUCHACAL, or CTSUCHACBL register. If such channels are set, measurement is completed immediately after it starts. - In multi-scan mode (CTSUCRAL.MD0 = 1) The MCH1[5:0] bits indicate the value of the reception channel that is being measured, and writing to these bits has no effect.						
0	0	0	0	0	0	TS00
0	0	0	0	0	1	TS01
0	0	0	0	1	0	TS02
0	0	0	0	1	1	TS03
:	:	:	:	:	:	:
1	0	0	0	1	0	TS34
1	0	0	0	1	1	TS35
Other than above						Setting prohibited

Note Do not modify the CH1[5:0] and MCH0[5:0] bits during measurement (CTSUCRAL.STRT = 1). Otherwise, operation is not guaranteed. The value of the CH1[5:0] and MCH0[5:0] bits is 11111B when the measurement is stopped.

Remark x = 0 to 3

35.2.6 CTSU channel enable control registers AL, AH, and BL (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCHACAL register can respectively be set as the CTSUCHAC1 and CTSUCHAC0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHACAH register can respectively be set as the CTSUCHAC3 and CTSUCHAC2 registers by a 1-bit or 8-bit memory manipulation instruction.
- The lower-order bits of the CTSUCHACBL register can respectively be set as the CTSUCHAC5 and CTSUCHAC4 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 35 - 11 Format of CTSU Channel Enable Control Registers AL, AH, and BL
(CTSUCHACAL, CTSUCHACAH, CTSUCHACBL)

Address: F050CH, F050DH (CTSUCHACAL), F050EH, F050FH (CTSUCHACAH),
F0510H, F0511H (CTSUCHACBL)

After reset: 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
CTSUCHACBL	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	CHAC35	CHAC34	CHAC33	CHAC32
CHACm	Channel Enable Control							
0	Disables measurement.							
1	Enables measurement.							
The CHACm bit enables or disables the measurement of the TSm pin. Setting the CHACm bit to 1 enables the capacitance measurement of the TSm pin (both transmission and reception).								

Caution 1. Make sure that the bits to which no measurement pins have been assigned are set to 0.

Caution 2. Set the CTSUCHACAH, CTSUCHACAL, and CTSUCHACBL registers when CTSUCRAL.STRT is 0.

Remark m = 00 to 35

35.2.7 CTSU channel transmit/receive control registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCHTRCAL register can respectively be set as the CTSUCHTRC1 and CTSUCHTRC0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHTRCAH register can respectively be set as the CTSUCHTRC3 and CTSUCHTRC2 registers by a 1-bit or 8-bit memory manipulation instruction.
- The lower-order bits of the CTSUCHTRCBL register can respectively be set as the CTSUCHTRC5 and CTSUCHTRC4 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 35 - 12 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL)

Address: F0514H, F0515H (CTSUCHTRCAL), F0516H, F0517H (CTSUCHTRCAH),
F0518H, F0519H (CTSUCHTRCBL)

After reset: 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
CTSUCHTRCBL	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	CHTRC35	CHTRC34	CHTRC33	CHTRC32
CHTRCm	Channel Transmit/Receive Control							
0	Reception							
1	Transmission							
The CHTRCm bit assigns the TS _m pin for use in transmission or reception in accord with the value of the HTRCm bit. When the self-capacitance method is in use (CTSUCRAL.MD1 = 0), setting this bit to 1 (transmission) allows the corresponding pin to be used as an active shield signal output. ^{Note}								

Note When the active shield output is to be used, do not set multiple bits to 1.

Caution 1. Make sure that the bits to which no measurement pins have been assigned are set to 0.

Caution 2. Set the CTSUCHTRCAH, CTSUCHTRCAL, and CTSUCHTRCBL registers when CTSUCRAL.STRT is 0.

Remark m = 00 to 35

35.2.8 CTSU status register L (CTSUSRL)

This register can be set by a 16-bit memory manipulation instruction. The higher- and lower-order bits of this register can respectively be set as the CTSUST and CTSUST1 registers by a 1-bit or 8-bit memory manipulation instruction. The value of the CTSUSRL register following a reset is 0000H.

Figure 35 - 13 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 ^{Note 1}	Mutual Capacitance Measurement State
0	First measurement
1	Second measurement

In the mutual capacitance method, which handles two measurements (CTSUCRAL.MD1 = 1), the PS bit indicates whether the measurement for each channel is the first or second measurement. The value of this bit is 0 when measurement is stopped or in self-capacitance method (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 register 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.

[Clearing conditions]

- 0 has been written to SUCKOVF after reading SUCKOVF = 1
- The CTSUCRAL.INIT bit is set to 1^{Note 2}

[Setting condition]

- The CTSUUC register overflows during measurement

Figure 35 - 13 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 register 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. [Clearing conditions] • 0 has been written to SUCKOVF after reading SUCKOVF = 1 • The CTSUCRAL.INIT bit is set to 1^{Note 2} [Setting condition] • The CTSUUC register overflows during measurement			

DTSR ^{Note 1}	Data Transfer State		
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. [Clearing conditions] • Reading data from the CTSUSC or CTSUUC register by using software or the DTC • The CTSUCRAL.INIT bit is set to 1 [Setting condition] • The measurement result is stored in the CTSUSC and CTSUUC registers following the completion of measurement.			

STC[2:0] ^{Note 1}			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 state of measurement. For details, see 35.1.2 Measurement states .			

ICOMP0 ^{Note 1}	Abnormal Voltage (Reference Resistance Comparison Result)		
0	Normal		
1	Abnormal TSCAP voltage		
Abnormal voltage is detected based on the comparison result of the reference resistance. This bit is cleared to 0 when the CTSUCRAL.PON bit is set to 0 or the ICOMPRST bit is set to 1. When this bit is set to 1 (abnormal voltage detection), the CTSUSU register indicates 0000H.			

Figure 35 - 13 Format of CTSU Status Register L (CTSUSRL) (3/3)

ICOMP1 Note 1	Current Error	
0	Normal	
1	Abnormal current	
Abnormal output current from the power supply used for the measurement is detected. This bit is cleared to 0 when the CTSUCRAL.PON bit is set to 0 or the ICOMPRST bit is set to 1. When this bit is set to 1 (abnormal current detection), the CTSUSC register indicates FFFFH.		
ICOMPRST	ICOMP0 and ICOMP1 Reset	
Writing 1 to this bit clears the ICOMP1 and ICOMP0 bits to 0. This bit is read as 0.		
MFC[1:0]		Multi-clock Counter
0	0	MCA0 (SUCLK0)
0	1	MCA1 (SUCLK1)
1	0	MCA2 (SUCLK2)
1	1	MCA3 (SUCLK3)
The MFC[1:0] bits indicate the clock being used for the multi-clock measurement, i.e., when CTSUCRAH.SDPSEL = 1 and two or more of the CTSUMCH.MCAn bits (n = 0 to 3) are set to 1. Only writing 00B is allowed.		

Note 1. This bit is read-only.

Note 2. To clear the SUCKOVF and SENSOVF bits while CTSUCRAL.INIT is set to 1, do so when the setting of the CTSUCRAL.STRT bit is 0.

35.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)

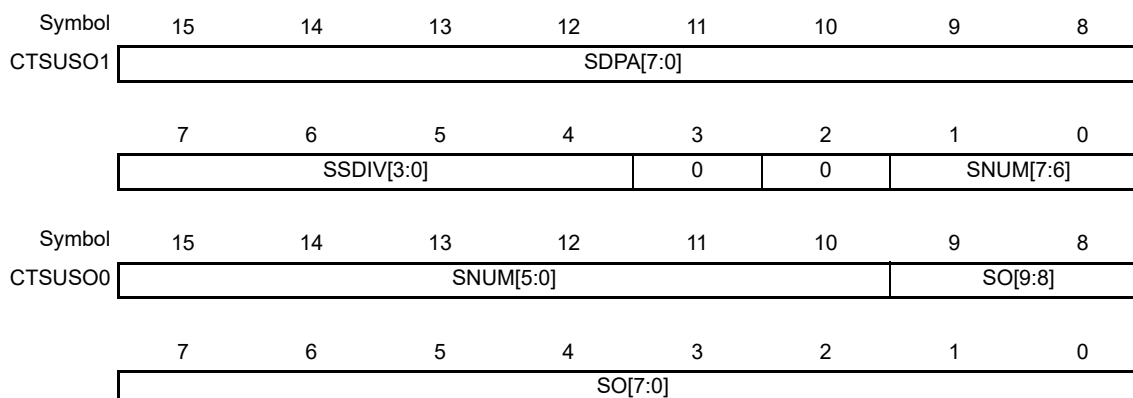
These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 35 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (1/3)

Address: F0520H, F0521H (CTSUSO0), F0522H, F0523H (CTSUSO1)

After reset: 0000H

R/W: R/W



Write to the CTSUSO0 and CTSUSO1 registers after an INTCTSUWR interrupt is generated. Writing to the CTSUSO1 register causes a transition to state 3. Be sure to write to the CTSUSO0 register before writing to the CTSUSO1 register. Also, set all 16 bits of the CTSUSO1 register at once.

Figure 35 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (2/3)

SDPA[7:0]								Base Clock (Sensor Driving Pulse Frequency Divisor) Setting
Random pulse mode: When CTSUCRAH.SDPSEL = 0 The operating clock is divided to generate a base clock to be the source of sensor driving pulse. These bits are also available for setting the voltage stabilization time of the CTSU. When the setting of these bits is n, the base clock is obtained by frequency-dividing the operating clock by 2 (n+1).								
0	0	0	0	0	0	0	0	Operating clock divided by 2 ^{Note 1}
0	0	0	0	0	0	0	1	Operating clock divided by 4
0	0	0	0	0	0	1	0	Operating clock divided by 6
0	0	0	0	0	0	1	1	Operating clock divided by 8
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	0	Operating clock divided by 510
1	1	1	1	1	1	1	1	Operating clock divided by 512
High-resolution pulse mode: When CTSUCRAH.SDPSEL = 1 The SUCLK clock is divided to generate a sensor driving pulse. The frequency of SUCLK is determined by the following equation. SUCLK = STCLK × SUCLK multiplication rate set by the CTSUSUCLKx.SUMULTIx[7:0] bits When the setting of these bits is n, the sensor driving pulse is obtained by frequency-dividing the SUCLK clock by 2 (n+1).								
0	0	0	0	0	0	0	0	SUCLK divided by 2
0	0	0	0	0	0	0	1	SUCLK divided by 4
0	0	0	0	0	0	1	0	SUCLK divided by 6
0	0	0	0	0	0	1	1	SUCLK divided by 8
:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	1	0	SUCLK divided by 510
1	1	1	1	1	1	1	1	SUCLK divided by 512

SSDIV[3:0] ^{Note 2}				Spectrum Diffusion Sampling Cycle Control
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 spread spectrum. Set the clock division ratio for resampling the sensor driving pulse (synchronized with fCLK) with SUCLK. Set these bits so that the sampling cycle is less than 1/4 of the sensor driving pulse cycle. Set these bits for the pin to be measured next after an INTCTSUWR interrupt is generated. These bits are only valid when CTSUCRAH.SDPSEL (random pulse mode) is 0.

Figure 35 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (3/3)

SNUM[7:0]	Measurement Count Setting
- In random pulse mode (CTSUCRAH.SDPSEL = 0) These bits set the measurement count. The SNUM[7:0] bits specify how many times the measurement pulse (basic measurement unit) determined by the CTSUCRBL.PRRATIO[3:0] and CTSUCRBL.PRMODE[1:0] bits is repeated during measurement. Measurement count = (SNUM[5:0] + 1) × 2 Set the SNUM[7: 6] bits to 00B. - In high-resolution pulse mode (CTSUCRAH.SDPSEL = 1) These bits set the measurement time based on the STCLK cycle. Measurement time = 8 × (SNUM[7:0] + 1) × STCLK cycle Set these bits for the pin to be measured next after an INTCTSUWR interrupt is generated.	
SO[9:0]	Sensor CCO Offset Adjustment
The SO[9:0] bits adjust the offset of the input current to the sensor CCO. The input current to the sensor CCO, which is generated by the capacitance in the non-touch state, is set as an offset to prevent the sensor counter from overflowing during measurement. Set these bits to specify the input current offset for the pin to be measured next after the INTCTSUWR interrupt is generated.	

Note 1. When the mutual capacitance method is selected (CTSUCRAL.MD1 = 1) or the spread spectrum is disabled (CTSUCRBL.SOFF = 1), setting SDPA[7:0] = 00000000B is prohibited.

Note 2. The SSDIV[3:0] bits are only valid when the CTSUCRAH.SDPSEL bit is 0 (random pulse mode).

35.2.10 CTSU sensor counter registers (CTSUSC, CTSUUC)

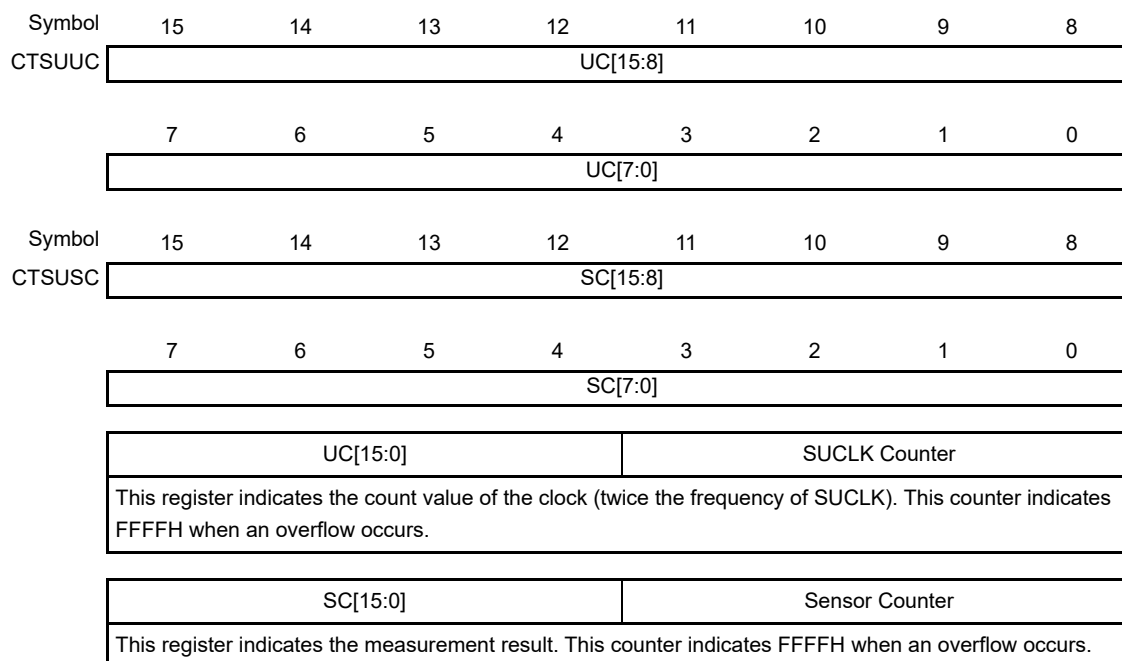
These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 35 - 15 Format of CTSU Sensor Counter Registers (CTSUSC, CTSUUC)

Address: F0524H, F0525H (CTSUSC), F0526H, F0527H (CTSUUC)

After reset: 0000H

R/W: R



Read these registers when an INTCTSURD interrupt is generated. Reading these registers the number of times specified by the CTSUDBGR0.CNTRDSEL bit causes measurement to transition to state 0 or 2. The register value is cleared immediately before transition to state 4 (CTSUSRL.STC[2:0] = 100B) in the next measurement. This register is also cleared by writing 1 to the CTSUCRAL.INIT bit.

- When CTSUDBGR0.CNTRDSEL = 0

Reading the CTSUSC register or CTSUUC register once causes measurement to transition to the next state and both registers are cleared before transition to state 4. If the user require the measurement result, be sure to read the CTSUSC register.

- When CTSUDBGR0.CNTRDSEL = 1

Reading the CTSUSC or CTSUUC register twice causes measurement to transition to the next state. Use this setting if the user require the results of both the CTSUSC and CTSUUC registers by 16-bit access.

35.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (1/4)

Address: F0528H, F0529H (CTSUDBGR0), F052AH, F052BH (CTSUDBGR1)

After reset: 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	IOCSSEL	DCOFF	0	IOC	CNTRDSEL
	7	6	5	4	3	2	1	0
	TSOC	SUCLKEN	CLKSEL0[1:0]	DRV	TSOD	0	0	0
TXREV	Transmission Pin Inverted Output							
0	The pulse output from the transmission pin is not inverted.							
1	The pulse output from the transmission pin is inverted.							
The TXREV bit is used to invert the output from the transmission pin. • In the self-capacitance method (CTSUCRAL.MD1 = 0) 0: A signal of the same phase as the reception pin is output. 1: A signal in reverse phase to the reception pin is output • In the mutual capacitance method (CTSUCRAL.MD1 = 1) 0: First time: A signal of the same phase as the reception pin is output, Second time: A signal in reverse phase to the reception pin is output 1: First time: A signal in reverse phase to the reception pin is output; Second time: A signal of the same phase as the reception pin is output Set this bit to 0 for normal measurement (the setting of 1 is intended for use in debugging).								
CCOCALIB	CCO Calibration Mode Select							
0	Normal mode							
1	Oscillator calibration mode The oscillator calibration mode is used when the external current is compared with the output of the current offset DAC to correct the oscillator characteristics by software.							
CCOCLK	CCO Modulation Circuit Clock Select							
0	Operating clock (specified by the CTSUCRAL.CLK[1:0] bits)							
1	SUCLK							

Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (2/4)

DACCLK	DAC Modulation Circuit Clock Select
0	Operating clock (specified by the CTSUCRAL.CLK[1:0] bits)
1	SUCLK

SUCARRY	CCO Carry Input
0	Normal operation
1	Adds 32 to input data.

The SUCARRY bit is used in the input current test mode of the current oscillator. Set this bit to 0 when the electrostatic capacitance is to be measured.

SUMSEL	Current Oscillator Input Current Matrix Test
0	Normal operation
1	Current oscillator input current test mode

The SUMSEL bit is used in the input current test mode of the current oscillator. Set this bit to 0 when the electrostatic capacitance is to be measured.

DACCARRY	DAC Upper Current Source Carry Input
0	Normal operation
1	Adds 64 to input data.

The DACCARRY bit is used in the input current test mode of the current offset DAC. Set this bit to 0 when the electrostatic capacitance is to be measured.

DACMSEL	Current Offset DAC Current Matrix Test
0	Normal operation
1	Current offset DAC current test mode

The DACMSEL bit controls the current for the current offset DAC. Set this bit to 0 when the electrostatic capacitance is to be measured.

IOCSEL	TSm Pins Fixed Output Select (m = 00 to 35)
0	Sensor driving pulses are output.
1	The fixed-level signal selected by the IOC bit is output.

The IOCSEL bit is used for CTSU calibration.
Setting this bit to 1 while the TSOD bit is set to 1 allows the output of a sensor driving pulse from the TSm pin. This applies to the following pins.

- Pins for which the measurement is enabled by the corresponding bit (CHACm) in the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers.
- Pins assigned as reception by the corresponding bit (CHTRCm) in the CTSUCHTRCAL, CTSUCHTRCAH, and CTSUCHTRCBL registers.

This bit is used with the multiple electrode connection (MEC). Setting this bit to 1 while the TSOD bit is set to 1 allows the output of the signal level specified by the IOC bit from all the TSm pins.

DCOFF	Down-Conversion Disable
0	Normal operation mode
1	Down-conversion disabled

The DCOFF bit disables the output from the LDO. Set this bit to 0 when the electrostatic capacitance is to be measured.

Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (3/4)

IOC	Transmission 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. This bit is ignored when the TSOD and SUCRAL.DCMODE bits are 0.		

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.	
Set this bit to 1 to read both the CTSUSC and the CTSUUC registers.		

TSOC	Calibration Setting 2	
0	Electrostatic capacitance measurement mode	
1	Calibration setting 2	
The TSOC bit is used to set calibration of the CTSU. Set this bit to 0 when the electrostatic capacitance is to be measured.		

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 4
1	0	Setting prohibited
1	1	SUCLK divided by 4
The CLKSEL0[1:0] bits select the clock to be observed from the two clocks generated within the CTSU.		

DRV	Calibration Setting 1	
0	Electrostatic capacitance measurement mode	
1	Calibration setting 1	
The DRV bit is used to set calibration of the CTSU. Set this bit to 0 when the electrostatic capacitance is to be measured. Setting this bit to 1 causes measurement to transition to pseudo state 3 or 4, in which operations differ in the following points from those in states 3 and 4 in normal measurement. • The sensor driving pulses are not output. • The measurement counter is not enabled, and the reset signal is not output.		

Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (4/4)

TSOD	Output Control of All TSm Pins	
0	Electrostatic capacitance measurement mode	
1	The signals selected by the IOCSEL bit are output from TSm pins for the multiple electrode connection (MEC) or for calibration.	

This bit is used for the MEC function or to set calibration of the CTSU. The signals specified by the IOCSEL bit are output from TSm pins when the setting of this bit is 1.

<Mode selection for measurement pins>

TSOD	IOCSEL	Measurement Pin Mode Select
0	0	Electrostatic capacitance measurement mode
0	1	—
1	0	When the MEC function is selected, sensor driving pulses are output from all the TSm pins to be measured. When using the MEC function, set the CTSUCRAL.MD0 bit to 0 to select single scan mode.
1	1	The level specified by the IOC bit is output from TSm pins.

35.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 35 - 17 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

R/W: R/W

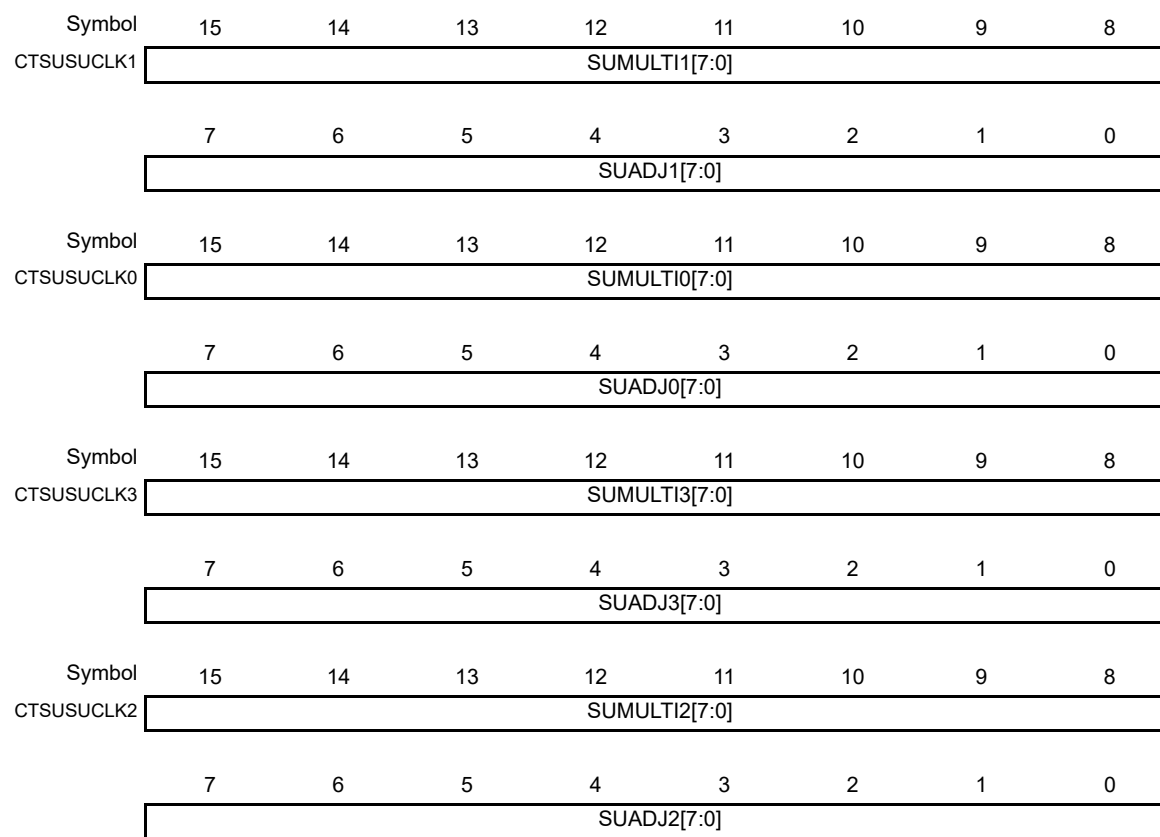


Figure 35 - 17 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH
(CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) (2/2)

SUMULTIx[7:0]	STCLK Multiplication Rate Setting							
The SUMULTIx[7:0] bits set the multiplication rate for generating SUCLK from STCLK (which is taken to be at 0.5 MHz, obtained by frequency-dividing fCLK). The SUADJx[7:0] bits are updated in accord with SUCLK generated by this setting. The frequency of the SUCLK must be set in the range between 16 MHz and 30 MHz. The frequency of SUCLK is determined by the following equation. SUCLK = STCLK × SUCLK multiplication rate set by the CTSUSUCLKx.SUMULTIx[7:0] bits								
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

SUADJx[7:0]	SUCLK Frequency Adjustment							
The SUADJx[7:0] bits set the initial value of the SUCLK frequency. The clock recovery function corrects SUCLK for clock drift, resulting in updating of this register value. Do not modify the updated value.								

35.2.13 CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUTRIM0 register can respectively be set as the DACTRIM and RTRIM registers by an 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUTRIM1 register can respectively be set as the TRESULT4 and SUADJD registers by an 8-bit memory manipulation instruction.

Figure 35 - 18 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1)

Address: F0600H, F0601H (CTSUTRIM0), F0602H, F0603H (CTSUTRIM1)

After reset: **Note**

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 hold the coefficient of variation for the 120-kΩ reference load resistance. These bits hold the initial value set as shipped. Do not modify these bits.								
SUADJD[7:0]		SUCLK Frequency Adjustment						
The SUADJD[7:0] bits are used to adjust the SUCLK frequency in the random pulse mode (CTSUCRAH.SDPSEL = 0). These bits hold the initial value set as shipped. Do not modify these bits.								
DACTRIM[7:0]		Offset Current Adjustment						
The DACTRIM[7:0] bits are used to adjust the DAC coefficient for offset current. These bits hold the initial value as shipped. Do not modify this value.								
RTRIM[7:0]		Reference Resistance Adjustment						
The RTRIM[7:0] bits are used to adjust the value of the internal reference resistor. These bits hold the initial value set as shipped. Do not modify this value.								

Note The values of these registers following a reset are those adjusted as shipped.

35.2.14 CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

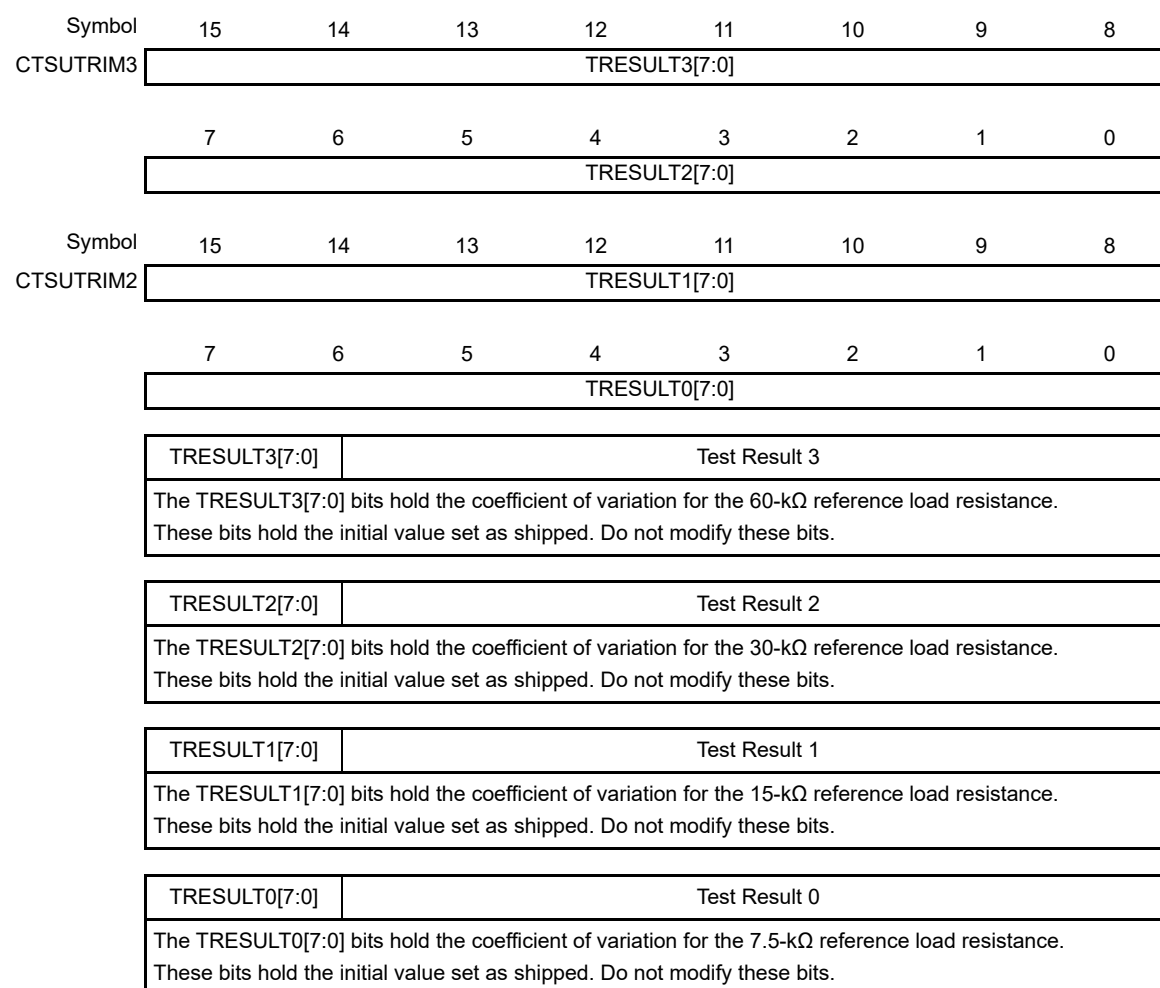
- The higher- and lower-order bits of the CTSUTRIM2 register can respectively be set as the TRESULT1 and TRESULT0 registers by an 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUTRIM3 register can respectively be set as the TRESULT3 and TRESULT2 registers by an 8-bit memory manipulation instruction.

Figure 35 - 19 Format of CTSU Trimming Registers BL and BH (CTSUTRIM2, CTSUTRIM3)

Address: F0604H, F0605H (CTSUTRIM2), F0606H, F0607H (CTSUTRIM3)

After reset: **Note**

R/W: R/W



Note The values of these registers following a reset are those adjusted as shipped.

35.2.15 Registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensor

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensor.

- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- LCD port function registers (PFSEGx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.16 LCD port function registers (PFSEGx)**

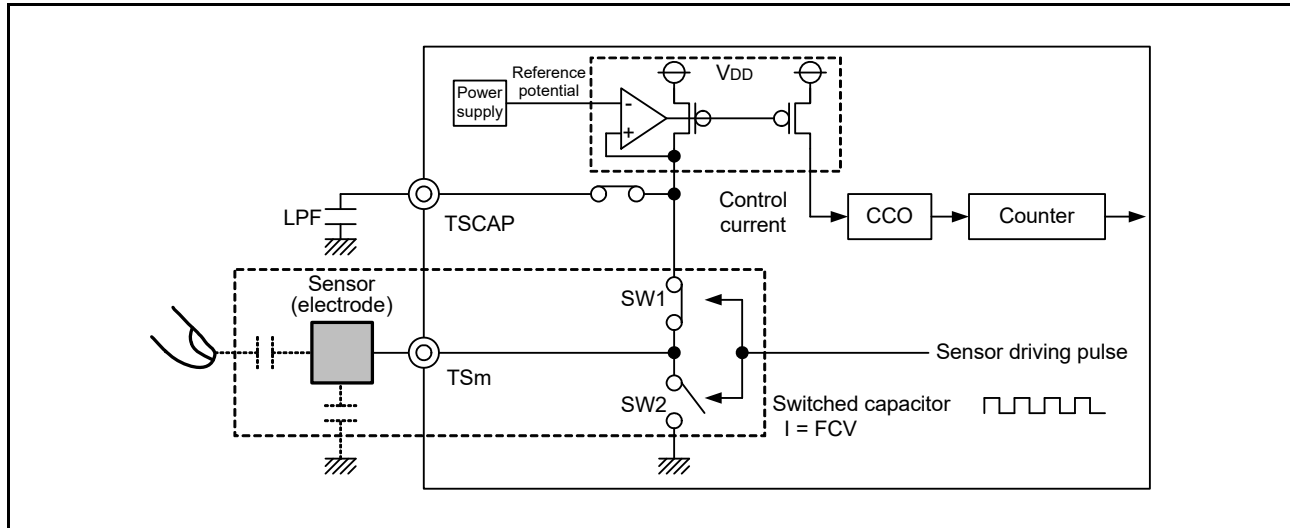
For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and alternate functions**.

35.3 Operation

35.3.1 Principles of measurement

Figure 35 - 20 shows the measurement circuit.

Figure 35 - 20 Measurement Circuit



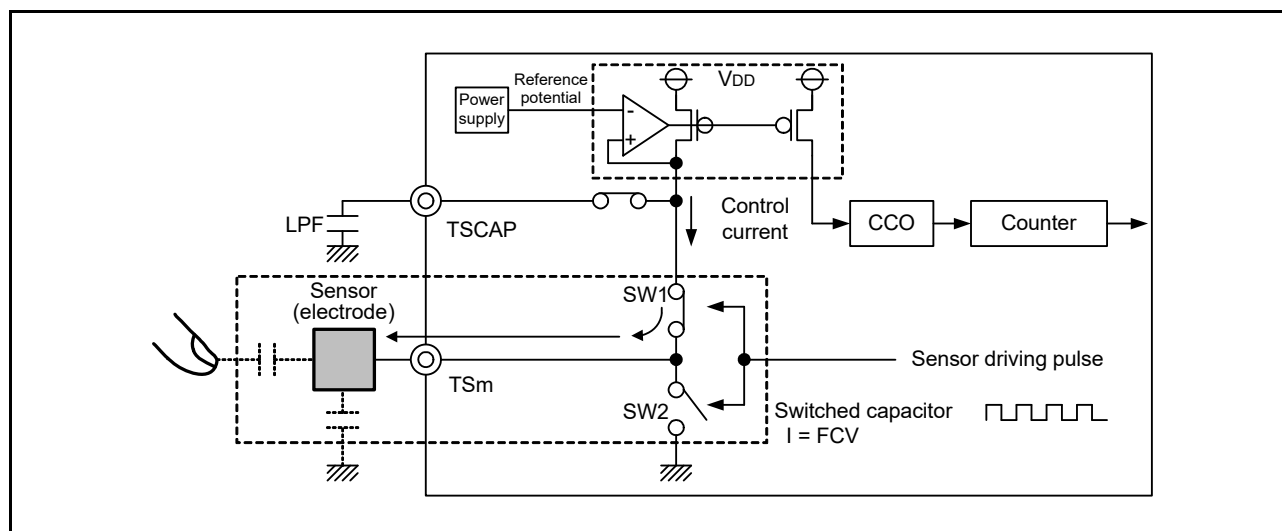
Remark m = 00 to 35

Figure 35 - 21 to **Figure 35 - 23** explain the principle of the electrostatic capacitance measurement of the current-to-frequency conversion method for the CTSU.

1. The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (**Figure 35 - 21**).
2. The charged capacitance is discharged by turning SW1 off and SW2 on (**Figure 35 - 22**).

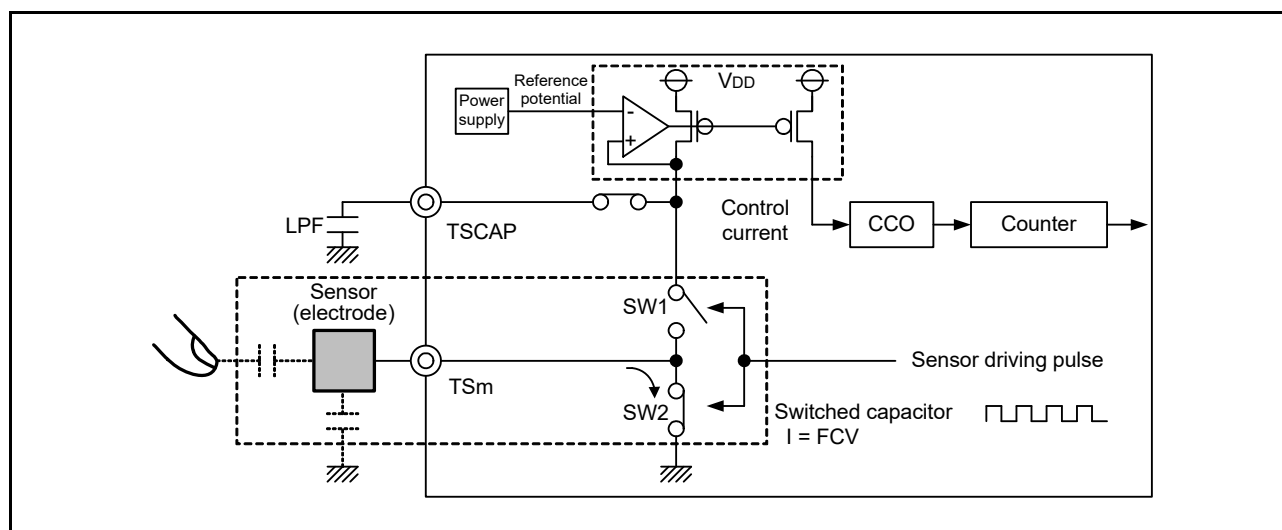
Current flows through the switched capacitor by rapidly switching between charging and discharging in steps 1. and 2. At this time, the value of electrostatic capacitance varies depending on whether the human body (e.g., a finger) is in close proximity, so the flowing current changes. A clock is generated by supplying a control current, which is proportional to the amount of the current flowing through the switched capacitor, from the power supply (LDO) that generates the TSCAP power supply to the current controlled oscillator (CCO). The counter is used to measure the clock frequency which changes depending on whether the human body is in close proximity, and the value read from the counter is used by software to determine whether the human body is in close proximity (**Figure 35 - 23**).

Figure 35 - 21 Charging



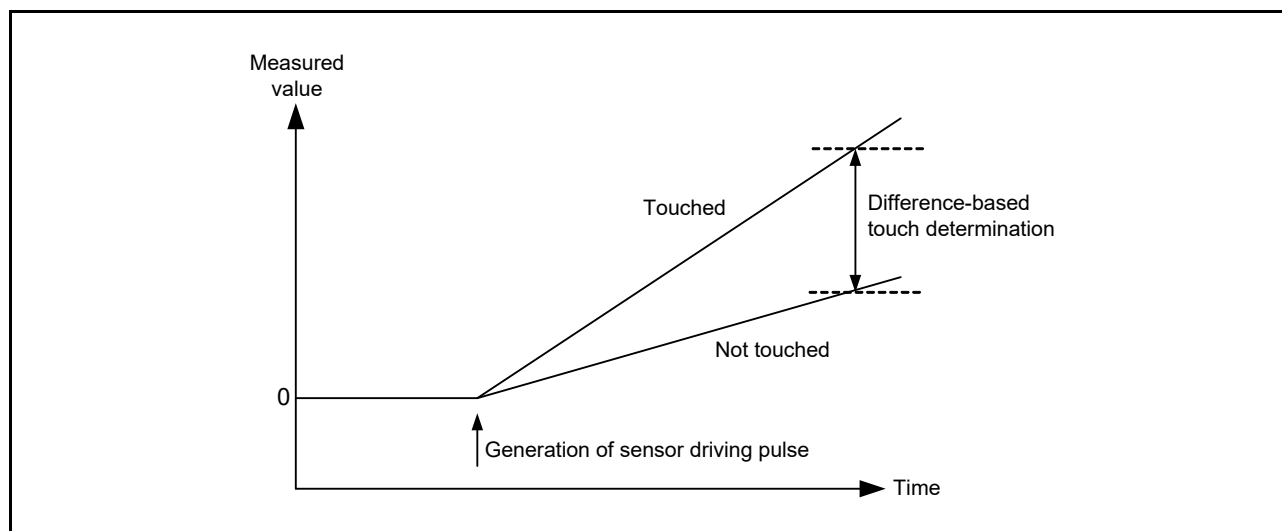
Remark m = 00 to 35

Figure 35 - 22 Discharging



Remark m = 00 to 35

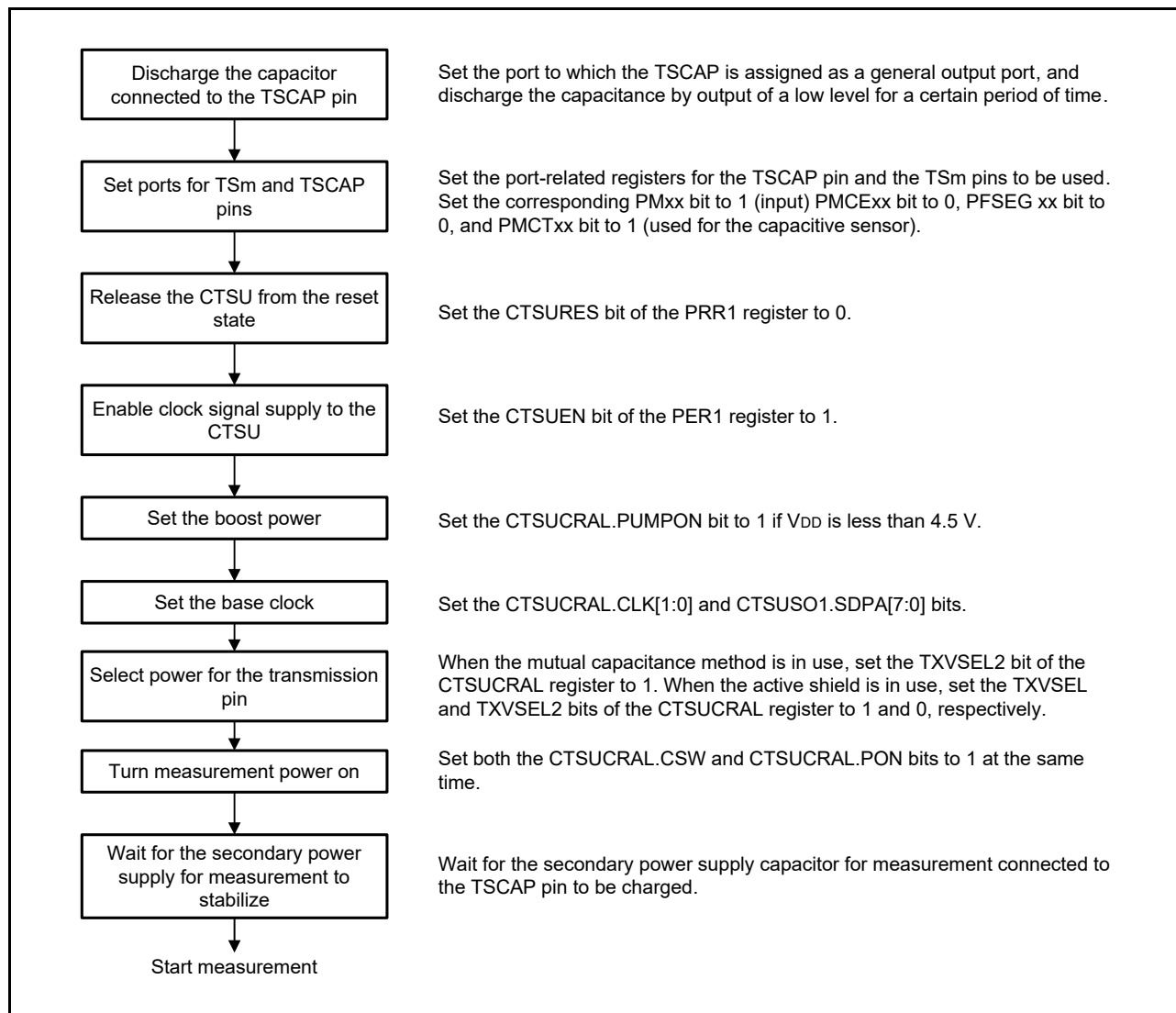
Figure 35 - 23 Change in Measured Value When the Electrode Is Touched or Not Touched



35.3.2 Initial setting procedure

Figure 35 - 24 shows the initial setting procedure for the CTSU.

Figure 35 - 24 CTSU Initial Setting Procedure



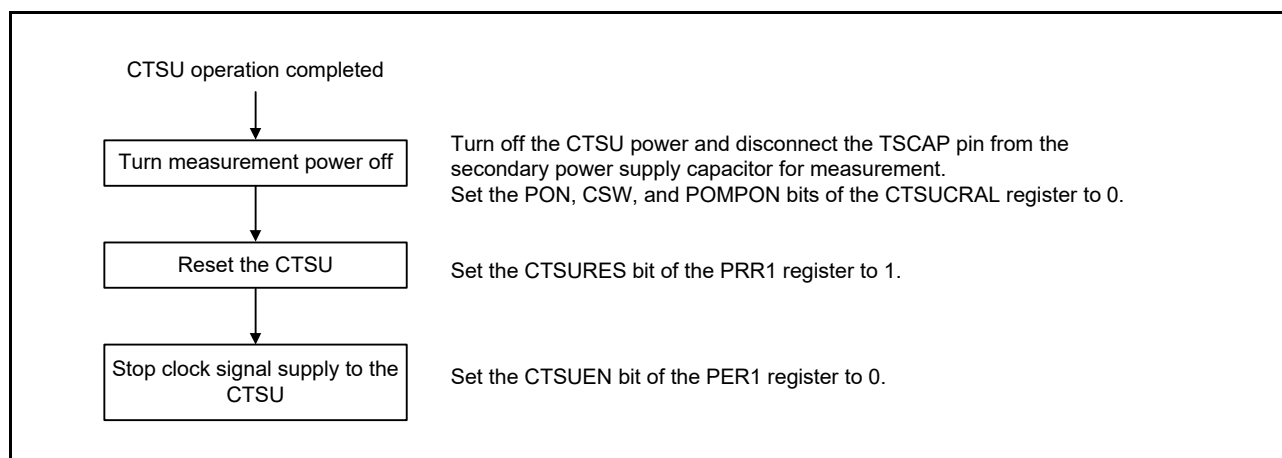
Remark 1. m = 00 to 35

Remark 2. xx = 0, 3, 5, 7, 9, 12

Note that PMCE3, PMCE5, PMCE9, PMCE12, PFSEG9, and PFSEG12 are not present in the RL78/L23 products.

Figure 35 - 25 shows the flow for stopping CTSU operation and placing it in the standby state.

Figure 35 - 25 CTSU Stopping Flowchart

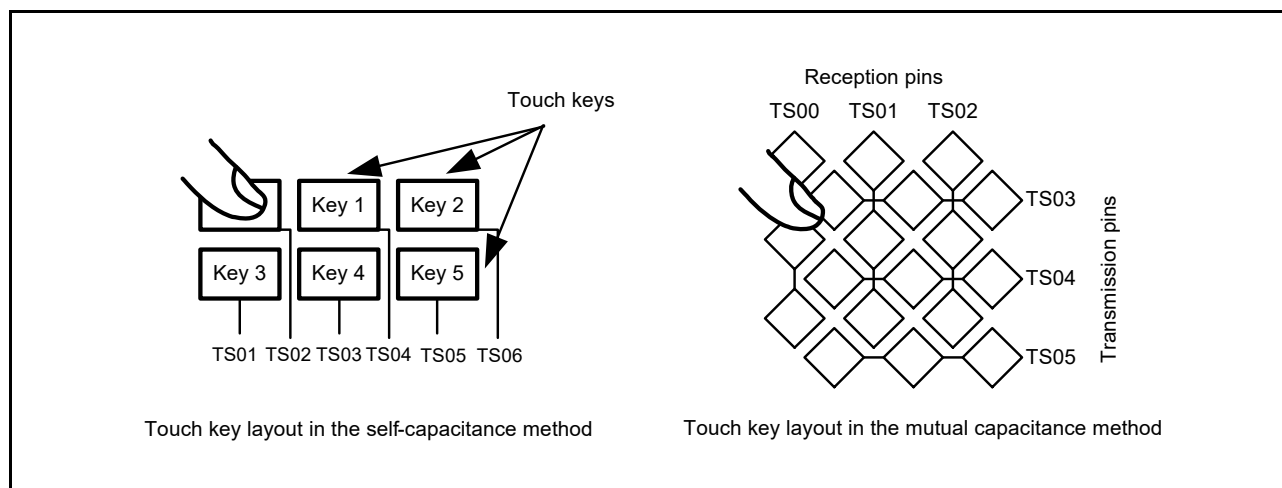


To restart operation after it has been stopped, follow the procedure of the initial settings for the CTSU shown in **Figure 35 - 24**.

35.3.3 Measurement methods

The CTSU supports self-capacitance and mutual capacitance methods. **Figure 35 - 26** illustrates these methods.

Figure 35 - 26 Overview of the Self-capacitance Method and the Mutual Capacitance Method



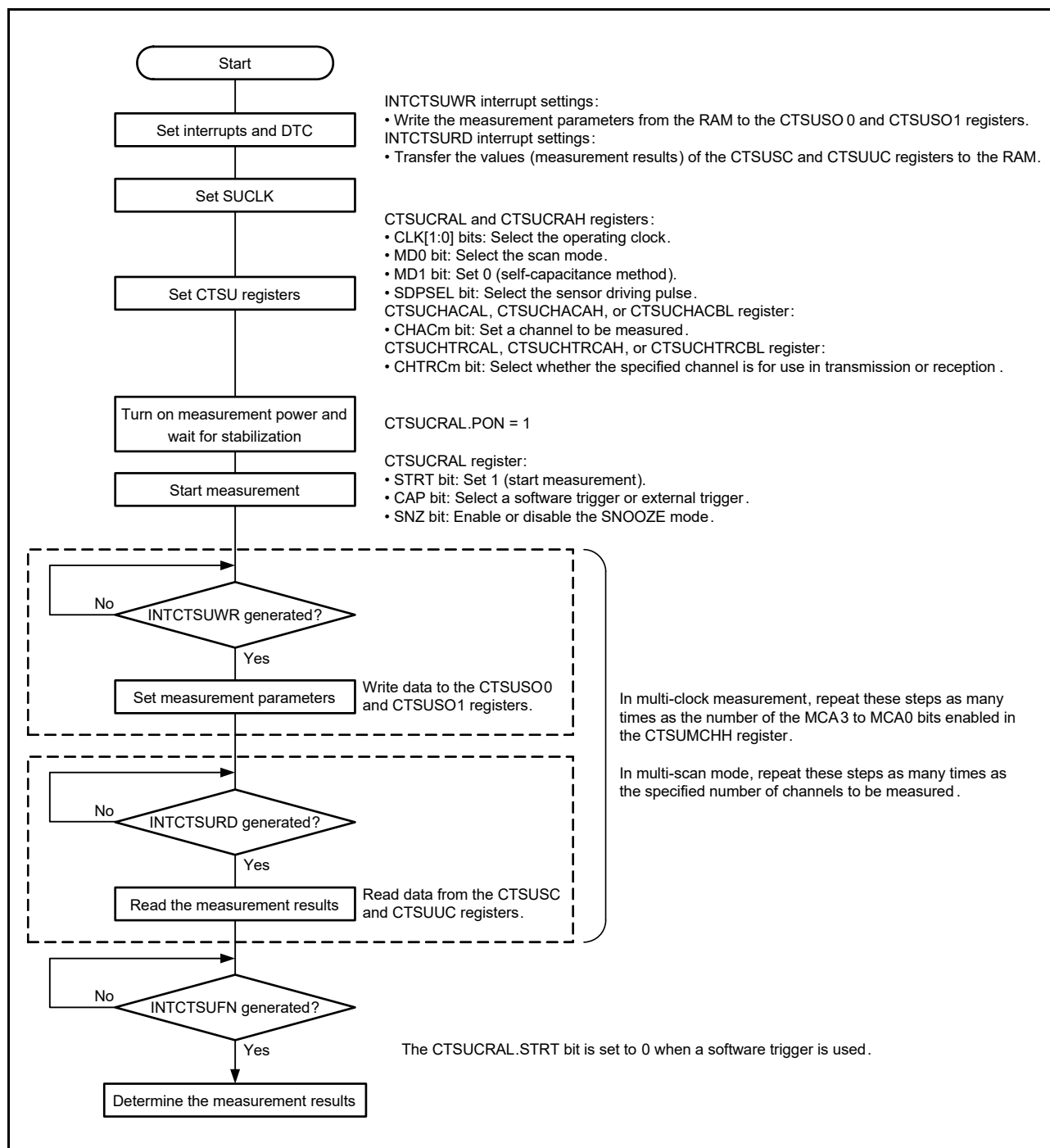
In the self-capacitance method, a single touch pin (TSM) is allocated to a single touch key for the measurement of the individual electrostatic capacitance when the human body (e.g., a finger) is in close proximity.

In the mutual capacitance method, the capacitance between two facing electrodes (transmission and reception pins) is measured.

1. Operation in the self-capacitance method

Figure 35 - 27 shows the control procedure in the self-capacitance method.

Figure 35 - 27 Control Procedure in the Self-capacitance Method



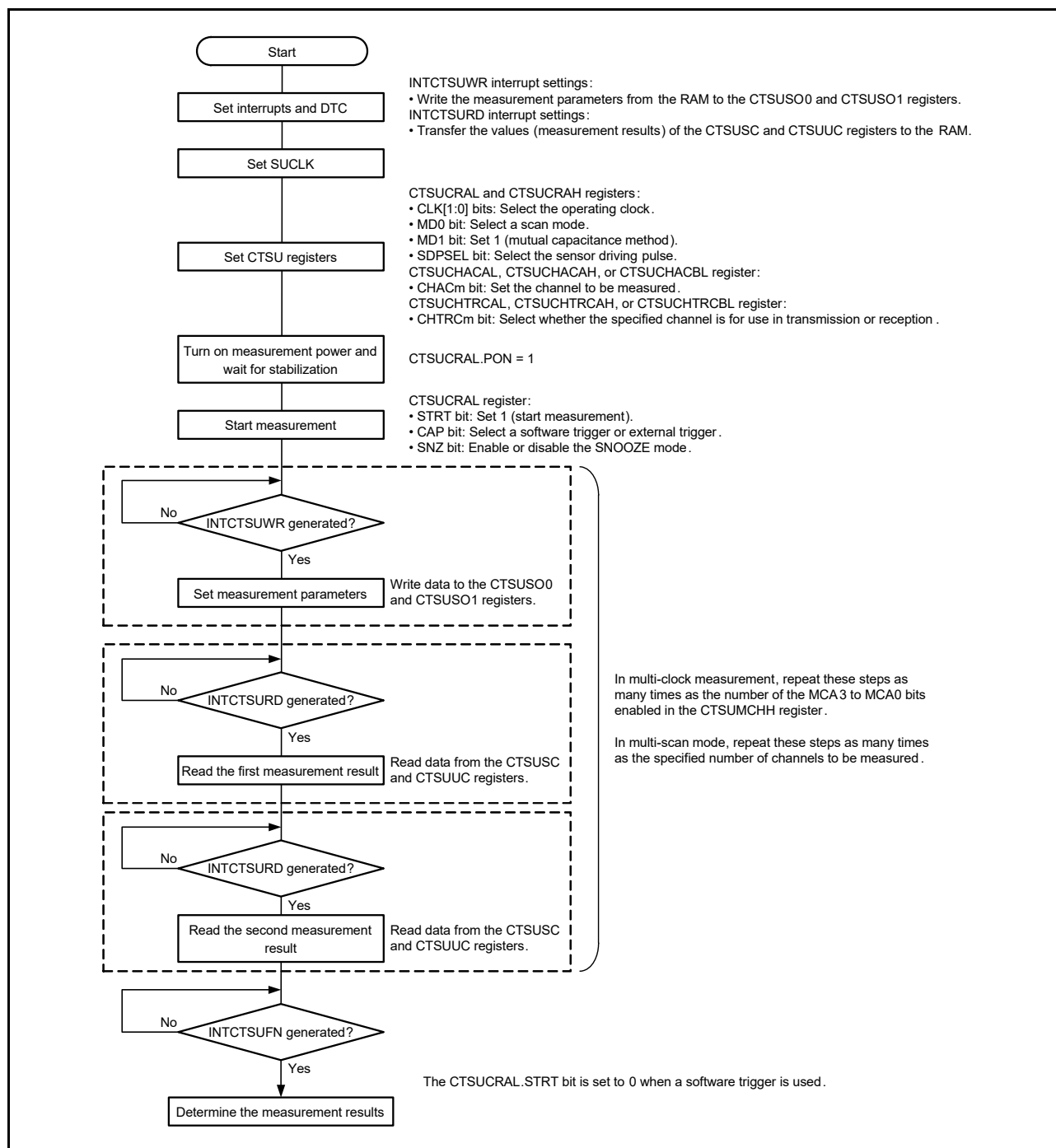
Remark m = 00 to 35

2. Operation in the mutual capacitance method

In the mutual capacitance method, measurement proceeds by applying edges to the target transmit channel to be measured during the high-level period of the sensor driving pulse on the reception channel. A single measurement target is measured twice, on the rising and falling edges. The difference between the data of these two measurements determines whether or not the electrode was touched. The electrostatic capacitance is measured for all combinations of the set transmission and reception pins.

Figure 35 - 28 shows the control procedure in the mutual capacitance method.

Figure 35 - 28 Control Procedure in the Mutual Capacitance Method



Remark m = 00 to 35

35.3.4 Scan modes

The CTSU has two scan modes: single scan mode and multi-scan mode. Use the CTSUCRAL.MD0 bits to select a scan mode.

1. Single scan mode

A single measurement proceeds by using the combination of the reception and transmission channels respectively specified by the MCH0[5:0] and MCH1[5:0] bits of the CTSUMCHL register.

2. Multi-scan mode

Measurement proceeds once for each combination of the reception and transmission channels, respectively selected by the CTSUCHTRCAL, CTSUCHTRCAH, and CTSUCHTRCBL registers, among the channels to be measured specified by the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers. For both the reception and transmission channels, the channel with the lowest channel number is selected first. When the measurement for all combinations of the reception channels for one transmission channel is completed, the transmit channel will switch.

<R>

35.3.5 Multi-clock measurement

Measurements sequentially proceeds switching multiple clock signals with different frequencies. This is effective in the high-resolution pulse mode (CTSUCRAH.SDPSEL = 1).

The clock used for measurement (SUCKLx) should be specified using the MCAx bits (x = 0 to 3) of the CTSUMCHH register. Also, set the frequency of each clock in the CTSUSUCKLx register.

Measurement proceeds in ascending order, starting from SUCLK0. On completion of the measurement using all the specified clock signals, the measurement in the next channel proceeds.

35.3.6 Multiple electrode connection

The multiple electrode connection (MEC) handles multiple electrodes as a single electrode. This function allows multiple electrodes to be measured at the same time.

When CTSUCRAL.MD1 = 0 (self-capacitance method), CTSUCRAL.MD0 = 0 (single scan mode), CTSUDBGR0.TSOD = 1 and CTSUDBGR0.IOCSEL = 0 (multiple electrode connection function) are set, sensor driving pulses are output from all TSm pins for which the CHACm bit of the CTSUCHACAL, CTSUCHACAH, or CTSUCHACBL register is set to 1 (measurement enabled) and the CHTRCm bit of the CTSUCHTRCAL, CTSUCHTRCAH, or CTSUCHTRCBL register is set to 0 (reception). Using this function enables the measurement of the total electrostatic capacitance of all the electrodes connected to the selected TSm pin.

When this function is in use to detect a touch, it is not possible to determine which electrode that has been enabled for measurement has changed in electrostatic capacitance. Set both the TSOD and IOCSEL bits to 0 (electrostatic capacitance measurement mode) and measure the electrostatic capacitance of each electrode to identify the electrode that has been determined to be touched.

<R>

Remark m = 00 to 35

35.3.7 Interrupts

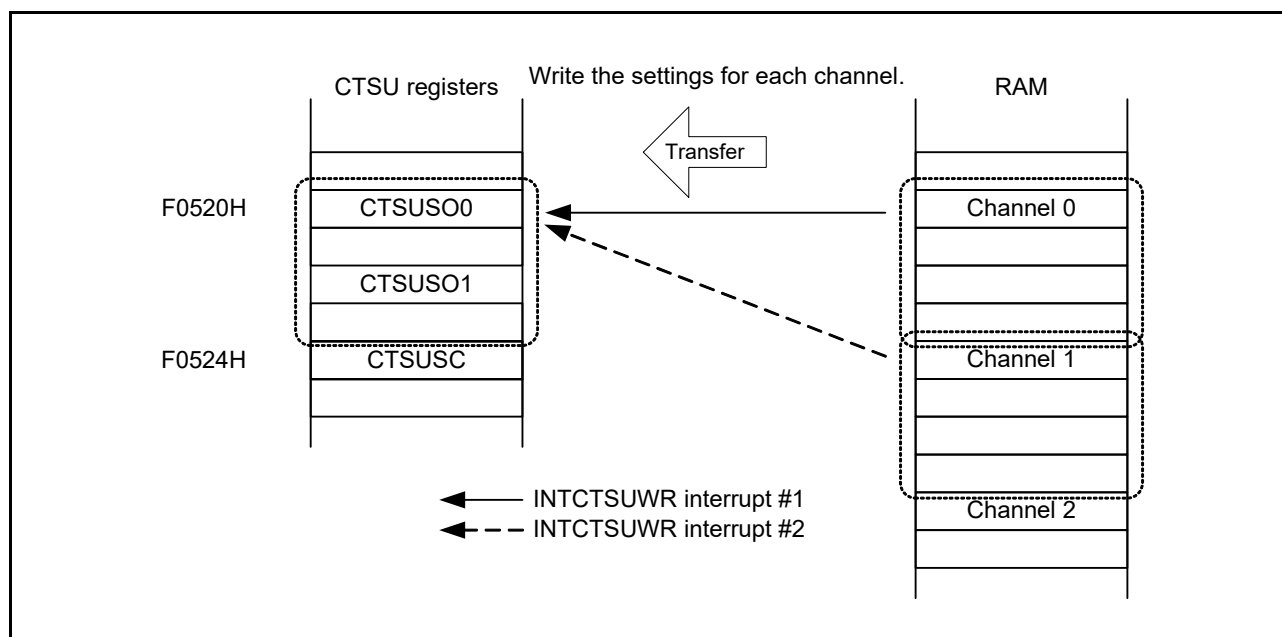
There are three types of interrupts for the CTSU:

- Request to write to a configuration register of an individual CTSU channel (INTCTSUWR)
- Request to transfer data measured by the CTSU (INTCTSURD)
- CTSU measurement end (INTCTSUFN)

1. INTCTSUWR interrupt

Store the settings for each measurement channel in the RAM, and set up the transfer associated with the INTCTSUWR interrupt in advance. An INTCTSUWR interrupt is generated when the measurement transitions from state 1 to 2. Write the setting data to the CTSUSO0 and CTSUSO1 registers in response to an INTCTSUWR interrupt. Writing data to the CTSUSO1 register causes measurement to transition to state 3.

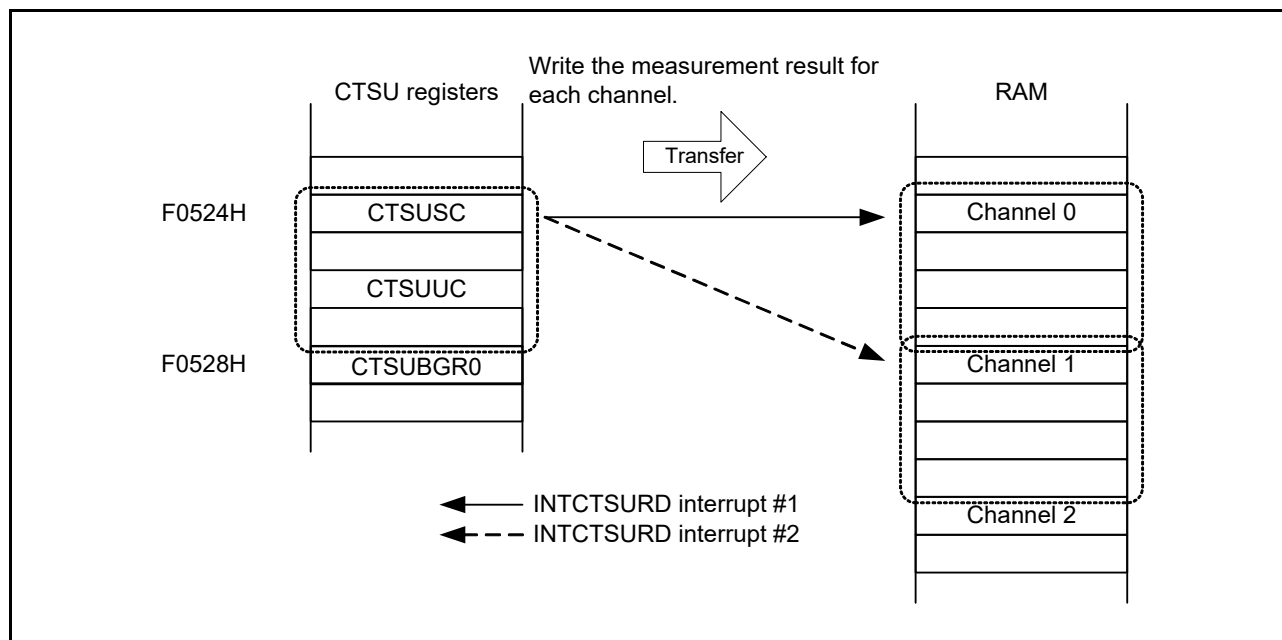
Figure 35 - 29 Data Transfer on an INTCTSUWR Interrupt



2. INTCTSURD interrupt

An INTCTSURD interrupt is generated when the measurement of one channel is finished and the measurement transitions from state 5 to 1. Read the measurement result (sensor counter value) from the CTSUSC register in response to this interrupt. If reading the CTSUUC register is required, set the CTSUDBGR0.CNTRDSEL bit to 1 (transitions to the next state after reading twice). Reading the CTSUSC and CTSUUC registers the number of times specified by the CNTRDSEL bit causes measurement to transition to state 0 or 2.

Figure 35 - 30 Data Transfer Operation on an INTCTSURD Interrupt



3. INTCTSUFN interrupt

An INTCTSUFN interrupt is generated when the measurement of all specified channels is completed and the measurement transitions from state 1 to 0.

35.4 Usage Notes on the Capacitive Sensing Unit

35.4.1 Evaluation of detection by the capacitive sensing unit (CTS2La)

The final stages of product development require operating the system under conditions close to those at the time of product shipment to judge the validity of the results of detection by the touch sensor. "QE for Capacitive Touch" (a development assistance tool for capacitive touch sensors) is provided for use in evaluation. Monitor the states in the measurement of capacitance and evaluate them thoroughly. If the expected results are not obtained, use the "QE for Capacitive Touch" tool to adjust and re-evaluate the CapTouch parameters (mainly the touch thresholds).

35.4.2 TSCAP pin

A capacitor must be connected to the TSCAP pin to stabilize the internal voltage of the CTSU. Make sure the wiring between the TSCAP pin and the capacitor, and between the capacitor and GND, is as thick and as short as possible. Before setting the CTSUCRAL.CSW bit to 1 (the external capacitance connection switch turned on), the capacitor connected to the TSCAP pin must output a low level from the port to which the TSCAP function is assigned and be fully discharged.

35.4.3 Changing the clock and port settings during measurement

Do not change or stop the CPU/peripheral hardware clock (fCLK) while measurement is in progress (CTSUCRAL.STRT = 1). Also, do not change the port settings for the measurement pins (TSM and TSCAP). When the settings are to be changed, forcibly stop the CTSU by respectively setting the INIT and START bits in the CTSUCRAL register to 1 and 0 at the same time. For restarting measurement by the CTSU, make the initial settings of the CTSU by using either of the following procedures.

- After writing PRR1.CTSURES = 1 and PER1.CTSUEN = 0, make the initial settings of the CTSU according to **35.3.2 Initial setting procedure**.
- After writing CTSUCRAL.PON, CTSUCRAL.CSW = 00B, and CTSUCRAL.SNZ = 0, make the initial settings of the CTSU according to **35.3.2 Initial setting procedure**.

35.4.4 Software trigger

When 10B (fCLK/4) or 11B (fCLK/8) is selected by the CTSUCRAL.CLK[1:0] bits, for restarting measurement immediately after the measurement is completed, write 1 to the CTSUCRAL.STRT bit after the following time has elapsed since the interrupt is generated.

- When CLK[1:0] = 10B (fCLK/4): At least 3 cycles
- When CLK[1:0] = 11B (fCLK/8): At least 7 cycles

35.4.5 External trigger

If an external trigger (an event from the ELCL) is input during a measurement period, the trigger is ignored. The next external trigger is enabled when one cycle of the operating clock has elapsed after an INTCTSUFN interrupt is generated. To release the CTSU from the external trigger mode (forced stop), respectively set the CTSUCRAL.STRT and CTSUCRAL.INIT bits to 0 and 1 at the same time. Follow the procedure below to use external triggers.

1. Make the initial settings of the CTSU.
2. Select the capacitive sensing unit as the destination of the event link controller (ELCL) while the CTSUCRAL.STRT bit is 0 (measurement stopped).
3. After setting the CTSUCRAL.CAP bit to 1 (external trigger), set the CTSUCRAL.STRT bit to 1 (starting measurement by the CTSU).
4. Start the operation of the peripheral function that generates external trigger sources.

35.4.6 Reading the CTSU sensor counter registers (CTSUSC and CTSUUC)

Do not attempt to read the CTSU sensor counter registers (CTSUSC and CTSUUC) during measurement (states 3 to 5). The value cannot be guaranteed if reading of these registers is attempted.

35.4.7 Forced stop

To forcibly stop measurement, be sure to set the CTSUCRAL.STRT bit to 0 and the CTSUCRAL.INIT bit to 1 at the same time. Setting the INIT bit to 1 initializes the internal measurement state and also initializes the following registers.

- CTSUSC and CTSUUC registers
- CTSUMCHL and CTSUMCHH registers
- All bits in CTSUSRL except bits ICOMP0 and ICOMP1

If operation is forcibly stopped, an interrupt request might be generated depending on the internal state. Disable the interrupts and stop the DTC before forcibly stopping the operation. If stoppage of the DTC transfer is required when the DTC is in use for controlling the CTSU, also forcibly stop the CTSU and proceed with the initialization.

35.4.8 Sampling cycle when the spread spectrum is in use

When the spread spectrum is enabled (CTSUCRBL.SOFF = 0), set the CTSUCRAL.CLK[1:0] and CTSUSO1.SDPA[7:0] bits so that the sampling cycle of the sensor driving pulse is less than 1/4 of the sensor driving pulse cycle.

35.4.9 Settings of the transmission pin when the self-capacitance method is to be used

The transmission pin is not used for measurement when the self-capacitance method is in use. A pulse that is in phase with the measurement pulse is output from the transmission pin. Use the transmission pin as an active shield on the board. However, the active shield cannot be used when CTSUCRAL.ATUNE0 = 1 (measurement power-supply voltage = 1.2 V). Also, when the self-capacitance method is to be used, do not select multiple transmission pins at the same time.

35.4.10 Points to note on noise countermeasures

Fluctuations in the VDD power supply due to noise or other factors may lead to variation in the results of measurement. In addition, the application of ripple noise to the VDD power supply may change (decrease) the control current of the capacitive sensing unit and in turn decrease the measured values of the electrostatic capacitance, depending on the frequency band of the ripple noise. For details, refer to the application note, **Capacitive Touch Ripple Noise Prevention Guide (R30AN0453)**. Take note of these points when designing the power supply circuit.

35.4.11 Memory occupancy

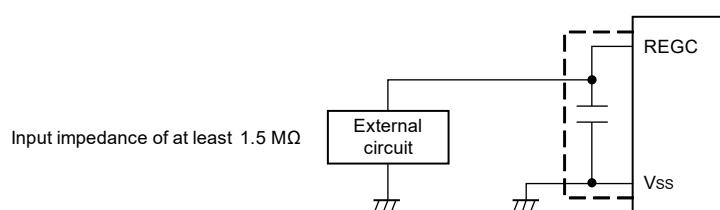
The amount of memory required to run the capacitive sensing unit depends on the configuration of the buttons. For details, refer to the following application notes.

- **RL78 Family CTSU Module Software Integration System (R11AN0484)**
- **RL78 Family TOUCH Module Software Integration System (R11AN0485)**

Section 36 Regulator

36.1 Overview

The RL78/L23 incorporates a circuit for constant voltage operation. To stabilize the output voltage from the regulator, connect the REGC pin to Vss 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.

Section 37 Option Bytes

37.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/L23 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 functions described in **37.1.1** by using the option bytes. For the bits to which no function is allocated, do not change their initial values. To use boot or bank swapping during self-programming, set the same values as the settings in 000C0H to 000C3H in the bytes of the address range indicated below of the area to be used in swapping.

Caution The option bytes should always be set regardless of whether each function is used.

Table 37 - 1 List of Addresses in the Areas to be Used in Swapping

List of Addresses in the Areas to be Used in Swapping	User option bytes			On-chip Debugging Option Byte
Option bytes area	000C0H	000C1H	000C2H	000C3H
Bank swapping is in use (R7F100LxL (x = F, G, J, L, M, P))	400C0H	400C1H	400C2H	400C3H
Bank swapping is in use (R7F100LxJ (x = F, G, J, L, M, P))	200C0H	200C1H	200C2H	200C3H
Boot swapping (128 Kbytes) is in use	200C0H	200C1H	200C2H	200C3H
Boot swapping (64 Kbytes) is in use	100C0H	100C1H	100C2H	100C3H
Boot swapping (32 Kbytes) is in use	080C0H	080C1H	080C2H	080C3H
Boot swapping (16 Kbytes) is in use	040C0H	040C1H	040C2H	040C3H
Boot swapping (8 Kbytes) is in use	020C0H	020C1H	020C2H	020C3H
Boot swapping (4 Kbytes) is in use	010C0H	010C1H	010C2H	010C3H
Boot swapping (2 Kbytes) is in use	008C0H	008C1H	008C2H	008C3H

37.1.1 User option bytes (000C0H to 000C2H or addresses of the area to be used in swapping)

1. 000C0H or address of the area to be used in swapping
 - 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 or bank swapping is to be used, set the same value as the setting in 000C0H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see Table 37 - 1 List of Addresses in the Areas to be Used in Swapping.

2. 000C1H or address of the area to be used in swapping
 - 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 42.4 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 address of the area to be used in swapping).

Caution 2. When boot or bank swapping is to be used, set the same value as the setting in 000C1H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see Table 37 - 1 List of Addresses in the Areas to be Used in Swapping.

3. 000C2H or address of the area to be used in swapping
 - Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (VDD) 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, 48 MHz, or 64 MHz

Caution When boot or bank swapping is to be used, set the same value as the setting in 000C2H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see Table 37 - 1 List of Addresses in the Areas to be Used in Swapping.

37.1.2 On-chip debug option byte (000C3H or address of the area to be used in swapping)

- 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 or bank swapping is to be used, set the same value as the setting in 000C3H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see Table 37 - 1 List of Addresses in the Areas to be Used in Swapping.

37.2 Format of User Option Bytes

Figure 37 - 1 Format of User Option Byte (000C0H or address of the area to be used in swapping)

Address: 000C0H or address of the area to be used in swapping^{Note 1}

Symbol	7	6	5	4	3	2	1	0
	WDTINT	WINDOW1	WINDOW0	WDTON	WDSCS2	WDSCS1	WDSCS0	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 + 1/4 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)							
WDSCS2	WDSCS1	WDSCS0	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 or bank swapping is to be used, set the same value as the setting in 000C0H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see **Table 37 - 1 List of Addresses in the Areas to be Used in Swapping**.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless of the value of the WINDOW1 and WINDOW0 bits.

Figure 37 - 2 Format of User Option Byte (000C1H or address of the area to be used in swapping)

Address: 000C1H or address of the area to be used in swapping^{Note}

Symbol	7	6	5	4	3	2	1	0
	LVD0EN	LVD0SEL	1	1	1	LVD0V2	LVD0V1	LVD0V0

- LVD0 setting (in the 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.				

- LVD0 setting (in the 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.				

- LVD0 off setting (external reset input from the $\overline{\text{RESET}}$ 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, Cautions, and Remarks continue on the next page.)

Note When boot or bank swapping is to be used, set the same value as the setting in 000C1H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see **Table 37 - 1 List of Addresses in the Areas to be Used in Swapping**.

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 42.4 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 address of the area to be used in swapping).

Remark 1. *: Don't care.

Remark 2. For details on the LVD0 circuit, see **Section 30 Voltage Detector (LVD)**.

Remark 3. The detection voltage is a typical value. For details, see **42.6.6 LVD circuit characteristics**.

Figure 37 - 3 Format of User Option Byte (000C2H or address of the area to be used in swapping)

Address: 000C2H or address of the area to be used in swapping^{Note 1}

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	FRQSEL4	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 Note 2	1 to 2 MHz (Rewriting of flash memory is not possible.)	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 2 MHz	1.6 to 5.5 V
			1 to 4 MHz (Rewriting of flash memory is not possible.)	
1	1	HS (high-speed main) mode	1 to 24 MHz	1.8 to 5.5 V
			1 to 2 MHz	1.6 to 5.5 V
			1 to 4 MHz (Rewriting of flash memory is not possible.)	
			1 to 32 MHz	1.8 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the High-speed On-chip Oscillator Clock	
					fHOCO	fIH
1	1	0	0	0	64 MHz	32 MHz
1	0	0	0	0	48 MHz	24 MHz
0	1	0	0	0	32 MHz	32 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	1 MHz
Other than the above					Setting prohibited	

Note 1. When boot or bank swapping is to be used, set the same value as the setting in 000C2H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see **Table 37 - 1 List of Addresses in the Areas to be Used in Swapping**.

Note 2. In LP mode, only the main on-chip oscillator clock (fIH or fIM) can be selected as the CPU/peripheral hardware clock (fCLK).

(Cautions are listed on the next page.)

- Caution 1.** Be sure to set bit 5 to 1.
- Caution 2.** The operating frequency range and operating voltage range vary depending on the flash operation mode. For details, see 42.4 AC Characteristics.
- Caution 3.** Setting of FRQSEL4 to 1 requires that VDD be no less than 2.2 V.

37.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 37 - 4 Format of On-chip Debug Option Byte (000C3H or address of the area to be used in swapping)

Address: 000C3H or address of the area to be used in swapping^{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 or bank swapping is to be used, set the same value as the setting in 000C3H in the address of the area to be used in swapping. For addresses of the area to be used in swapping, see **Table 37 - 1 List of Addresses in the Areas to be Used in Swapping**.

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).

37.4 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 or bank swapping is to be used during self-programming, set the same values as the settings in 000C0H to 000C3H in the address of the area to be used in swapping as shown below. For addresses of the area to be used in swapping, see **Table 37 - 1 List of Addresses in the Areas to be Used in Swapping**.

OPT2	.CSEG	AT	0x040C0	; Boot swapping is in use (16 Kbytes)
	.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 bytes in assembly language, use OPT_BYTE as the relocation attribute name of the CSEG directive. To specify the address of the option-byte range for use in swapping when boot or bank swapping is to be used, use the relocation attribute AT to specify the absolute address.

Section 38 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.

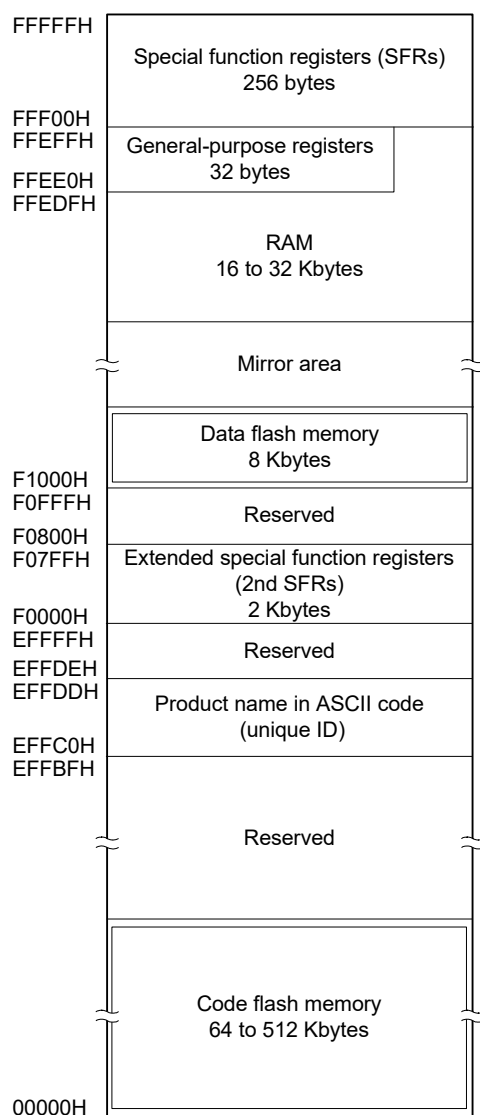


Table 38 - 1 Specifications of the Code Flash Memory and Data Flash Memory

Item	Code Flash Memory	Data Flash Memory
Memory capacity	64 to 512 Kbytes	8 Kbytes
Read cycle	fCLK ≤ 32 MHz (no wait)	fCLK ≤ 32 MHz (1 wait access)
Programming/erase	• Programming or erasure of the contents in the code flash memory and data flash memory by using the control commands specified in the relevant registers • Serial programming through the serial interface by using the dedicated flash memory programmer • Self-programming of the flash memory by using the user program	
Security	• Flash shield window For details, see 38.9 Flash Shield Window. 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 are possible by serial programming or self-programming. For details, see 38.10 Security Settings. - Disabling block erasure - Disabling writing - Disabling changing of the boot cluster size and bank swapping setting (BTBLS) - Disabling rewriting the boot area - Disabling connection to the programmer and on-chip debugger - Enabling programmer connection ID authentication	Not available
Protection	• Flash read protection For details, see 38.6 Self-programming.	Not available
Bank programming	Products with 256 or 512 Kbytes of code flash memory have the bank programming function, which enables updating of the program in one bank while the user program is running from the other bank. For details, see 38.8 Bank Programming. • Products with 256 Kbytes of code flash memory: 128 Kbytes × 2 banks • Products with 512 Kbytes of code flash memory: 256 Kbytes × 2 banks	Not available
Background operation (BGO)	• Instructions can be executed from the code flash memory while rewriting the data flash memory. (Simultaneous use of background operation (BGO) and bank programming in bank programming mode is not possible.)	
Unit of programming/erase	• Unit of programming in the user area: 32 bits • Unit of erasure in the user area: 2 Kbytes	• Unit of programming in the user area: 8 bits • Unit of erasure in the user area: 256 bytes
Other functions	• Boot swapping Boot cluster can be set within the range from 2 to 128 Kbytes. For details, see 38.7 Boot Swapping. • On-chip debugging	Not available
On-board programming	Serial programming using the flash memory programmer • Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer. For details, see 38.1 Serial Programming Using Flash Memory Programmer and 38.12 Data Flash Memory. Serial programming using external device (that Incorporates UART) • Data can be written to the flash memory on-board through UART communications with an external device (a microcontroller or ASIC). For details, see 38.2 Serial Programming Using External Device (that Incorporates UART) and 38.12 Data Flash Memory. Self programming • The user application can execute self-programming of the code flash memory and data flash memory. For details, see 38.6 Self-programming, and 38.12 Data Flash Memory.	

38.1 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 38 - 2 Wiring between RL78/L23 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Number of Pins						
					44	48	52	64	80	100	
Signal Name		I/O	Pin Function	Pin Name	LQFP (10 × 10)	LFQFP (7 × 7) HWQFN (7 × 7)	LQFP (10 × 10)	LFQFP (10 × 10) LQFP (12 × 12)	LFQFP (12 × 12) LQFP (14 × 14)	LFQFP (14 × 14)	LQFP (14 × 20)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator				—	TOOL0	—	2	3	4	5
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	2	3	4	5	9	89	12
SI/RxD	—	I/O									
—	RESET	Output	Reset signal	RESET	3	4	5	6	10	90	13
/RESET	—	Output									
VCC	VDD	I/O	VDD voltage generation/ power monitoring	VDD	11	12	13	14	18	98	21
GND		—	Ground	VSS	10	11	12	13	17	97	20
				EVSS	—	—	—	—	—	15	38
				REGC Note	9	10	11	12	16	96	19
FLMD1	EMVDD	—	Driving power for TOOL0 pin	VDD	11	12	13	14	18	98	21

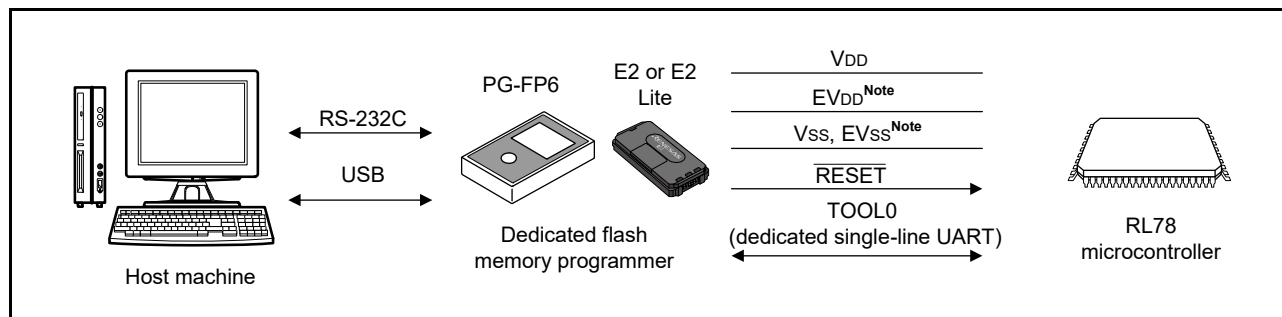
Note Connect the REGC pin to the ground via a capacitor (0.47 to 1 μF).

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.

38.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 38 - 1 Environment for Writing Program to Flash Memory



Note This pin is only present in the 100-pin products.

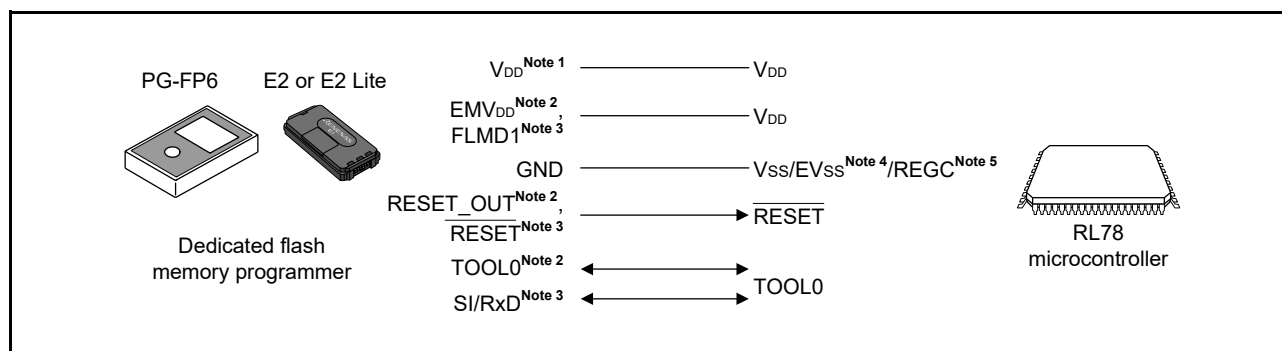
A host machine that controls the dedicated flash memory programmer is necessary. For the interface between the dedicated flash memory programmer and RL78 microcontroller, the TOOL0 pin of the RL78 microcontroller is used for writing and erasure via the dedicated single-line UART.

38.1.2 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 38 - 2 Communications with Dedicated Flash Memory Programmer



Note 1. The signal name for the PG-FP6 is Vcc.

Note 2. This connection applies when an E2 or E2 Lite on-chip debugging emulator is in use.

Note 3. This connection applies when a PG-FP6 is in use.

Note 4. This pin is only present in the 100-pin products.

Note 5. Connect the REGC pin to the 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 38 - 3 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, EVss, 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 38 - 2**.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μF).

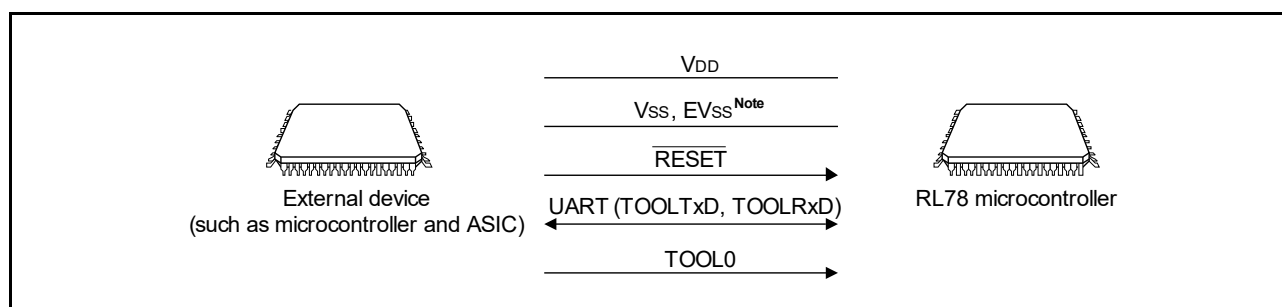
38.2 Serial Programming Using External Device (that Incorporates UART)

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).

38.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 38 - 3 Environment for Writing Program to Flash Memory



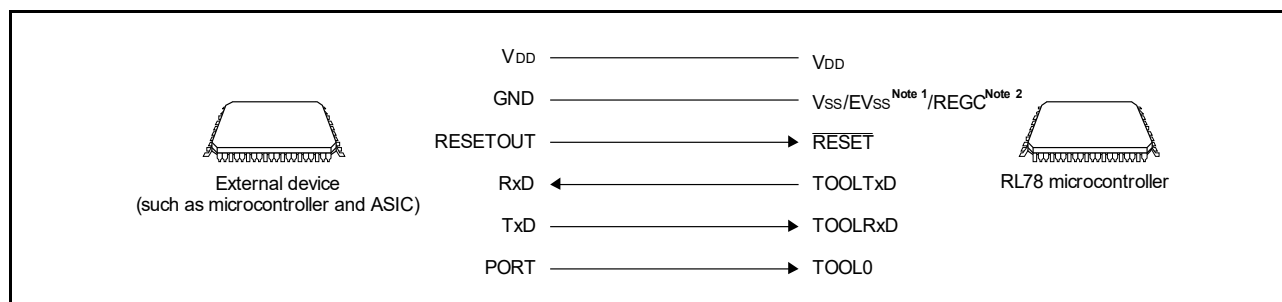
Note This pin is only present in the 100-pin products.

38.2.2 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 38 - 4 Communications with External Device



Note 1. This pin is only present in the 100-pin products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μ F).

The external device generates the following signals for the RL78 microcontroller.

Table 38 - 4 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 1} , EVss ^{Note 2}
RESETOUT	Output	Reset signal output	RESET
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note 1. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μ F).

Note 2. This pin is only present in the 100-pin products.

38.3 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 **38.4.2 Flash memory programming mode**.

38.3.1 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.

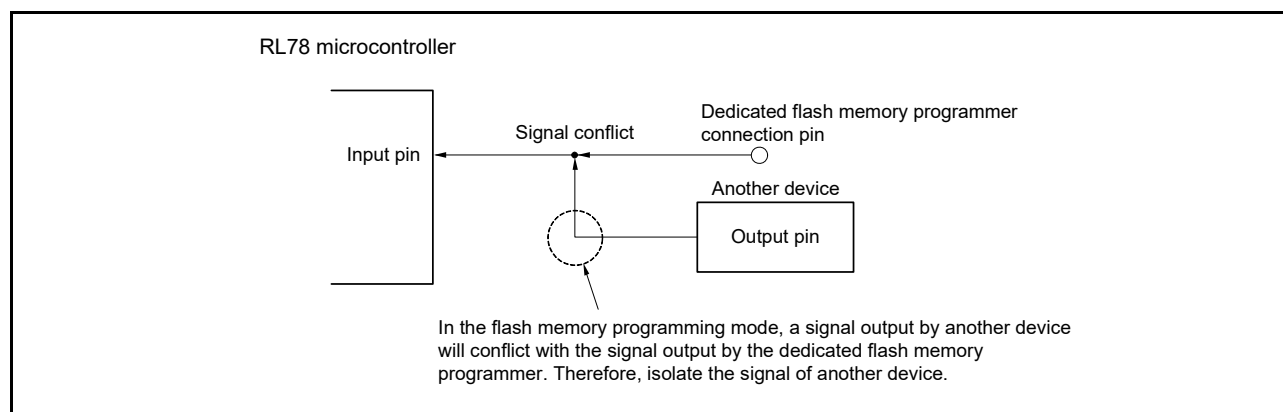
Remark 1. t_{HD} : This is the time over which the TOOL0 pin must be kept at the low level following the end of the external and internal resets for setting of the flash memory programming mode. See **42.10 Timing of Entry to Flash Memory Programming Modes**.

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.

38.3.2 $\overline{\text{RESET}}$ 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 38 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



38.3.3 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.

38.3.4 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.

38.3.5 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.

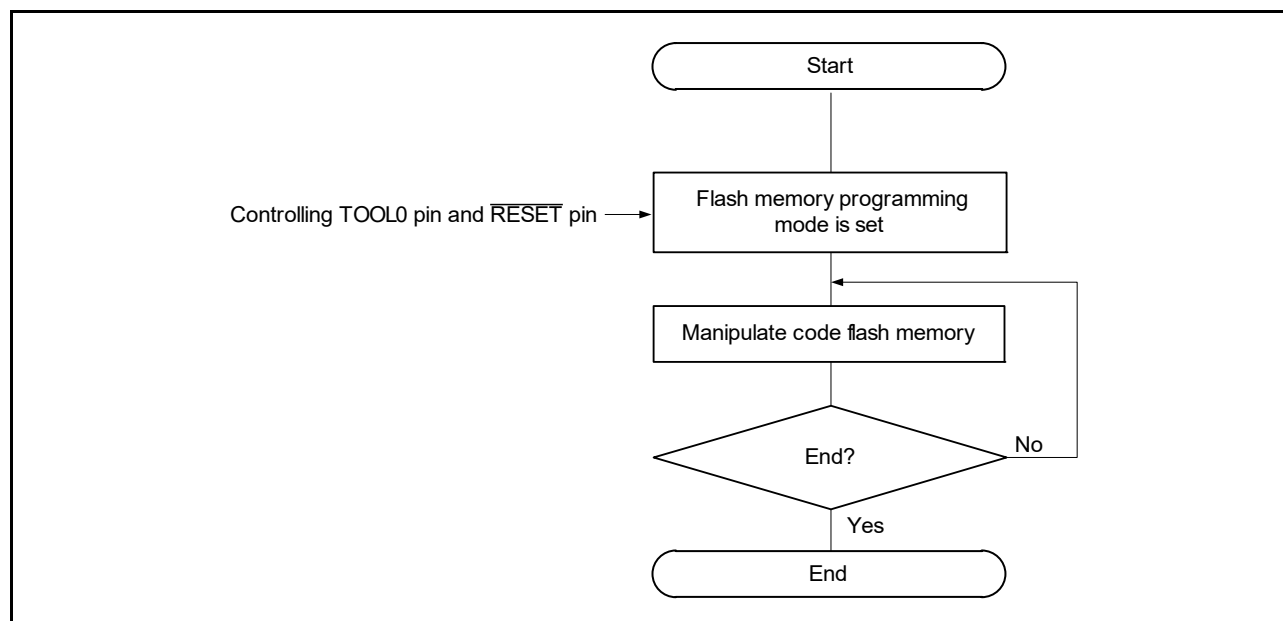
Note The signal name for the PG-FP6 is VCC.

38.4 Programming Method

38.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 38 - 6 Code Flash Memory Manipulation Procedure



38.4.2 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.

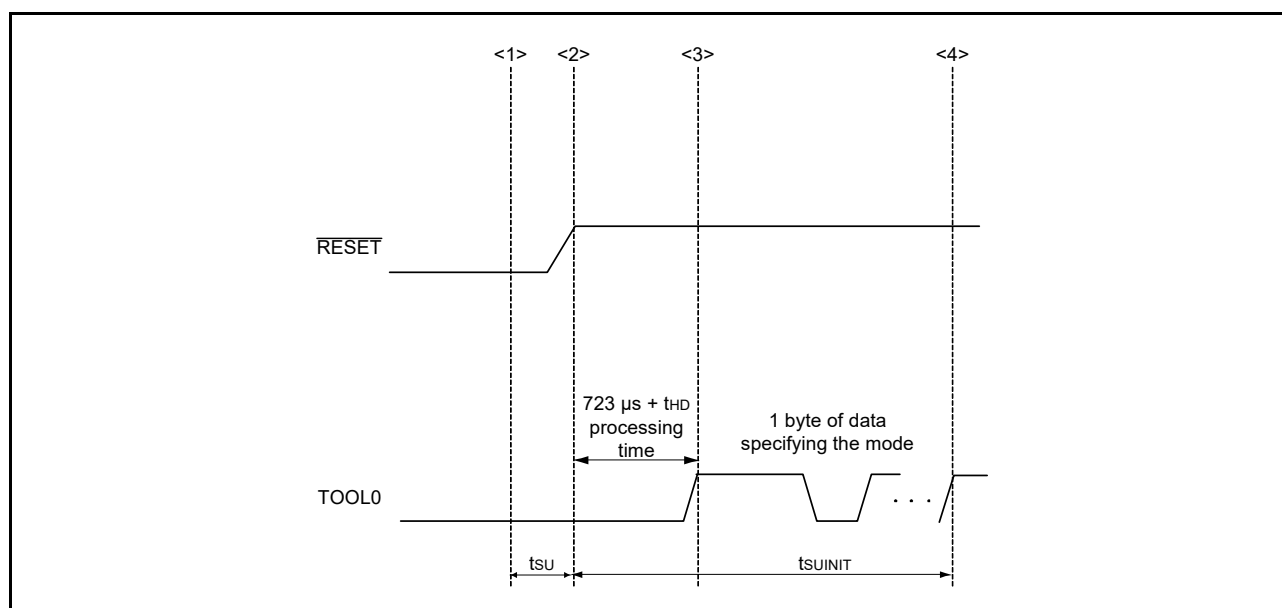
<For serial programming by using an external device (UART communications)>

Set the TOOL0 pin to the low level, and then release the MCU from the reset state (see **Table 38 - 5**). After that, enter flash memory programming mode according to steps <1> to <4> shown in **Figure 38 - 7**. For details, refer to the application note, RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756).

Table 38 - 5 Relationship between the Voltage Applied to the TOOL0 Pin and the Operating Mode after Release from the Reset State

TOOL0	Operating Mode
VDD	Normal operation mode
0 V	Flash memory programming mode

Figure 38 - 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 tsuINIT : 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.

tsu : This is the time from the TOOL0 pin being placed at the low level until the end of the external reset.

tHD : This is the time over which the TOOL0 pin must be kept at the low level following the end of an external reset (excluding the processing time of the firmware to control the flash memory).

For details, see **42.10 Timing of Entry to Flash Memory Programming Modes**.

38.4.3 Selecting communications mode

The communications modes of the RL78 microcontroller are shown in the table below.

Table 38 - 6 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.

38.4.4 Communications commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 38 - 7**. 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 38 - 7 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.
Erase	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 and boot cluster size or bank swapping setting (BTBLS).
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 38 - 8** and **38 - 9** show the signature data list and an example of signature data, respectively.

Table 38 - 8 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 7FFFFH (512 Kbytes) → FFH, FFH, 07H)	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 F2FFFFH (8 Kbytes) → FFH, 2FH, 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. 4.00 → 04H, 00H, 00H)	3 bytes

Table 38 - 9 Example of Signature Data

Field Name	Description	Number of Bytes of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol C	3 bytes	10 00 0D
Device name	R7F100LPL	10 bytes	52 = "R" 37 = "7" 46 = "F" 31 = "1" 30 = "0" 30 = "0" 4C = "L" 50 = "P" 4C = "L" 20 = " "
Code flash memory area end address	Code flash memory area 00000H to 7FFFFH (512 Kbytes)	3 bytes	FF FF 07
Data flash memory area end address	Data flash memory area F1000H to F2FFFFH (8 Kbytes)	3 bytes	FF 2F 0F
Firmware version	Ver. 4.00	3 bytes	04 00 00

38.5 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 38 - 10 Processing Times for Commands When the PG-FP6 Is in Use (Reference Values)

PG-FP6 Command	Code Flash Memory			
	64 Kbytes	128 Kbytes	256 Kbytes	512 Kbytes
Erasure	1.0 s	1.3 s	2.1 s	3.5 s
Writing	1.6 s	2.9 s	5.2 s	9.5 s
Verification	1.2 s	2.2 s	3.9 s	7.3 s
Writing after erasure	2.3 s	4 s	6.9 s	12.7 s

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

38.6 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.

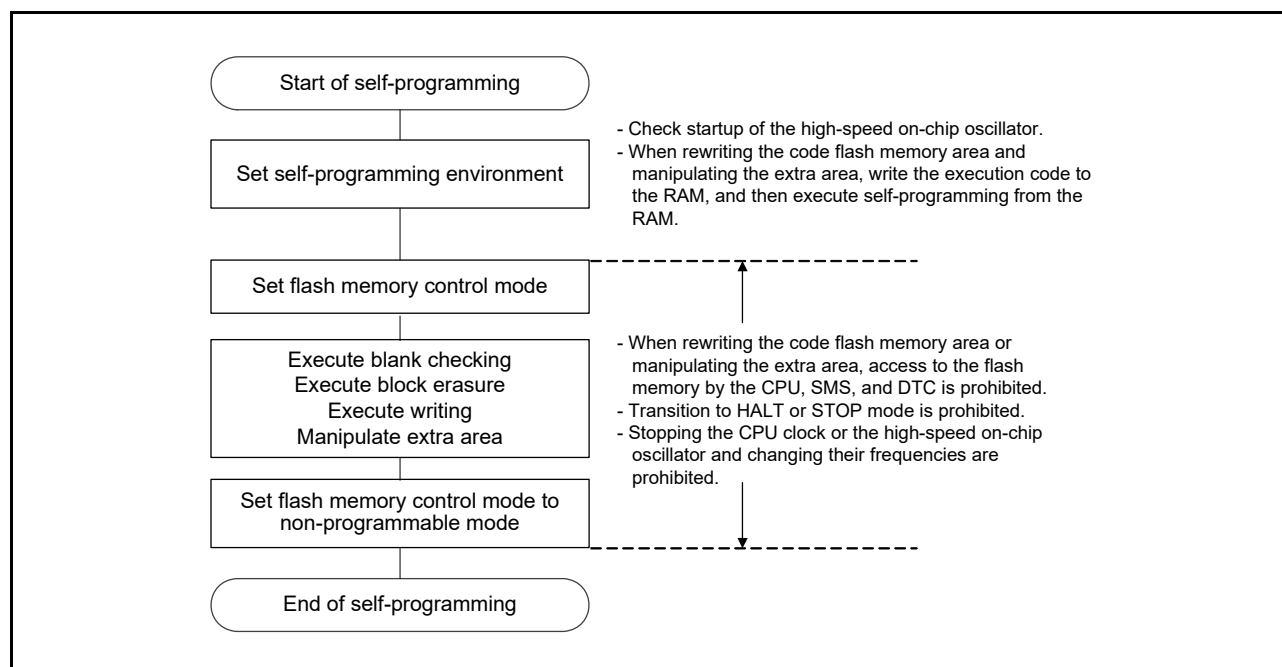
For details, refer to the **RL78 Family Renesas Flash Driver RL78 Type11 User's Manual (R20UT5539EJ)**.

- Caution 1.** The self-programming function cannot be used when the CPU operates with the subsystem clock (fSUB).
- 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 μ s 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.

38.6.1 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 **38.6.2 Registers for controlling the flash memory**.

Figure 38 - 8 Flow of Self-programming (Rewriting the Flash Memory)



38.6.2 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)

38.6.2.1 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 38 - 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 (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

38.6.2.2 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 38 - 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 (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 3. The settings of the EWA1 and EWA0 bits are meaningless during programming of the code flash memory.

Table 38 - 11 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

38.6.2.3 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 38 - 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 (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

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.

38.6.2.4 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 **38.6.3 Setting the flash memory control mode**. The PFCMD register can be set by an 8-bit memory manipulation instruction.

Figure 38 - 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

38.6.2.5 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 **38.6.3.1 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 38 - 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.							

38.6.2.6 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 **38.6.3 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 38 - 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 (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register), rewriting to the FLPMC register is enabled.

38.6.2.7 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 38 - 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).

38.6.2.8 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 38 - 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	For use in boot swapping: Selection of boot area setting ^{Note}	For use in bank swapping: Selection of startup bank setting ^{Note}
0	Specifies the boot area according to the setting of EX bit 8 (BTFLG) in the security flag and boot swapping 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)	Specifies the startup bank according to the setting of EX bit 8 (BTFLG) in the security flag and boot swapping setting area of the extra area. BTFLG = 0: Bank 1 as the startup bank BTFLG = 1: Bank 0 as the startup bank (default)
1	Specifies the boot area according to the setting of the TMBTSEL bit.	Specifies the startup bank according to the setting of the TMBTSEL bit.

TMBTSEL	For use in boot swapping: Selection of boot area when TMSPMD = 1 ^{Note}	For use in bank swapping: Selection of startup bank when TMSPMD = 1 ^{Note}
0	Boot cluster 0 as the boot area	Bank 0 as the startup bank
1	Boot cluster 1 as the boot area	Bank 1 as the startup bank

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 38 - 12 .

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 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. The set values of the boot area in the BTFLG bit 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 started as the boot area at 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.

Caution 3. When the setting of the BANKPGEN bit in the FLMODE register is 1 (specifying the bank programming mode), changing the values of the TMSPMD and TMBTSEL bits is prohibited.

Table 38 - 12 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 the operating frequency 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 to the value (00100B) 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.

38.6.2.9 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 38 - 17 Format of Flash Memory Sequencer Control Register (FSSQ)

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.								
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								

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

- 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 **38.6.6.4 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 **38.6.6.4 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 **38.6.6.4 Operations for rewriting the code flash memory area** and **38.6.6.5 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).

38.6.2.10 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 38 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE)

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.

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 (ESEQER) 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 (ESEQER) to 1.
0	1	1	1	• Write to the security flag and boot swapping and bank swapping setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash memory security flag and boot swapping and bank swapping setting areas 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, boot swapping setting area, and bank swapping setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) 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 the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

38.6.2.11 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 **38.6.4 Initializing the registers for use with the flash memory sequencer**. The value of this register following a reset is 00H.

Figure 38 - 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 the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

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.

38.6.2.12 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 38 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (1/2)

Address: F02CBH
After reset: 00H/04H
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTH	ESQEND	SQEND	0	0	0	×	0	0

Address: F02CAH
After reset: 00H/80H^{Note}
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTL	MBTSEL	MOPEN	ESEQER	SEQR	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}	
	For use in boot swapping	For use in bank swapping
0	BTFLG = 1 (Boot cluster 0 as the boot area)	BTFLG = 1 (Bank 0 as the startup bank)
1	BTFLG = 0 (Boot cluster 1 as the boot area)	BTFLG = 0 (Bank 1 as the startup bank)

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.

ESEQER	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.	

Figure 38 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (2/2)

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 checking 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 or bank area switching or boot area switching flag) stored in the extra area.

38.6.2.13 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 38 - 21 Format of Flash Security Flag Monitoring Register (FLSEC) (1/2)

Address: F00B0H
 After reset: Undefined
 R/W: R

Symbol	15	14	13	12	11	10	9	8
FLSEC	BAPR	0	0	WRPR	0	SEPR	BTPR	BTFLG
	7	6	5	4	3	2	1	0
	BTBLS3	BTBLS2	BTBLS1	BTBLS0	SWPR	0	IFPR	IDEN
BAPR	Changing of the boot cluster size and bank swapping setting (BTBLS[3:0]) disabled flag							
0	Changing of the BTBLS[3:0] setting area is disabled.							
1	Changing of the BTBLS[3:0] setting area is enabled.							
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	For use in boot swapping: Block erase-disabled flag				For use in bank swapping: Startup bank switching-disabled flag			
0	Writing of the boot area is disabled.				Switching of the startup bank is disabled.			
1	Writing of the boot area is enabled.				Switching of the startup bank is enabled.			
BTFLG	For use in boot swapping: Boot area switching flag				For use in bank swapping: Startup bank switching flag			
0	Boot cluster 1 as the boot area				Bank 1 as the startup bank			
1	Boot cluster 0 as the boot area				Bank 0 as the startup bank			

Figure 38 - 21 Format of Flash Security Flag Monitoring Register (FLSEC) (2/2)

BTBLS3	BTBLS2	BTBLS1	BTBLS0	Boot cluster size setting flag	Bank swapping setting flag
0	0	0	0	2 Kbytes	Setting prohibited
0	0	0	1	4 Kbytes	Setting prohibited
0	0	1	0	8 Kbytes	Setting prohibited
0	0	1	1	16 Kbytes	Setting prohibited
0	1	0	0	32 Kbytes ^{Note 1}	Setting prohibited
0	1	0	1	64 Kbytes ^{Note 2}	Setting prohibited
0	1	1	0	128 Kbytes ^{Note 3}	Setting prohibited
0	1	1	1	Setting prohibited	Bank swapping is in use ^{Note 4}
1	1	1	1	16 Kbytes (default)	Setting prohibited
Other than the above				Setting prohibited	

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.

Note 1. This value can be specified in products with 128 Kbytes to 512 Kbytes of code flash memory.

Note 2. This value can be specified in products with 256 Kbytes or 512 Kbytes of code flash memory.

Note 3. This value can be specified in products with 512 Kbytes of code flash memory.

Note 4. The bank size is set to 128 Kbytes in a product with 256 Kbytes of code flash memory and to 256 Kbytes in a product with 512 Kbytes of code flash memory.

38.6.2.14 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 **38.9 Flash Shield Window**. The FLFSWE register can be read by a 16-bit memory manipulation instruction.

Figure 38 - 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 38 - 16 Relationship between Flash Shield Window Function Setting/Change Methods and Commands**.

38.6.2.15 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 **38.9 Flash Shield Window**. The FLFSWS register can be read by a 16-bit memory manipulation instruction.

Figure 38 - 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							

38.6.2.16 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 38 - 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.

38.6.2.17 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 38 - 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 **38.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

38.6.2.18 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 **38.6.7 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 38 - 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.

38.6.3 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 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).

38.6.3.1 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.

38.6.3.2 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).

38.6.3.3 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).

38.6.3.4 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 μs

<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 μs.

<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 μs.

38.6.4 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 clock.
- <3> Clear the FLRST bit.

38.6.5 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 to the value (00100B) 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 **38.6.3.1 Procedure for executing the specific sequence**, **38.6.3.2 Procedure for entry to the code flash memory programming mode**, and **38.6.3.3 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.

38.6.6 Rewriting the flash memory

38.6.6.1 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.

38.6.6.2 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).

38.6.6.3 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 **38.6.2.9 Flash memory sequencer control register (FSSQ)**.

38.6.6.4 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.2 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 **38.6.6.9 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.4 Procedure for entry to the non-programmable mode**.

(Notes are listed on the next page.)

- 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 details on 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).

38.6.6.5 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.3 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 **38.6.6.9 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.4 Procedure for entry to the non-programmable mode**.

(Note is listed on the next page.)

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.

38.6.6.6 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.

38.6.6.7 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.2 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 **38.6.6.8 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, boot swapping, and bank swapping 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 **38.6.6.9 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 **38.6.3.1 Procedure for executing the specific sequence** and **38.6.3.4 Procedure for entry to the non-programmable mode**.

38.6.6.8 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 to FSWE0	Flash shield window end block setting area Specify the block number (end block number + 1). Note
FSWS8 to 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, boot swapping, and bank swapping setting area

Set the data in the FLWH and FLWL registers to the security flag, boot swapping, and bank swapping setting area.

For details of the security settings, see **38.10 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.
- Caution 5.** When bank swapping is to be enabled, set the BTPR bit to 1.

38.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas

Follow the procedure below to check 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.

38.6.6.10 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 <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.

38.6.7 Interrupts in code flash memory programming mode

38.6.7.1 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. In the bank programming mode, reference to the code stored in the startup bank is possible.

38.6.7.2 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.

38.6.7.3 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.

- <1> Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- <2> 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).
- <3> Specify an address in RAM in the FLSIVC1 and FLSIVC0 registers.
- <4> Set the VECTCTRL register to 01H to set the RAM address as the branch destination for interrupts.
- <5> 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).

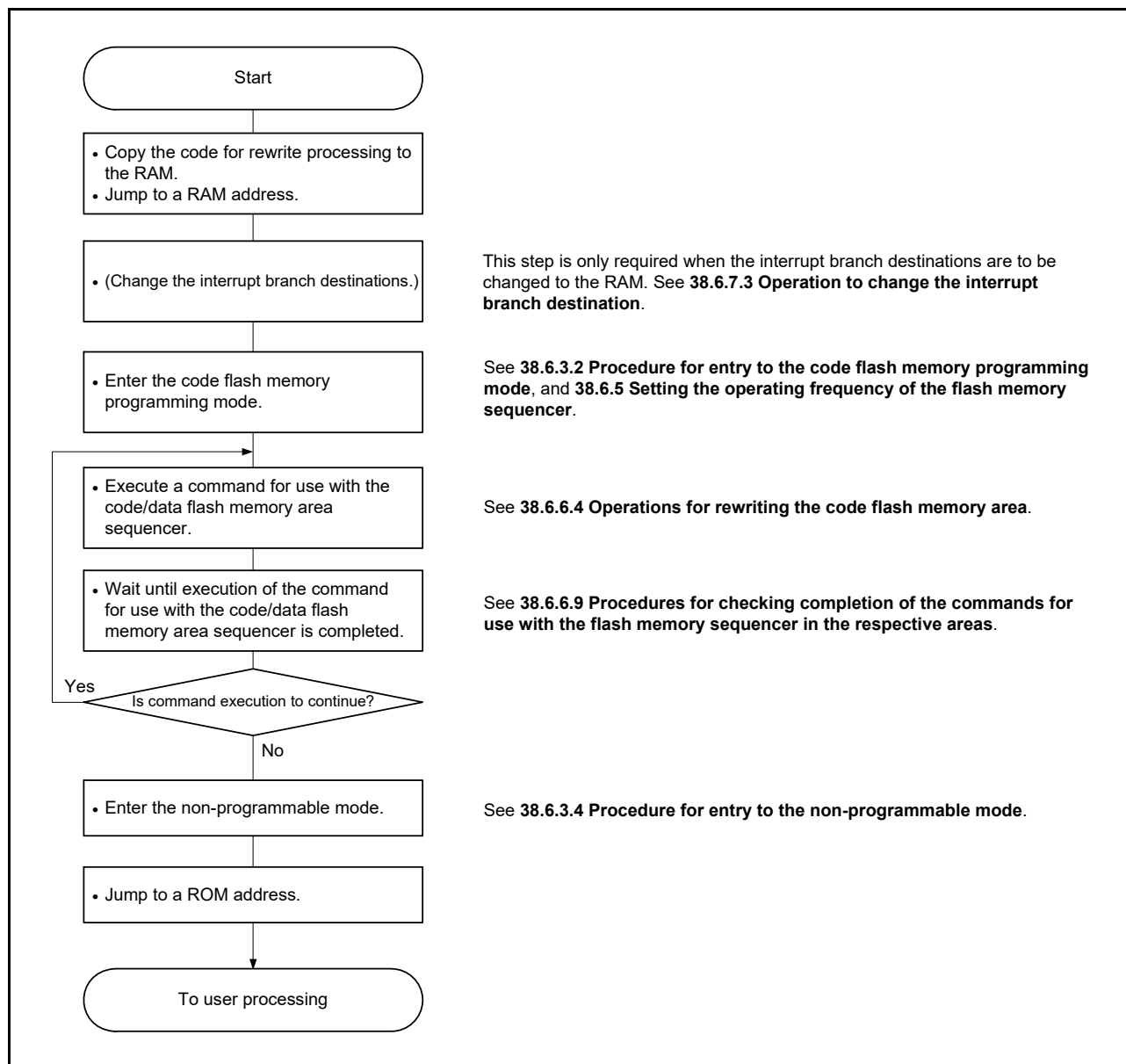
- <1> Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.
- <2> Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 1.
 - 1. Write A5H to the PFCMD register.
 - 2. Write 08H to the 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).
- <3> Set the VECTCTRL register to 00H to set the vector addresses in the ROM as the interrupt branch destinations.
- <4> Restore the saved setting for enabling or disabling interrupts.

38.6.8 Example of executing the commands to rewrite the flash memory areas

38.6.8.1 Example of executing the commands to rewrite the code flash memory area

Figure 38 - 27 shows the flow of executing the commands to rewrite the code flash memory area.

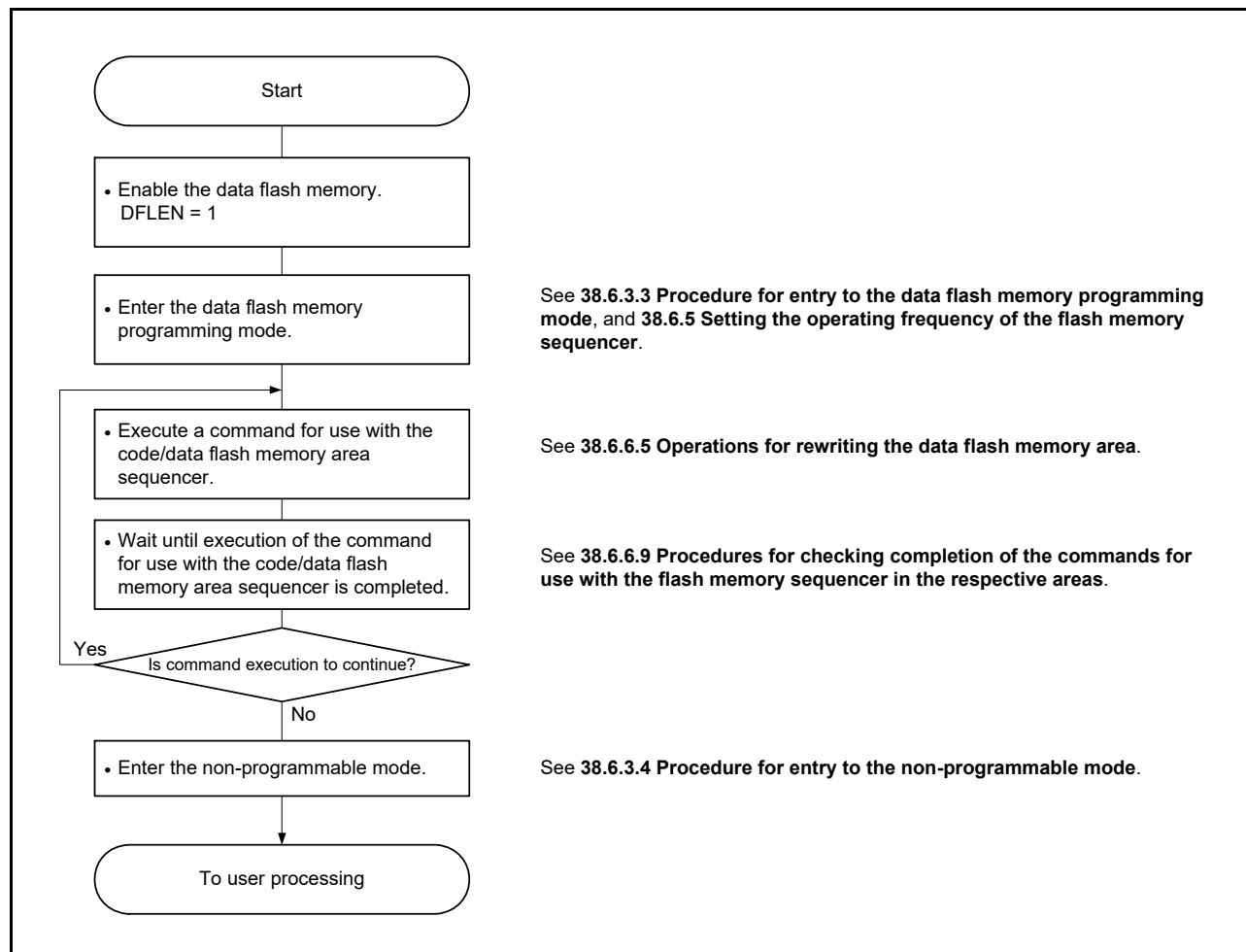
Figure 38 - 27 Flow of Executing the Commands to Rewrite the Code Flash Memory Area



38.6.8.2 Example of executing the commands to rewrite the data flash memory area

Figure 38 - 28 shows the flow of executing the commands to rewrite the data flash memory area.

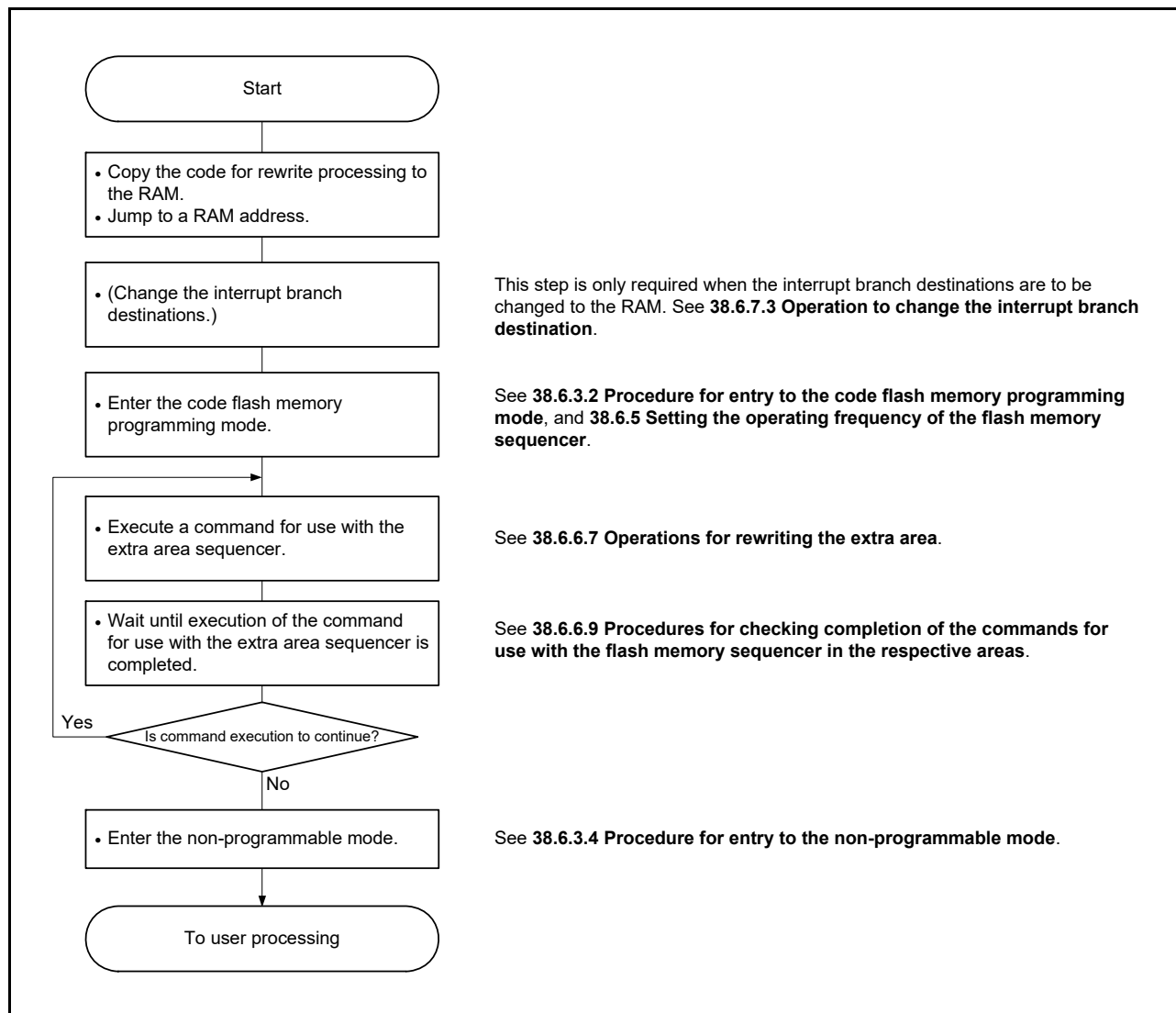
Figure 38 - 28 Flow of Executing the Commands to Rewrite the Data Flash Memory Area



38.6.8.3 Example of executing the commands to rewrite the extra area

Figure 38 - 29 shows the flow of executing the commands to rewrite the extra area.

Figure 38 - 29 Flow of Executing the Commands to Rewrite the Extra Area



38.6.9 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.

5. Specifying the range of code flash memory area

For products that have a 256- or 512-Kbyte code flash memory, do not specify a range that spans areas A and B in **Table 38 - 13** for blank checking or block erasure. For blank checking or block erasure of a range that does span areas A and B, specify an address in each of the ranges A and B, and proceed with separate blank checking or erasure in each range.

Table 38 - 13 Range of Blank Checking and Erasure

Size of the Code Flash Memory	Specifiable Range	Range of Blank Checking ^{Note 1}	Range of Block Erasure ^{Note 2}
256 Kbytes	0 Kbytes < A ≤ 128 Kbytes	00000H to 1FFFFH	000H to 03FH
	128 Kbytes < B ≤ 256 Kbytes	20000H to 3FFFFH	040H to 07FH
512 Kbytes	0 Kbytes < A ≤ 256 Kbytes	00000H to 3FFFFH	000H to 07FH
	256 Kbytes < B ≤ 512 Kbytes	40000H to 7FFFFH	080H to 0FFH

Note 1. Range of addresses that can be specified for a single blank checking operation

Note 2. Range of block numbers that can be specified for a single block erasure operation

38.7 Boot Swapping

The boot area consists of the following areas.

- Area specified as boot cluster size starting from address 00000H
- Vector table area
- CALLT table area
- Option bytes area^{Note 1}
- Setting areas for the on-chip debug security ID and programmer connection ID^{Note 2}
- Program area

The required settings and information respectively for starting the program and for connecting an on-chip debugger and programmer are stored in these areas. Thus, 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^{Note 3}. 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^{Note 4} 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 next started following a reset. The boot cluster size is specified by the setting of the BTBLS[3:0] bits. The specifiable size depends on the product. For details, see **38.11 Settings of Boot Cluster Size and Bank Swapping**.

Note 1. The functions of each of the bits in the option bytes area are not reflected until the generation of a reset. To reflect changes to settings in the option bytes area, reset the MCU after boot swapping.

Note 2. Make sure that the values in the areas for setting of the on-chip debugging security ID and programmer connection ID are the same in boot clusters 0 and 1.

Note 3. Set the values shown in **Table 38 - 15 Data for Setting in the Boot Cluster**, as the offset addresses for the starting addresses of the cluster listed in **Table 38 - 14 Starting Addresses of the Boot Cluster Allocated to the Higher-order Address Bits for the Memory Space**, in the new boot program.

Note 4. To change the boot area, use the BTFLG bit allocated to the setting area for the security flag, boot swapping, and bank swapping within the extra area, or the TMSPMD and TMBTSEL bits in the FSSET register.

The following are examples of the addresses settings when the boot cluster size is 16 Kbytes (BTBLS[3:0] = 0011B).

- 04000H to 0407FH (128 bytes): Vector table area
- 04080H to 040BFH (64 bytes): CALLT table area
- 040C0H to 040C3H (4 bytes): Option bytes area
- 040C4H to 040CDH (10 bytes): On-chip debug security ID setting area

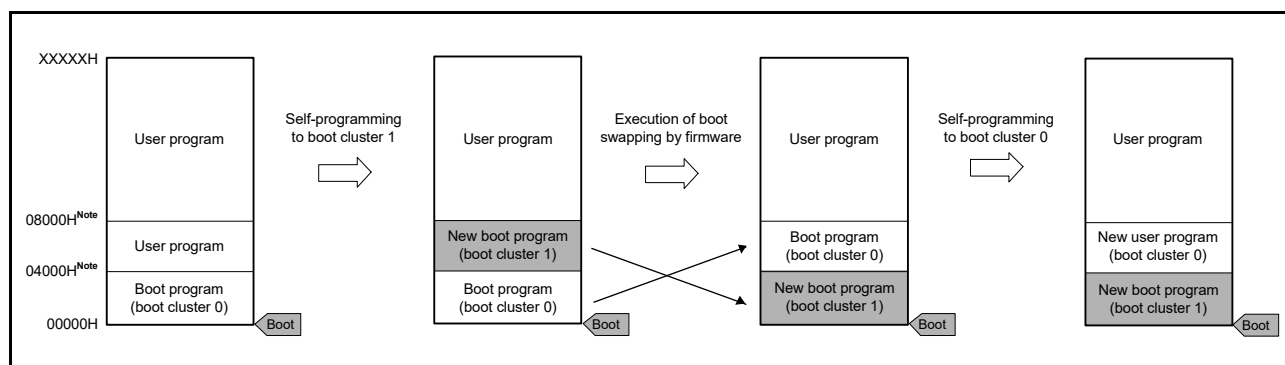
Table 38 - 14 Starting Addresses of the Boot Cluster Allocated to the Higher-order Address Bits for the Memory Space

BTBLS	Boot Cluster Size	Starting Addresses of the Boot Cluster Allocated in the Higher-order Address of the Memory Space
00H	2 Kbytes	00800H
01H	4 Kbytes	01000H
02H	8 Kbytes	02000H
03H	16 Kbytes	04000H
04H	32 Kbytes	08000H
05H	64 Kbytes	10000H
06H	128 Kbytes	20000H

Table 38 - 15 Data for Setting in the Boot Cluster

Offset Address	Size	Setting Data
00000H to 0007FH	128 bytes	Vector table area
00080H to 000BFH	64 bytes	CALLT table
000C0H to 000C3H	4 bytes	Option bytes area
000C4H to 000CDH	10 bytes	Area for setting the on-chip debugging security ID and programmer connection ID

Figure 38 - 30 Boot Swapping



Note The address of all boot clusters in the figure applies when the boot cluster size is 16 Kbytes. The boot cluster size depends on the value specified in the BTBLS[3:0] bits. For details, see **38.11 Settings of Boot Cluster Size and Bank Swapping**.

Remark Boot cluster 0: Boot area before boot swapping
 Boot cluster 1: Boot area after boot swapping

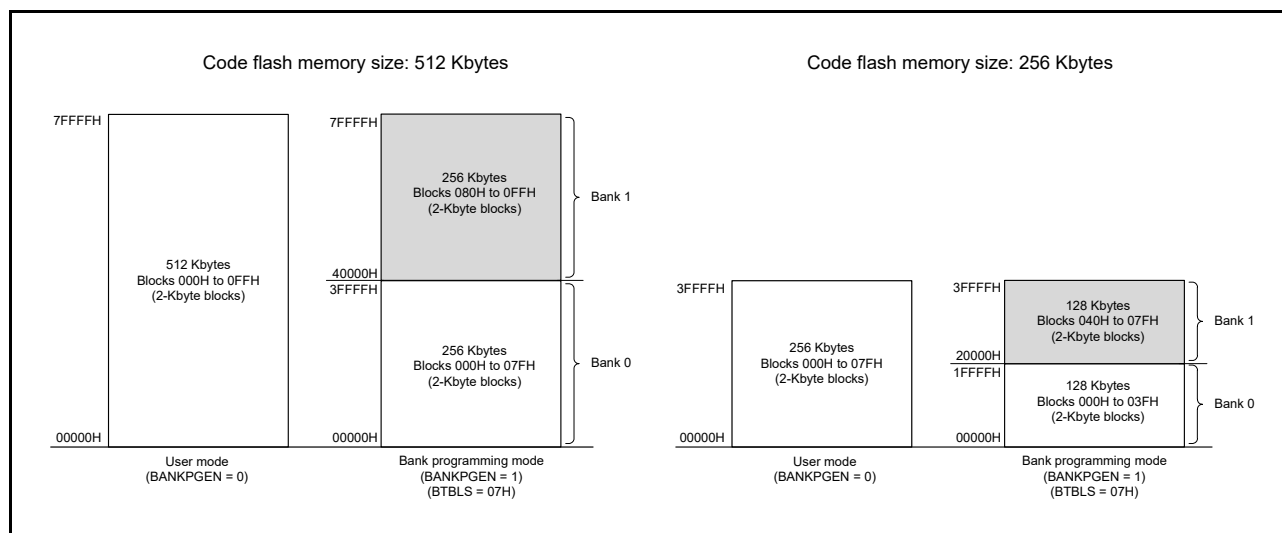
38.8 Bank Programming

This function enables updating of the user program during execution.

38.8.1 Bank mode switching

This function enables switching between the two bank modes: the user mode, in which the user area in the code flash memory is handled as a single area, and the bank programming mode, in which the user area is divided into two bank areas and the user program in one bank area can be updated while execution of the program in the other bank area is in progress. **Figure 38 - 32** shows bank mode switching. The bank mode is selected by setting the BANKPGEN bit in the flash operating mode select register (FLMODE). Bank programming is enabled when the bank programming mode is selected. For the bank size in the bank programming mode, set the BTBLS bits to 07H through serial programming by using the flash memory programmer. For details, see **38.11 Settings of Boot Cluster Size and Bank Swapping**.

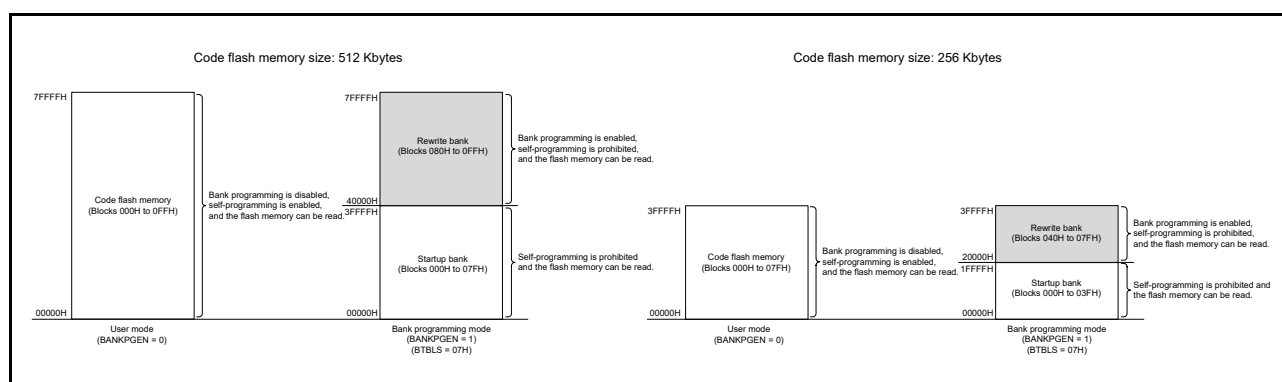
Figure 38 - 32 Bank Mode Switching



38.8.2 Bank programming

This function enables updating (erasure, programming, or verification) in the rewrite bank during execution of the user program from the startup bank or reading from the entire area of the code flash memory when the bank programming mode is selected ($BANKPGEN = 1$). Setting the FLSPM bit in the FLPMC register to 1 in the bank programming mode ($BANKPGEN = 1$) places the startup bank area of the code flash memory in the non-programmable mode and the rewrite bank area in the programming mode. In this state, the startup bank can be read and the rewrite bank can be updated (erased, programmed, or verified). When the FLSPM bit in the FLPMC register is set to 0, the entire area of the code flash memory is in the non-programmable mode and both the startup bank and rewrite bank can be read. Self-programming is prohibited in the bank programming mode. When the user mode is selected ($BANKPGEN = 0$), bank programming is disabled.

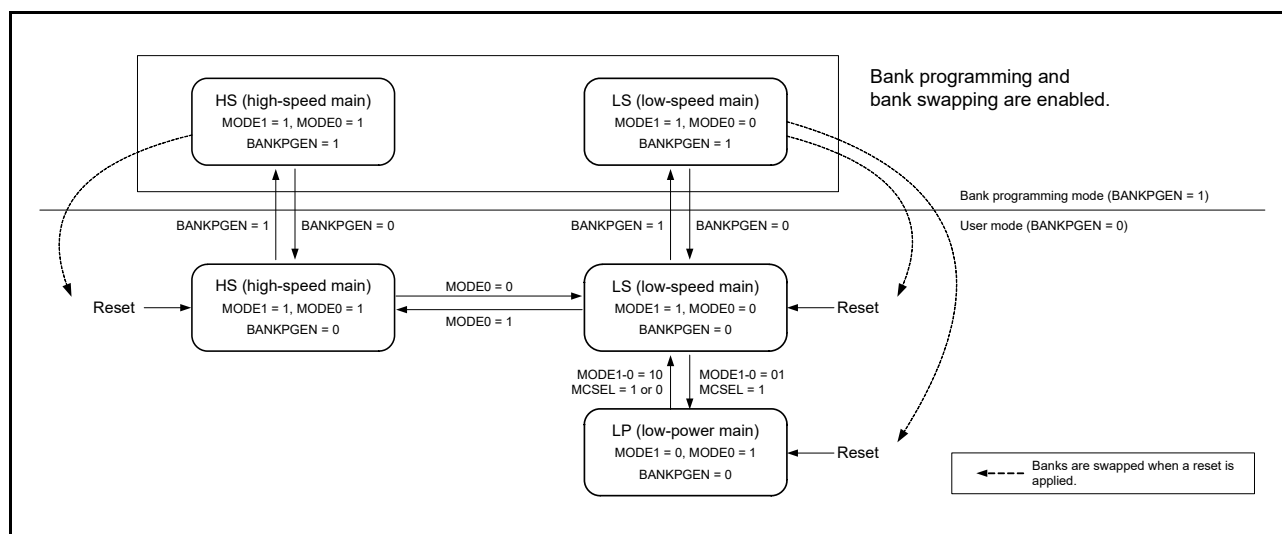
Figure 38 - 33 Bank Programming



Caution Proceed with bank programming in the HS (high-speed main) or LS (low-speed main) mode after having confirmed V_{DD} voltage of no less than 2.2 V through the voltage detector (LVD). Bank programming is prohibited in the LP (low-power main) mode.

When the MCU is in the LP (low-power main) mode, shift it to the LS (low-speed main) mode, after which bank programming and bank swapping can proceed. **Figure 38 - 34** shows the transitions between flash operation modes for bank programming.

Figure 38 - 34 Transitions between Flash Operation Modes for Bank Programming



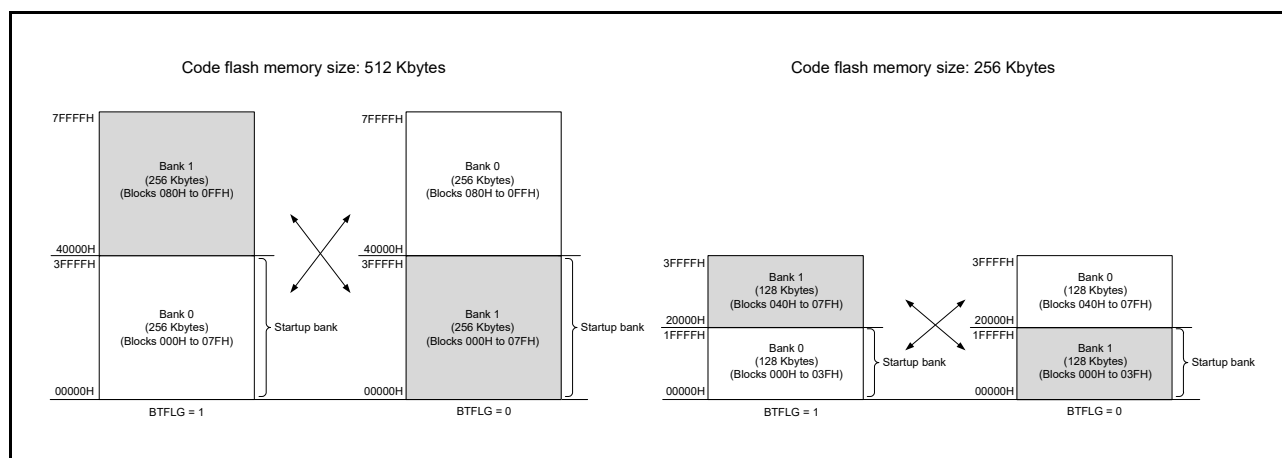
Caution Simultaneous use of background operation (BGO) and bank programming in the bank programming mode is not possible.

38.8.3 Bank swapping

This function is provided to ensure safe updating of the program in the bank programming mode (BANKPGEN = 1) even if updating is interrupted by a reset or for some other reason. **Figure 38 - 35** shows bank swapping. After updating of the program in the bank programming mode (BANKPGEN = 1) is completed, shift to the user mode (BANKPGEN = 0), invert the value of the BTFLG bit for the flash memory, and then use the bank swapping function to switch the flash memory addresses between banks 0 and 1 and start the program from the updated area. Two methods of bank swapping are available: active bank swapping and bank swapping after a reset.

Caution The value of the BTFLG bit cannot be changed in the bank programming mode (BANKPGEN = 1). Be sure to change it in the user mode (BANKPGEN = 0).

Figure 38 - 35 Bank Swapping (User Mode and Bank Programming Mode)



38.8.4 Flash operating mode select register (FLMODE)

The FLMODE register is an 8-bit register for controlling the flash memory operation modes. This register can be set by 1-bit or 8-bit memory manipulation instructions. Note that the value of this register cannot be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 0. The settings of the MODE1 and MODE0 bits following a reset are updated to those of CMODE1 and CMODE0 in the option byte at address 000C2H.

Figure 38 - 36 Format of Flash Operating Mode Select Register (FLMODE)

Address: F00AAH
 After reset: 00H/80H/C0H^{Note 1}
 R/W: R/W

Symbol	<7>	<6>	5	4	3	2	<1>	0
FLMODE	MODE1	MODE0	0	0	0	0	BANKPGEN	0
	BANKPGEN							
	Switching between the user mode and bank programming mode ^{Note 2}							
	0 User mode (bank programing is disabled.)							
	1 Bank programming mode (bank programming is enabled.) ^{Notes 3, 4}							

Note 1. The initial value of the FLMODE register reflects the value of the MODE1 and MODE0 bits updated from the settings of the CMODE1 and CMODE0 bits in the option byte at address 000C2H.

Note 2. Only change this value in the non-programmable mode (the EEEMD and FLSPM bits in the FLPMC register are 0).

Note 3. Select bank swapping by the BTBLS bits (BTBLS = 07F) before setting the BANKPGEN bit. For details, see **38.11 Settings of Boot Cluster Size and Bank Swapping.**

Note 4. The extra area cannot be written to in the bank programming mode. Be sure to set the BANKPGEN bit to 0 before rewriting the extra area.

Caution 1. The value of the FLMODE register can only be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 1. After having changed the value of the FLMODE register, set the FLMWEN bit to 0. Set the FLMWEN bit to 1 before writing to the BANKPGEN bit.

Caution 2. When rewriting the FLMODE register, insert at least one clock cycle of the CPU/peripheral hardware clock (fCLK) after having rewritten the FLMODE register. Do not consecutively write to the FLMODE register.

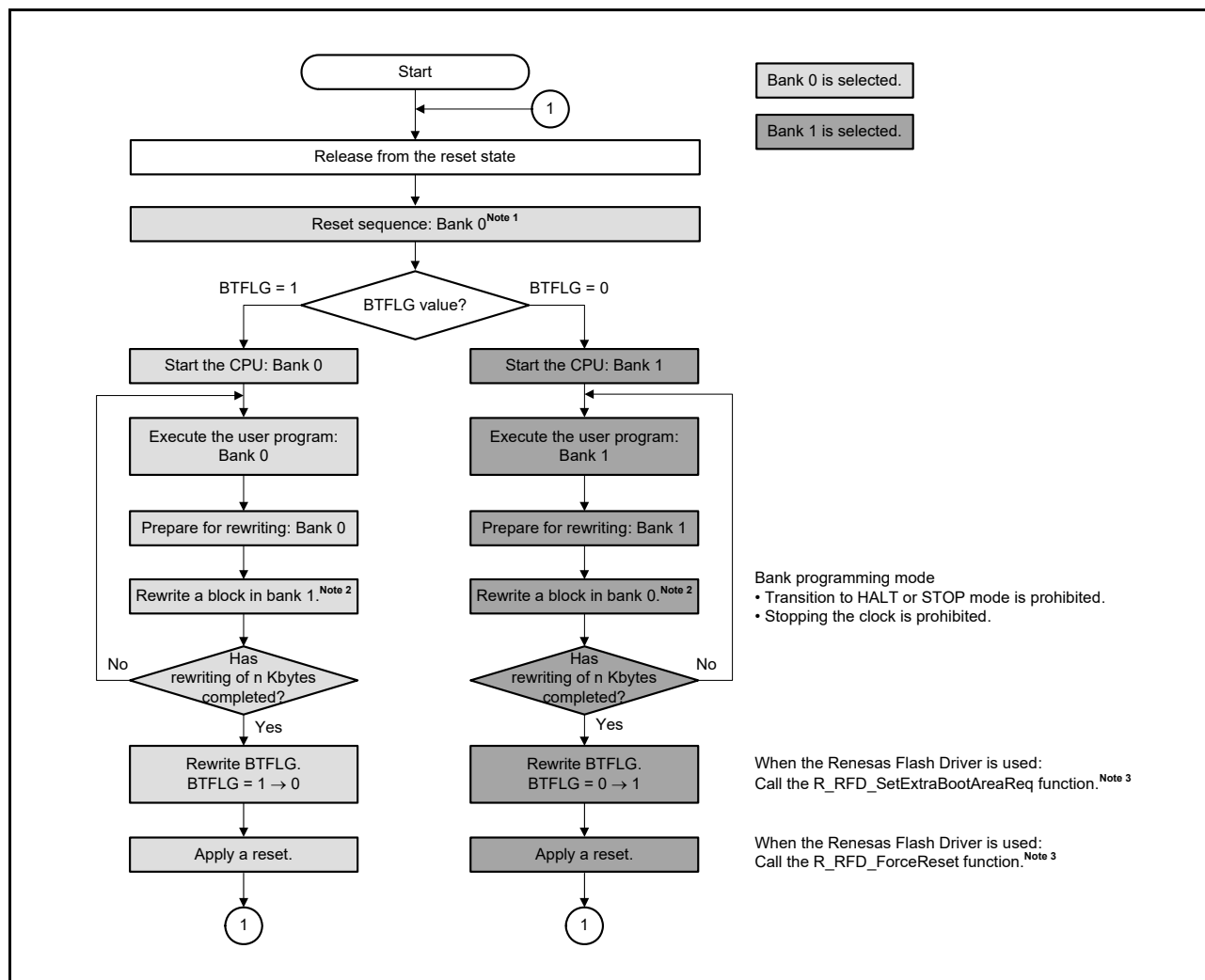
Caution 3. Do not change the FLMODE register while rewriting the flash memory.

Caution 4. Rewriting the FLMODE register should proceed 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.

38.8.5 Procedures for updating the program

The following shows the procedures for updating the user program during execution through the use of bank programming and bank swapping.

Figure 38 - 37 Procedure for Updating the Program (Bank Swapping after a Reset)



Note 1. The values of the option bytes are read from bank 0. Therefore, set the option byte areas in bank 0 and bank 1 and the on-chip debug security ID setting area to the same values.

Note 2. While the rewrite bank is in the programming mode (the FLSPM bit in the FLPMC register is 1), the rewrite bank is not readable. Access to the rewrite bank while the rewrite bank is in the programming mode is prohibited.

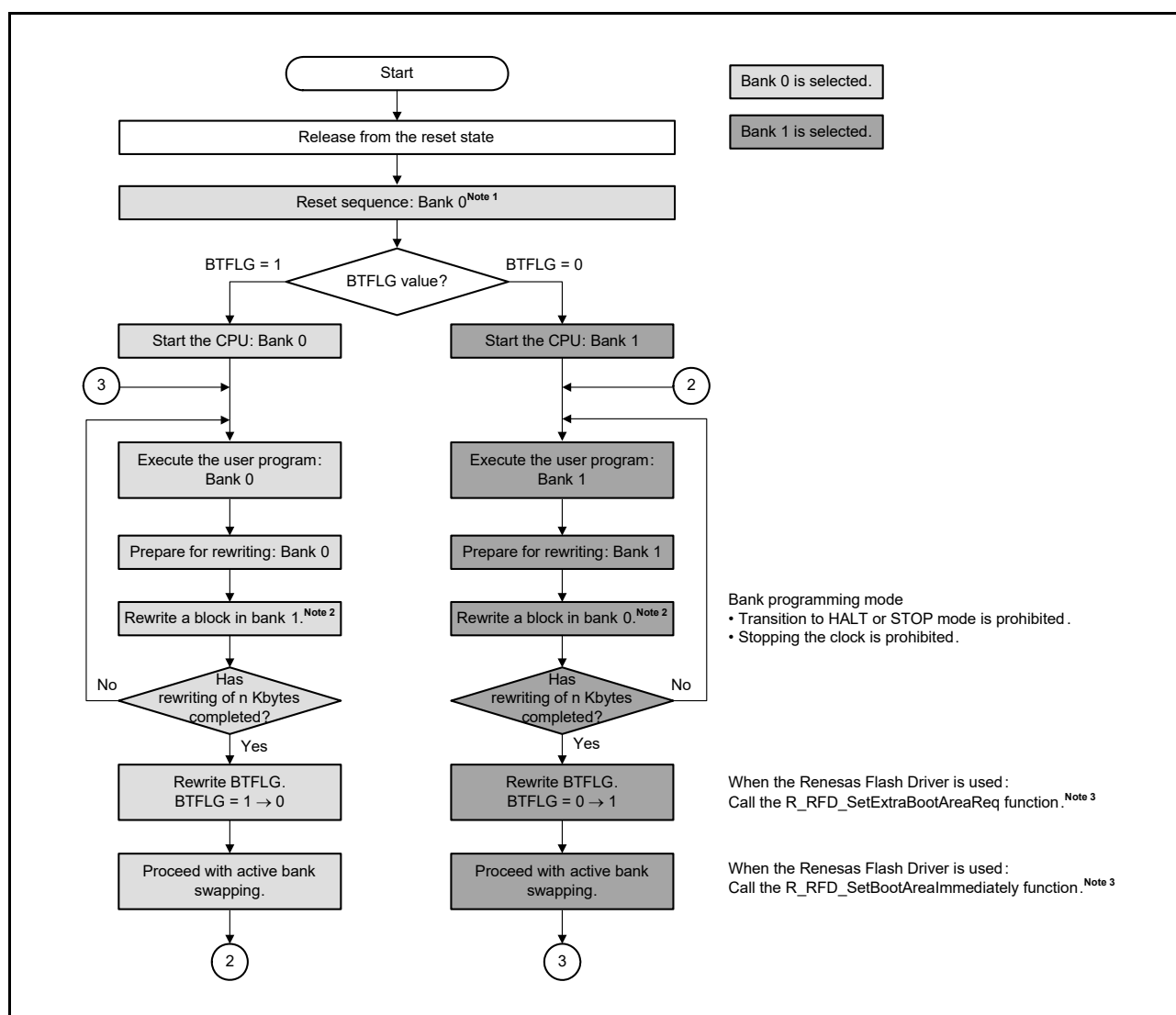
Note 3. For details of the Flash Driver functions, refer to the **RL78 Family Renesas Flash Driver RL78 Type11 User's Manual (R20UT5539EJ)**.

Caution When bank 1 is selected as the startup bank (BANKPGEN = 1 and BTFLG = 0), rewriting of the following blocks containing the option byte area in the respective cases is prohibited.

- Block 080H at address range from 40000H to 407FFH in products with 512 Kbytes of code flash memory
- Block 040H at address range from 20000H to 207FFH in products with 256 Kbytes of code flash memory

If rewriting of bank 0 has been interrupted and failed due to power being cut off, a reset caused by an external source, or for some other reason, the damaged option byte in bank 0 will be read in the subsequent reset sequence and correct booting up will not be possible.

Figure 38 - 38 Procedure for Updating the Program (Active Bank Swapping)



Note 1. The values of the option bytes are read from bank 0. Therefore, set the option byte areas in bank 0 and bank 1 and the on-chip debug security ID setting area to the same values.

Note 2. While the rewrite bank is in the programming mode (the FLSPM bit in the FLPMC register is 1), the rewrite bank is not readable. Access to the rewrite bank while the rewrite bank is in the programming mode is prohibited.

Note 3. For details of the Flash Driver functions, refer to the **RL78 Family Renesas Flash Driver RL78 Type11 User's Manual (R20UT5539EJ)**.

Caution When bank 1 is selected as the startup bank (BANKPGEN = 1 and BTFLG = 0), rewriting of the following blocks containing the option byte area in the respective cases is prohibited.

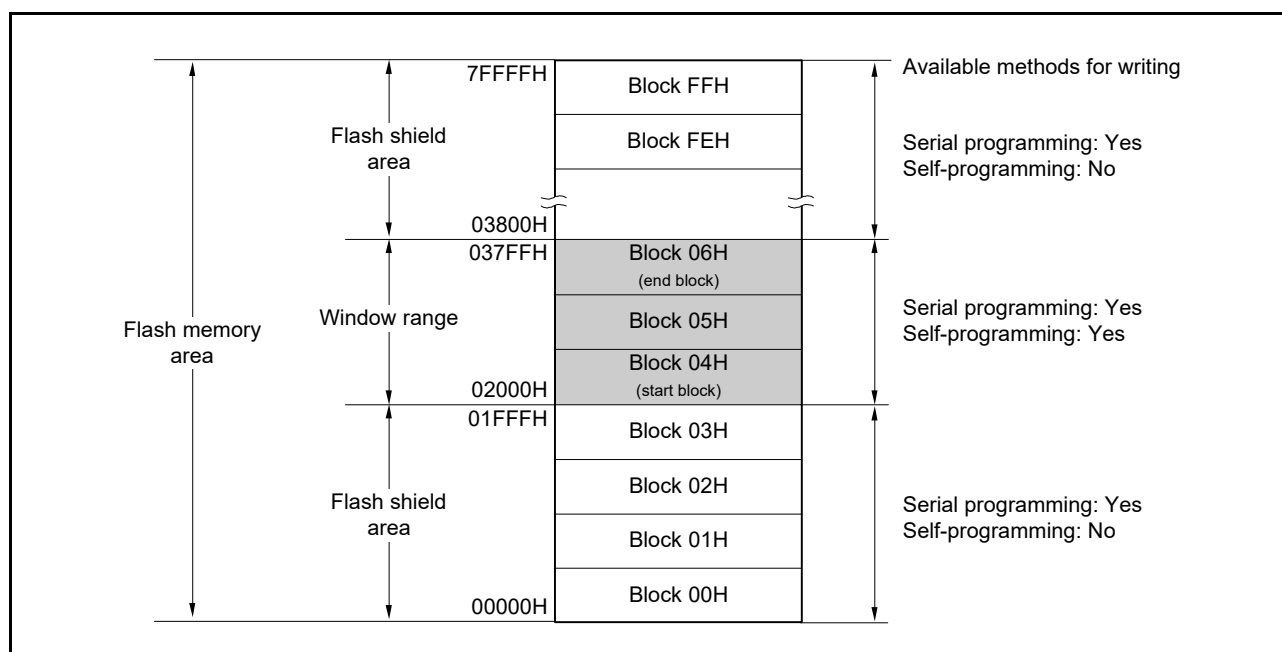
- Block 080H at address range from 40000H to 407FFH in products with 512 Kbytes of code flash memory
- Block 040H at address range from 20000H to 207FFH in products with 256 Kbytes of code flash memory

If rewriting of bank 0 has been interrupted and failed due to power being cut off, a reset caused by an external source, or for some other reason, the damaged option byte in bank 0 will be read in the subsequent reset sequence and correct booting up will not be possible.

38.9 Flash Shield Window

The flash shield window serves 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 38 - 39 Example of Setting the Flash Shield Window
(Target Devices: R7F100LPL, 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 38 - 16 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.

38.10 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 following operations can proceed by serial programming or self-programming.

- **Disabling block erasure**
Execution of the block erase command for a specific block in the code flash memory and data 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 and data 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 changing of the boot cluster size and bank swapping setting (BTBLS bits)**
Writing of the BTBLS bits that are used to set boot swapping or bank swapping is disabled. After the setting to disable 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 03FFFH)^{Note 1} 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 2}. 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 38 - 17** shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Note 1. This address range applies when the size of the boot area is 16 Kbytes. The size of the boot area depends on the value specified in the BTBLS bits. For details, see **38.7 Boot Swapping** and **38.11 Settings of Boot Cluster Size and Bank Swapping**.

Note 2. 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 **38.9 Flash Shield Window** for detail).

Table 38 - 17 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 1
Prohibition of writing	Blocks can be erased.	Data cannot be written.
Prohibition of changing the boot cluster size or bank swapping setting (BTBLS bits)	Blocks can be erased.	Data can be written.
Prohibition of rewriting the boot area Note 2	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 1. 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.

Note 2. Prohibition of writing the boot area cannot be set when bank programming mode is in use. If set, bank programming cannot be executed.

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 changing the boot cluster size or bank swapping setting (BTBLS bits)	Blocks can be erased.	Data can be written.
Prohibition of rewriting the boot area Note	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.

Note Prohibition of rewriting the boot area cannot be set when bank programming mode is in use. If set, bank programming cannot be executed.

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **38.9 Flash Shield Window** for detail).

Table 38 - 18 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 changing the boot cluster size or bank swapping setting (BTBLS bits)		
Prohibition of rewriting the boot area ^{Note}		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

Note Prohibition of rewriting the boot area cannot be set when bank programming mode is in use. If set, bank programming cannot be executed.

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 ^{Note}		
Prohibition of connection to the programmer and on-chip debugger		Disabling the setting is not possible.
Enabling programmer connection ID authentication		

Note Prohibition of writing the boot area cannot be set when bank programming mode is in use. If set, bank programming cannot be executed.

38.11 Settings of Boot Cluster Size and Bank Swapping

To use the boot swapping or bank swapping function, make settings of the boot cluster size or for bank swapping in the BTBLS bits. The settings of the BTBLS bits can be protected by setting the BAPR bit. Use the dedicated flash memory programmer to set the BTBLS and BAPR bits. The BAPR bit is used for security settings. For details, see **38.10 Security Settings**.

Table 38 - 19 Settings of the BTBLS and BAPR Bits

Bit Name	Setting
BTBLS ^{Note 1}	These bits are used to select the boot cluster size. The value specified in the BTBLS bits determines the boot cluster size. 00H: 2 Kbytes 01H: 4 Kbytes 02H: 8 Kbytes 03H: 16 Kbytes 04H: 32 Kbytes^{Note 2} 05H: 64 Kbytes^{Note 3} 06H: 128 Kbytes^{Note 4} 0FH: 16 Kbytes (default) Other than the above: Setting prohibited. In products with 256 or 512 Kbytes of code flash memory, specifying the following value disables boot swapping and enables bank swapping.^{Note 5} 07H: Bank size = 128 Kbytes or 256 Kbytes^{Note 6}
BAPR ^{Note 7}	This bit is used to disable changing the settings of the boot cluster size or for bank swapping in the BTBLS bits. 0: Changing the area specified in the BTBLS bits is disabled. 1: Changing the area specified in the BTBLS bits is enabled (default).

<R>

Note 1. In products designated as part of mass production, be sure to change the setting of these bits from the default value.

Note 2. This value can be specified in products with 128 Kbytes to 512 Kbytes of code flash memory.

Note 3. This value can be specified in products with 256 Kbytes or 512 Kbytes of code flash memory.

Note 4. This value can be specified in products with 512 Kbytes of code flash memory.

Note 5. When bank swapping is to be enabled, set the BTPR bit to 1.

Note 6. The bank size is set to 128 Kbytes in a product with 256 Kbytes of code flash memory and to 256 Kbytes in a product with 512 Kbytes of code flash memory.

Note 7. Be sure to clear the bit to 0 after having set the BTBLS bits.

Caution 1. The value of the BAPR bit can be changed from 0 (disabled) to 1 (enabled) 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 has been 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 BAPR 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 2. The value of the BTBLS bits cannot be changed while the BAPR bit is set to 0.

38.12 Data Flash Memory

38.12.1 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 self-programming. 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 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 HALT or STOP mode is prohibited while rewriting the data flash memory.
- Simultaneous use of background operation (BGO) and bank programming in the bank programming mode is not possible.

Caution 1. The data flash memory is stopped after release from a reset. 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 self-programming should be executed after 5 μ s have elapsed.

Remark For rewriting the code flash memory via a user program, see **38.6 Self-programming**.

38.12.2 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 self-programming should be 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.

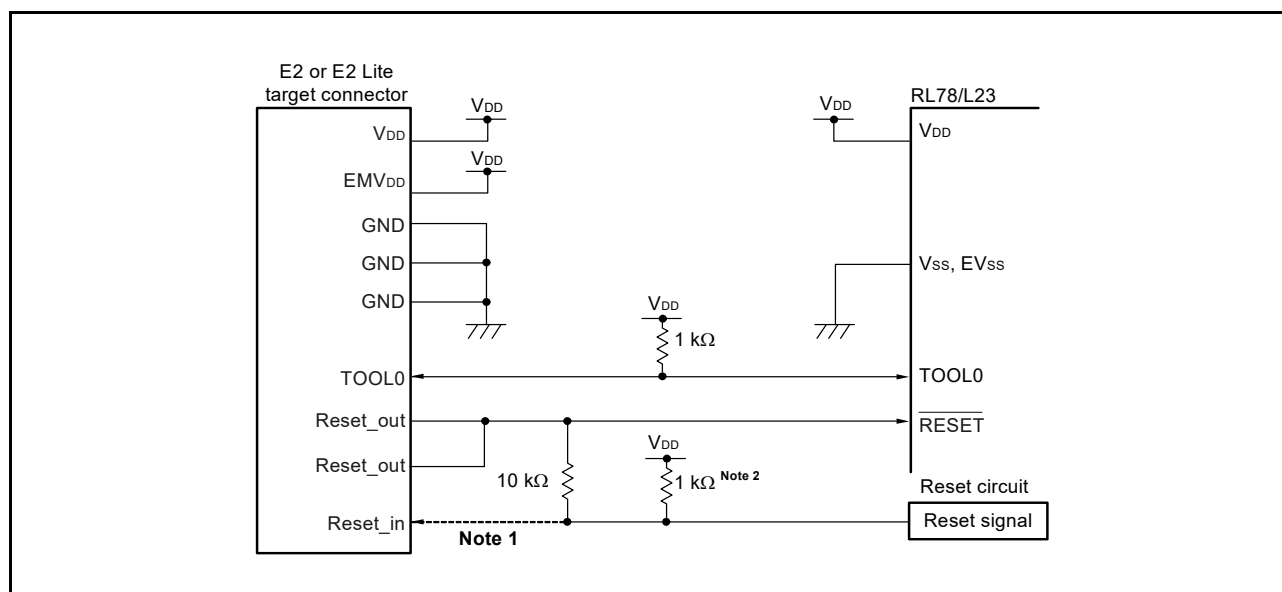
Section 39 On-chip Debugging

39.1 Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/L23

On-chip debugging is handled by connecting the RL78 microcontroller and host machine through the E2 or E2 Lite on-chip debugging emulator. Pins V_{DD} , $\overline{\text{RESET}}$, TOOL0, and V_{SS} are used for connection to the emulator. 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 the 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 39 - 1 Example of Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/L23



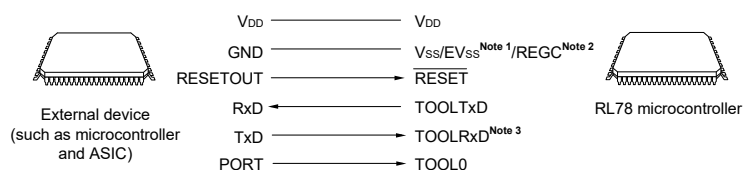
Note 1. The connection shown as a broken line is not required for serial programming.

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 $100\ \Omega$.

39.2 Connection between the External Device that Incorporates UART and RL78/L23

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, $\overline{\text{RESET}}$, 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 the RL78 Debugging Functions Using the Serial Port (R20AN0632EJ0100).



Note 1. This pin is only present in the 100-pin products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μF).

Note 3. 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.

39.3 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 **38.10 Security Settings** in **Section 38 Flash Memory**).
- On-chip debugging control bits in the flash memory at 000C3H (see **Section 37 Option Bytes**)
- An area in the range from 000C4H to 000CDH to hold the security ID code for on-chip debugging.^{Note}

Note The area for holding the security ID code for use in on-chip debugging is also used to hold the ID code for programmer connection ID authentication when a programmer is to be used.

Table 39 - 1 Security ID Code for On-chip Debugging

BTBLS Setting		Address	Security ID Code for On-chip Debugging
The boot cluster is in use.	BTBLS = 00H (2 Kbytes)	000C4H to 000CDH	Any 10-byte ID code ^{Notes 1, 2}
		008C4H to 008CDH	
	BTBLS = 01H (4 Kbytes)	000C4H to 000CDH	
		010C4H to 010CDH	
	BTBLS = 02H (8 Kbytes)	000C4H to 000CDH	
		020C4H to 020CDH	
	BTBLS = 03H (16 Kbytes)	000C4H to 000CDH	
		040C4H to 040CDH	
	BTBLS = 04H (32 Kbytes)	000C4H to 000CDH	
		080C4H to 080CDH	
Bank swapping is in use.	BTBLS = 05H (64 Kbytes)	000C4H to 000CDH	
		100C4H to 100CDH	
	BTBLS = 06H (128 Kbytes)	000C4H to 000CDH	
		200C4H to 200CDH	
	BTBLS = 07H (256 Kbytes)	000C4H to 000CDH	
		200C4H to 200CDH	
	BTBLS = 07H (512 Kbytes)	000C4H to 000CDH	
		400C4H to 400CDH	

Note 1. The setting FFFFFFFF is not allowed.

Note 2. Setting of a 10-byte security ID code for on-chip debugging is required when boot swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0. Specifically, it is to be placed in the address range from the list below that corresponds to the specified length for boot swapping, in addition to the range from 000C4H to 000CDH.

- 128 Kbytes: 200C4H to 200CDH
- 64 Kbytes: 100C4H to 100CDH
- 32 Kbytes: 080C4H to 080CDH
- 16 Kbytes: 040C4H to 040CDH
- 8 Kbytes: 020C4H to 020CDH
- 4 Kbytes: 010C4H to 010CDH
- 2 Kbytes: 008C4H to 008CDH

Setting of a 10-byte security ID code for on-chip debugging is also required when bank swapping is in use or the setting of the BTFLG bit in the FLSEC register is 0. Specifically, it is to be in the one of two address ranges listed below that corresponds to the specified length for bank swapping, in addition to the range from 000C4H to 000CDH.

- 256 Kbytes: 200C4H to 200CDH
- 512 Kbytes: 400C4H to 400CDH

39.4 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 on-chip debugging emulator, 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 39 - 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.

39.5 On-chip Debugging in the Bank Programming Mode

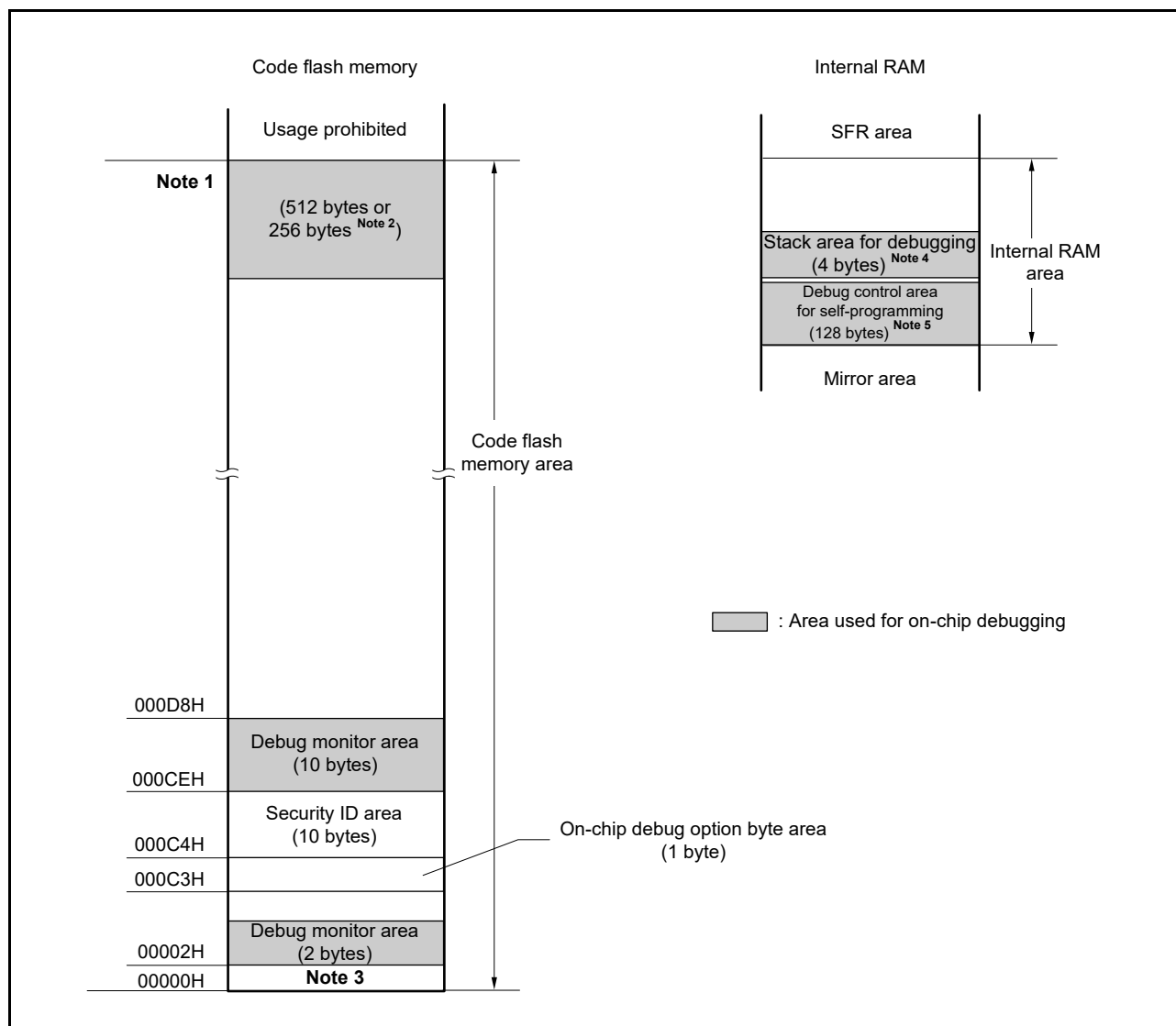
On-chip debugging can be used in the bank programming mode (BANKPGEN = 1).

Updating of data in the bank for rewriting by bank programming can be checked on the debugger. However, the user program cannot be stopped while bank programming is enabled and the bank for rewriting is in programming mode.

For details and usage notes on debugging in the bank programming mode, refer to the **E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (R20UT1994)**.

For details on the bank programming mode, see **38.8 Bank Programming**.

Figure 39 - 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
R7F100LxE (x = F, G, J, L)	0FFFFH
R7F100LxG (x = F, G, J, L, M, P)	1FFFFH
R7F100LxJ (x = F, G, J, L, M, P)	3FFFFH
R7F100LxL (x = F, G, J, L, M, P)	7FFFFH

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.

Note 5. The on-chip debugger uses the 128-byte RAM areas of the products listed in the table below for breaks in self-programming. The correspondences between target products and RAM areas are given in the table.

Products	RAM Area to be Used (128 bytes)
R7F100LxE (x = F, G, J, L)	FBF00H to FBF7FH
R7F100LxG (x = F, G, J, L, M, P)	FBF00H to FBF7FH
R7F100LxJ (x = F, G, J, L, M, P)	F7F00H to F7F7FH
R7F100LxL (x = F, G, J, L, M, P)	F7F00H to F7F7FH

If the setting to disable debugging by the on-chip debugger during self-programming has been made, the RAM areas stated above are not used. For details on the settings for debugging during self-programming, see the user's manual for the integrated development environment in use.

Section 40 BCD Correction Circuit

40.1 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).

40.2 Register for Controlling the BCD Correction Circuit

The following register is used to control the BCD correction circuit.

- BCD correction result register (BCDADJ)

40.2.1 BCD correction result register (BCDADJ)

The BCDADJ register holds correction values for obtaining the addition/subtraction 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 40 - 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								

40.3 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. BCD correction in the interrupt-enabled state requires saving and restoring the A register within the interrupt function. The RETI instruction restores the program status word (PSW), which contains the CY and AC flags.

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. BCD correction in the interrupt-enabled state requires saving and restoring the A register within the interrupt function. The RETI instruction restores the program status word (PSW), which contains the CY and AC flags.

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 41 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)**.

41.1 Conventions Used in Operation List

41.1.1 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 41 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only ^{Note}) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FFF1FH Immediate data or labels (even addresses only ^{Note})
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions ^{Note}) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 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 2nd Special Function Registers** for the symbols of the extended special function registers.

41.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 41 - 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
XH, XL Xs, XH, XL	16-bit registers: XH = higher 8 bits, XL = lower 8 bits 20-bit registers: Xs = (bits 19 to 16), XH = (bits 15 to 8), XL = (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)

41.1.3 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 41 - 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

41.1.4 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 41 - 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.

41.2 Operation List

Table 41 - 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 ^{Note 3} , A	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 2nd 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 $r = \text{A}$

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 41 - 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	$A \leftarrow (DE)$			
		[DE], A	1	1	—	$(DE) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	—	$(ES, DE) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (HL)$			
		[HL], A	1	1	—	$(HL) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	—	$(ES, HL) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow (DE + \text{byte})$			
		[DE+byte], A	2	1	—	$(DE + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((ES, DE) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow (HL + \text{byte})$			
		[HL+byte], A	2	1	—	$(HL + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow (SP + \text{byte})$			
		[SP+byte], A	2	1	—	$(SP + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	—	$(B + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	—	$((ES, B) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	—	$(C + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	—	$((ES, C) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	—	$(BC + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	—	$((ES, BC) + \text{word}) \leftarrow A$			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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 2nd 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 41 - 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 ^{Note 3} , AX	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 2nd 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 41 - 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 (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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 2nd 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 41 - 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 (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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 (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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 2nd 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 41 - 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 2nd 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 41 - 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 2nd 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 41 - 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 2nd 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 41 - 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 2nd 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 41 - 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 2nd 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 41 - 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 (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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	—	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC \leftarrow CS, rp,$ $SP \leftarrow SP - 4$			
		\$!addr20	3	3	—	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow PC + 3 + jdisp16,$ $SP \leftarrow SP - 4$			
		!addr16	3	3	—	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow 0000, addr16,$ $SP \leftarrow SP - 4$			
		!!addr20	4	3	—	$(SP - 2) \leftarrow (PC + 4)_S, (SP - 3) \leftarrow (PC + 4)_H,$ $(SP - 4) \leftarrow (PC + 4)_L, PC \leftarrow addr20,$ $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	—	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PCs \leftarrow 0000,$ $PCH \leftarrow (0000, addr5 + 1),$ $PCL \leftarrow (0000, addr5),$ $SP \leftarrow SP - 4$			
	BRK	—	2	5	—	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 2)_S,$ $(SP - 3) \leftarrow (PC + 2)_H, (SP - 4) \leftarrow (PC + 2)_L,$ $PCs \leftarrow 0000,$ $PCH \leftarrow (0007FH), PCL \leftarrow (0007EH),$ $SP \leftarrow SP - 4, IE \leftarrow 0$			
	RET	—	1	6	—	$PCL \leftarrow (SP), PCH \leftarrow (SP + 1),$ $PCs \leftarrow (SP + 2), SP \leftarrow SP + 4$			
	RETI	—	2	6	—	$PCL \leftarrow (SP), PCH \leftarrow (SP + 1),$ $PCs \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R
	RETB	—	2	6	—	$PCL \leftarrow (SP), PCH \leftarrow (SP + 1),$ $PCs \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or 2nd 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 41 - 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	—	$(SP - 1) \leftarrow rpH, (SP - 2) \leftarrow rpL, SP \leftarrow SP - 2$			
	POP	PSW	2	3	—	$PSW \leftarrow (SP + 1), SP \leftarrow SP + 2$	R	R	R
		rp	1	1	—	$rpL \leftarrow (SP), rpH \leftarrow (SP + 1), SP \leftarrow SP + 2$			
	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 2nd 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. The two numbers indicate the numbers of clock cycles when the condition is not met and when the 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 41 - 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	SEL	RBnNote 4	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 2nd 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. The two numbers indicate the numbers of clock cycles when the condition is not met and when the 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 42 Electrical Characteristics

- Caution 1.** 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.
- Caution 2.** For products that do not have EVss pins, read EVss as Vss.
- Caution 3.** The present pins differ depending on the products. For details, see section 2.1 Functions of Port Pins through section 2.2.1 Functions for each product.

42.1 Absolute Maximum Ratings

(1/2)

Item	Symbols	Conditions	Ratings	Unit
Supply voltage	V _{DD}		-0.5 to +6.5	V
	V _{VSS}		-0.5 to +0.3	V
REGC pin input voltage	V _{IREGC}	REGC	-0.3 to +2.1 and -0.3 to V _{DD} +0.3 ^{Note 1}	V
Input voltage	V _{I1}	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to 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, P21, P121 to P124, P137, EXCLK, EXCLKS, RESET	-0.3 to V _{DD} +0.3 ^{Note 2}	V
Output voltage	V _{O1}	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to P147	-0.3 to V _{DD} +0.3 ^{Note 2}	V
	V _{O2}	P20, P21, P121, P122	-0.3 to V _{DD} +0.3 ^{Note 2}	V
Analog input voltage	V _{AI1}	ANI16 to ANI26	-0.3 to V _{DD} +0.3 and -0.3 to AV _{REFP} +0.3 ^{Notes 2, 3}	V
	V _{AI2}	ANI0, ANI1	-0.3 to V _{DD} +0.3 and -0.3 to AV _{REFP} +0.3 ^{Notes 2, 3}	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μ F). 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 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. AV_{REFP} refers to the positive reference voltage of the A/D converter.

Remark 3. The reference voltage is V_{SS}.

(2/2)

Item	Symbols	Conditions		Ratings	Unit
High-level output current	IOH1	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to P147	-40	mA
		Total of all pins -170 mA	P40 to P47, P66, P67, P130, P140	-70	mA
			P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P64, P65, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P141 to P147	-100	mA
	IOH2	Per pin	P20, P21, P121, P122	-5	mA
		Total of all pins		-20	mA
Low-level output current	IOL1	Per pin	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to P147	40 Note	mA
		Total of all pins 170 mA	P40 to P47, P66, P67, P130, P140	70	mA
			P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P64, P65, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P141 to P147	100	mA
	IOL2	Per pin	P20, P21, P121, P122	10	mA
		Total of all pins		20	mA
Ambient operating temperature	TA	In normal operation mode		-40 to +105	°C
		In flash memory programming mode		-40 to +105	
Storage temperature	T _{stg}			-65 to +150	°C

Note The rating for the following port pins is 80 mA when IOL1 = 40.0 mA is specified by the 40-mA port output control register (PTDC).

Pins P00, P05, and P16 of 44- to 52-pin products, and of 64-pin products with 64 to 128 Kbytes of flash ROM

Pins P07, P10, and P27 of 64-pin products with 256 to 512 Kbytes of flash ROM, and of 80- and 100-pin products

<R>

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 The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

42.2 Characteristics of the Oscillators

42.2.1 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.

42.2.2 Characteristics of the XT1 oscillator

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
XT1 clock oscillation frequency (f _{XT}) ^{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.

42.2.3 Characteristics of the On-chip Oscillators

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

<R>

Item	Symbols	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		HIPREC = 1	-40 to +105°C	-1.0		+1.0	%
		HIPREC = 0	Note 1	-15		0	%
High-speed on-chip oscillator clock correction resolution					0.05		%
Middle-speed on-chip oscillator clock frequency	f _{IM}			1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy				-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	f _{IL}				32.768		kHz
Low-speed on-chip oscillator clock frequency accuracy				-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 listed condition applies when the setting of the FRQSEL3 bit is 1.**Note 2.** The listed values only indicate the characteristics of the oscillators. See AC Characteristics for instruction execution time.**Note 3.** These values are the results of characteristic evaluation and are not checked for shipment.

42.3 DC Characteristics

42.3.1 Pin characteristics

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(1/7)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Allowable high-level output current ^{Note 1}	IOH1	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to P147	1.6 V ≤ VDD ≤ 5.5 V			-10.0 Note 2	mA
		Total of P40 to P47, P66, P67, 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 P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P64, P65, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P141 to 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, P21, 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 = $(IOH \times 0.7)/(n \times 0.01)$

Example when n = 80% and IOH = -10.0 mA

Total output current from the listed pins = $(-10.0 \times 0.7)/(80 \times 0.01) \approx -8.7$ 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.

(Notes, Caution, and Remark continue on the next page.)

Note 4. The maximum value is -30 mA with an ambient operating temperature range of +85°C to +105°C.

Note 5. The maximum value is -50 mA with an ambient operating temperature range of +85°C to +105°C.

Note 6. The maximum values are respectively -100 mA and -60 mA with an ambient operating temperature range of -40 to +85°C and of +85°C to +105°C.

Caution The following pins do not output high-level signals in the N-ch open-drain mode: P00 to P07, P10, P11, P16, P17, P22, P26, P27, P33 to P35, P40 to P47, P50, P52 to P54, P64 to P66, P75 to P77, P80 to P87, P93 to P97, P125, P130, P140, and P143 to P147

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 = EVSS = 0 V)

(2/7)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Allowable low-level output current ^{Note 1}	IOL1	Per pin for P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to P147			20.0 Notes 2, 3	mA
		Per pin for P60 to P63			15.0 Note 2	mA
		Total of P40 to P47, P66, P67, P130, P140 (when duty ≤ 70% ^{Note 4})	4.0 V ≤ VDD ≤ 5.5 V		70.0 Note 5	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 P00 to P07, P10 to P17, P22 to P27, P30 to P35, P50 to P57, P64, P65, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P141 to P147 (when duty ≤ 70% ^{Note 4})	4.0 V ≤ VDD ≤ 5.5 V		80.0 Note 5	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 4})			150.0 Note 6	mA
	IOL2	Per pin for P20, P21, 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 4})	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 EVSS, or VSS pin.**Note 2.** The combination of these and other pins must also not exceed the value for maximum total current.**Note 3.** The maximum rating for the following port pins is 40 mA when IOL1 = 40.0 mA is specified by the 40-mA port output control register (PTDC).

Pins P00, P05, and P16 of 44- to 52-pin products, and of 64-pin products with 64 to 128 Kbytes of flash ROM

Pins P07, P10, and P27 of 64-pin products with 256 to 512 Kbytes of flash ROM, and of 80- and 100-pin products

Note 4. 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 5. The maximum value is 40 mA with an ambient operating temperature range of +85°C to +105°C.**Note 6.** The maximum value is 80 mA 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.

<R>

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(3/7)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Input voltage, high	VIH1	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to P147	Normal input buffer	0.8 VDD	VDD	V
	VIH2	P02, P03, P05, P06, P11, P16, P17, P22, P26, P27, P33, P34, P40 to P47, P51, P53, P54, P64 to P66, P75, P76, P80 to P86, P93, P94, P96, P97, P125, P140 to P142, P144 to P147	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, P21	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 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to P147	Normal input buffer	0	0.2 VDD	V
	VIL2	P02, P03, P05, P06, P11, P16, P17, P22, P26, P27, P33, P34, P40 to P47, P51, P53, P54, P64 to P66, P75, P76, P80 to P86, P93, P94, P96, P97, P125, P140 to P142, P144 to P147	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, P21	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 the following pins is VDD, even in the N-ch open-drain mode: P00 to P07, P10, P11, P16, P17, P22, P26, P27, P33 to P35, P40 to P47, P50, P52 to P54, P64 to P66, P75 to P77, P80 to P87, P93 to P97, P125, P130, P140, and P143 to P147.

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 = EVSS = 0 V)

(4/7)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Output voltage, high	VOH1	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to 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, P21, 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

Caution The following pins do not output high-level signals in the N-ch open-drain mode: P00 to P07, P10, P11, P16, P17, P22, P26, P27, P33 to P35, P40 to P47, P50, P52 to P54, P64 to P66, P75 to P77, P80 to P87, P93 to P97, P125, P130, P140, and P143 to P147

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 = EVSS = 0 V)

(5/7)

Item	Symbols	Conditions			Min.	Typ.	Max.	Unit	
Output voltage, low	VOL1	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P130, P140 to P147	4.0 V ≤ VDD ≤ 5.5 V	IOL1 = 20.0 mA			1.3	V	
				IOL1 = 40.0 mA ^{Note}			1.3	V	
			4.0 V ≤ VDD ≤ 5.5 V	IOL1 = 8.5 mA			0.7	V	
				IOL1 = 17.0 mA ^{Note}			0.7	V	
			2.7 V ≤ VDD ≤ 5.5 V	IOL1 = 3.0 mA			0.6	V	
				IOL1 = 6.0 mA ^{Note}			0.6	V	
			2.7 V ≤ VDD ≤ 5.5 V	IOL1 = 1.5 mA			0.4	V	
				IOL1 = 3.0 mA ^{Note}			0.4	V	
			1.8 V ≤ VDD ≤ 5.5 V	IOL1 = 0.6 mA			0.4	V	
				IOL1 = 1.2 mA ^{Note}			0.4	V	
			1.6 V ≤ VDD ≤ 5.5 V	IOL1 = 0.3 mA			0.4	V	
				IOL1 = 0.6 mA ^{Note}			0.4	V	
	VOL2	P20, P21, 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

Note The listed value applies when IOL1 = 40.0 mA is specified for the following port pins by the 40-mA port output control register (PTDC).

Pins P00, P05, and P16 of 44- to 52-pin products, and of 64-pin products with 64 to 128 Kbytes of flash ROM

<R> Pins P07, P10, and P27 of 64-pin products with 256 to 512 Kbytes of flash ROM, and of 80- and 100-pin products

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 = EVSS = 0 V)

(6/7)

Item	Symbols	Conditions			Min.	Typ.	Max.	Unit
Output current ^{Note}	CCDIOL	P74 to P77, P60 to P63	CCSm = 01H	4.0 V ≤ VDD ≤ 5.5 V	1.0	1.8	2.6	mA
				2.7 V ≤ VDD < 4.0 V	0.8	1.5	2.3	mA
			CCSm = 02H	4.0 V ≤ VDD ≤ 5.5 V	3.0	4.9	6.5	mA
				3.0 V ≤ VDD < 4.0 V	2.7	4.3	5.9	mA
			CCSm = 03H	4.0 V ≤ VDD ≤ 5.5 V	6.6	10.0	13.2	mA
				3.3 V ≤ VDD < 4.0 V	6.0	9.1	12.1	mA
		P60 to P63	CCSm = 04H	4.0 V ≤ VDD ≤ 5.5 V	10.2	15.0	19.8	mA
				3.3 V ≤ VDD < 4.0 V	9.4	13.8	18.2	mA

Note The listed currents apply when the output current control function is enabled.

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(7/7)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input leakage current, high	ILI _{H1}	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to P147	V _I = V _{DD}			0.5	μA
	ILI _{H2}	P20, P21, P137, $\overline{\text{RESET}}$	V _I = V _{DD}			0.5	μA
	ILI _{H3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V _I = V _{DD}			0.5	μA
Input leakage current, low	ILI _{L1}	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P125 to P127, P140 to P147	V _I = EV _{SS}			-0.5	μA
	ILI _{L2}	P20, P21, P137, $\overline{\text{RESET}}$	V _I = V _{SS}			-0.5	μA
	ILI _{L3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V _I = V _{SS}			-0.5	μA
On-chip pull-up resistance	R _U	P00 to P07, P10 to P17, P22 to P27, P30 to P35, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P121, P122, P125 to P127, P140 to P147	V _I = EV _{SS} , In input port	10	20	100	kΩ

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

42.3.2 Supply current characteristics

1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(1/4)

Item	Symbols	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		1.6	—	mA
						VDD = 1.8 V		1.6	—	
					Normal operation	VDD = 5.0 V		3.5	5.9	mA
						VDD = 1.8 V		3.5	5.9	
			LS (low-speed main) mode	fIH = 24 MHz ^{Note 2}	Normal operation	VDD = 5.0 V		2.6	4.5	mA
						VDD = 1.8 V		2.6	4.5	
				fIH = 16 MHz ^{Note 2}	Normal operation	VDD = 5.0 V		1.9	3.2	mA
						VDD = 1.8 V		1.9	3.2	
				fIM = 4 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		0.5	0.9	mA
						VDD = 1.6 V		0.5	0.9	
			LP (low-power main) mode	fIM = 2 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		227	379	μA
						VDD = 1.6 V		226	378	
				fIM = 1 MHz ^{Note 3}	Normal operation	VDD = 5.0 V		126	205	μA
						VDD = 1.6 V		125	204	
			HS (high-speed main) mode	fMX = 20 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		2.3	3.8	mA
						VDD = 1.8 V		2.2	3.8	
			LS (low-speed main) mode	fMX = 20 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		2.2	3.7	mA
						VDD = 1.8 V		2.1	3.6	
				fMX = 20 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		2.3	3.9	mA
						VDD = 1.8 V		2.3	3.8	
				fMX = 10 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		1.1	1.9	mA
						VDD = 1.8 V		1.1	1.9	
				fMX = 10 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		1.2	2.1	mA
						VDD = 1.8 V		1.2	2.1	
				fMX = 8 MHz ^{Note 4} , Square wave input	Normal operation	VDD = 5.0 V		0.9	1.5	mA
						VDD = 1.8 V		0.9	1.5	
				fMX = 8 MHz ^{Note 4} , Resonator connection	Normal operation	VDD = 5.0 V		1.0	1.7	mA
						VDD = 1.8 V		1.0	1.7	

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, VSS, or EVSS. In the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes, the currents in the Typ. column do not include the operating currents of the peripheral modules. The currents in the Max. column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules. The operating currents of other peripheral modules are not included.

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 and Remarks are listed on the next page.)

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. 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 (T_A) is +25°C unless otherwise specified.

1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(2/4)

Item	Symbols	Conditions					Min.	Typ.	Max.	Unit
Supply current Note 1	IDD1	Operating mode	Subsystem clock operation mode	fSUB = 32.768 kHz ^{Note 2} , Low-speed on-chip oscillator operation	Normal operation	TA = -40°C		3.8	6.7	μA
						TA = +25°C		4.2	7.0	
						TA = +50°C		4.7	12.1	
						TA = +70°C		5.6	25.5	
						TA = +85°C		6.8	41.0	
						TA = +105°C		10.8	87.2	
				fSUB = 32.768 kHz ^{Note 3} , Square wave input	Normal operation	TA = -40°C		4.0	7.3	μA
						TA = +25°C		4.2	8.2	
						TA = +50°C		4.7	13.4	
						TA = +70°C		5.5	26.7	
						TA = +85°C		6.7	42.8	
						TA = +105°C		10.7	89.3	
				fSUB = 32.768 kHz ^{Note 3} , Resonator connection	Normal operation	TA = -40°C		3.9	6.9	μA
						TA = +25°C		4.1	7.3	
						TA = +50°C		4.5	12.4	
						TA = +70°C		5.3	25.6	
						TA = +85°C		6.5	41.1	
						TA = +105°C		10.2	87.0	

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, VSS, or EVSS. In the subsystem clock operation mode, the currents in the Typ. and Max. columns do not include the operating currents of the peripheral modules.

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.

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).

Remark 1. fIL: Low-speed on-chip oscillator clock frequency

Remark 2. fSUB: Subsystem clock frequency (XT1 clock oscillation frequency)

1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(3/4)

Item	Symbols	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		0.61	2.24	mA
					VDD = 1.8 V		0.60	2.23	
			LS (low-speed main) mode	f _{IH} = 24 MHz Note 3	VDD = 5.0 V		0.51	1.74	mA
					VDD = 1.8 V		0.50	1.73	
				f _{IH} = 16 MHz Note 3	VDD = 5.0 V		0.47	1.33	mA
					VDD = 1.8 V		0.47	1.32	
				f _{IM} = 4 MHz Note 4	VDD = 5.0 V		0.09	0.31	mA
					VDD = 1.6 V		0.09	0.30	
			LP (low-power main) mode	f _{IM} = 2 MHz Note 4	VDD = 5.0 V		38	140	μA
					VDD = 1.6 V		37	138	
				f _{IM} = 1 MHz Note 4	VDD = 5.0 V		32	86	μA
					VDD = 1.6 V		31	85	
			HS (high-speed main) mode	f _{MX} = 20 MHz Note 5 , Square wave input	VDD = 5.0 V		0.21	1.19	mA
					VDD = 1.8 V		0.14	1.10	
			LS (low-speed main) mode	f _{MX} = 20 MHz Note 5 , Square wave input	VDD = 5.0 V		0.20	1.17	mA
					VDD = 1.8 V		0.14	1.10	
				f _{MX} = 20 MHz Note 5 , Resonator connection	VDD = 5.0 V		0.51	1.56	mA
					VDD = 1.8 V		0.49	1.54	
				f _{MX} = 10 MHz Note 5 , Square wave input	VDD = 5.0 V		0.15	0.65	mA
					VDD = 1.8 V		0.12	0.61	
				f _{MX} = 10 MHz Note 5 , Resonator connection	VDD = 5.0 V		0.30	0.84	mA
					VDD = 1.8 V		0.29	0.82	
				f _{MX} = 8 MHz Note 5 , Square wave input	VDD = 5.0 V		0.13	0.54	mA
					VDD = 1.8 V		0.10	0.50	
				f _{MX} = 8 MHz Note 5 , Resonator connection	VDD = 5.0 V		0.26	0.70	mA
					VDD = 1.8 V		0.25	0.68	

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, VSS, or EVSS. In the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes, the currents in the Typ. column do not include the operating currents of the peripheral modules. The currents in the Max. column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules. The operating currents of other peripheral modules are not included.

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.

(Remarks are listed on the next page.)

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 (T_A) is +25°C unless otherwise specified.

1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(4/4)

Item	Symbols	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		0.76	3.30	μA	
					TA = +25°C		0.90	3.38		
					TA = +50°C		1.12	8.20		
					TA = +70°C		1.61	21.16		
					TA = +85°C		2.53	36.29		
					TA = +105°C		5.39	81.33		
				fSUB = 32.768 kHz, Square wave inputNote 4	TA = -40°C		0.41	3.39	μA	
					TA = +25°C		0.63	4.23		
					TA = +50°C		0.87	9.24		
					TA = +70°C		1.38	22.24		
					TA = +85°C		2.35	38.01		
					TA = +105°C		5.22	83.22		
				fSUB = 32.768 kHz, Resonator connectionNote 5	TA = -40°C		0.36	3.04	μA	
					TA = +25°C		0.50	3.32		
					TA = +50°C		0.72	8.19		
					TA = +70°C		1.22	21.12		
					TA = +85°C		2.11	36.33		
					TA = +105°C		4.92	81.12		
			IDD3	STOP mode	RAMSDMD = 0, RAMSDS = 0Note 6	TA = -40°C		0.24	2.20	μA
						TA = +25°C		0.34	2.20	
						TA = +50°C		0.55	7.00	
						TA = +70°C		1.06	20.00	
						TA = +85°C		1.92	35.00	
						TA = +105°C		4.76	80.00	
RAMSDMD = 1, RAMSDS = 1Note 7	TA = -40°C					0.23	2.20	μA		
	TA = +25°C					0.33	2.20			
	TA = +50°C					0.51	7.00			
	TA = +70°C					0.91	17.00			
	TA = +85°C					1.66	30.00			
	TA = +105°C					4.06	70.00			
RAMSDMD = 1, RAMSDS = 1, 128-Hz realtime clock operationNote 8	TA = -40°C				0.28	2.30	μA			
	TA = +25°C				0.40	2.30				
	TA = +50°C				0.60	7.10				
	TA = +70°C				0.99	17.10				
	TA = +85°C				1.76	30.10				
	TA = +105°C				4.16	70.10				

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

- 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, VSS, or EVSS. In the subsystem clock operation mode, the currents in the Typ. and Max. columns do not include the operating currents of the peripheral modules.
- 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.
- 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.
- 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).
- Note 6.** The listed currents with this setting allow retention of the contents of the entire RAM area. The listed currents apply when the low-speed on-chip oscillator and subsystem clock oscillation are stopped. For the current for operation of the subsystem clock in the STOP mode, refer to that in the HALT mode.
- Note 7.** The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator and subsystem clock oscillation are stopped.
- Note 8.** The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator is stopped, the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 1, 1). The current flowing into the RTC is included.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

<R>

2. Peripheral functions (common to all products)

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(1/3)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
High-speed on-chip oscillator operating current	IFIH Note 1				345		μA
Middle-speed on-chip oscillator operating current	IFIM Note 1				20		μA
Low-speed on-chip oscillator operating current	IFIL Note 1				0.3		μA
RTC operating current	IRTC Notes 1, 2, 3	fRTCCLK = 32.768 kHz			0.005		μA
		fRTCCLK = 128 Hz			0.002		μA
32-bit interval timer operating current	II T Notes 1, 2, 4				0.04		μA
8-bit interval timers operating current	ITMT Notes 1, 2, 5	fSUB = 32.768 kHz, fMAIN stopped	8-bit counter mode × 2 ch operation		0.08		μA
			16-bit counter mode operation		0.06		μA
Timer RJ operating current	ITMRJ Note 6	fSUB = 32.768 kHz, fMAIN stopped, per unit			0.14		μA
16-bit timers KB40, KB41, and KB42 operating current	ITMKB Note 7	Standalone mode, timer output disabled, fHOCO = 64 MHz, per unit			530		μA
External signal sampler operating current	IEXSD Note 8	fSUB = 32.768 kHz, fMAIN stopped			0.03		μA
Watchdog timer operating current	IWDT Notes 1, 2, 9	fIL = 32.768 kHz (typ.)			0.32		μA
Oscillation stop detector operating current	IOSDC Note 1				0.02		μA
A/D converter operating current	IADC Notes 1, 10	When conversion at maximum speed	Normal mode, AVREFP = VDD = 5.0 V		0.95	1.6	mA
			Low voltage mode, AVREFP = VDD = 3.0 V		0.5	0.75	mA
AVREFP current	IADREF Note 11	AVREFP = 5.0 V			52		μA
A/D converter internal reference voltage current	IADREF Note 1				114		μA
Temperature sensor operating current	ITMPS Note 1				110		μA
D/A converter operating current	IDAC Notes 1, 12	Per channel			150		μA
Comparator operating current	ICMP Notes 1, 13				6		μA
LVD operating current	ILVD0 Notes 1, 14				0.02		μA
	ILVD1 Notes 1, 14				0.02		μA

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(2/3)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit	
Self-programming operating current	IFSP Notes 1, 15				2.5	12.2	mA	
Data flash rewrite operating current	IBGO Notes 1, 16				2.5	12.2	mA	
SNOOZE mode sequencer operating current	ISMS Notes 1, 17	f _{IH} = 32 MHz	44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- and 100-pin products		1.5		mA	
		f _{IL} = 32.768 kHz	44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- and 100-pin products		1.6		μA	
SNOOZE operating current	ISNOZ Note 1	f _{IH} = 32 MHz	ADC is in use	The ADC is shifting from the STOP mode to the SNOOZE mode. Note 18		0.6	0.81	mA
				The ADC is operating in the low-voltage mode. AVREFP = VDD = 3.0 V		1.2	1.56	mA
			Simplified SPI (CSI)/UART is in use			0.7	0.92	mA
			SMS Note 19	44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- and 100-pin products		2.0		mA
Low-speed peripheral clock supply current	ISXP Notes 1, 20	RTCLPC = 0			0.22		μA	
Output current control operating current	ICCDA Notes 1, 21	The setting of the CCDE register is not 00H.			100		μA	
	ICCDP Notes 22, 23	Per single controlled current drive port pin	Setting of the low-level output current: Hi-Z		30		μA	
			Setting of the low-level output current: 2 to 15 mA		200		μA	
Operating current of the true random number generator	ITRNG Note 1				1.1		mA	

<R> (TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(3/3)

Item	Symbols	Conditions			Min.	Typ.	Max.	Unit
LCD operating current	ILCD1 Notes 1, 24, 25	External resistance division method	fLCD = fSUB LCD clock = 128 Hz 1/3 bias, four time slices	VDD = 5.0 V VL4 = 5.00 V		0.02		μA
	ILCD2 Note 1	Internal voltage boosting method	fLCD = fSUB LCD clock = 128 Hz 1/3 bias, four time slices, VL1 reference	VDD = 3.0 V VL4 = 3.03 V (VLCD4-0 = 04H)		0.74		μA
				VDD = 5.0 V VL4 = 5.01 V (VLCD4-0 = 18H)		1.03		μA
			fLCD = fSUB LCD clock = 128 Hz 1/3 bias, four time slices, VL2 reference Note 26	VDD = 3.0 V VL4 = 3.03 V (VLCD4-0 = 04H)		0.63		μA
				VDD = 5.0 V VL4 = 5.03 V (VLCD4-0 = 18H)		0.83		μA
	ILCD3 Note 1	Capacitor split method	fLCD = fSUB LCD clock = 128 Hz 1/3 bias, four time slices, VDD reference	VDD = 3.0 V VL4 = 3.00 V		0.14		μA
			fLCD = fSUB LCD clock = 128 Hz 1/3 bias, four time slices, VL4 reference	VDD = 5.0 V VL4 = 3.04 V		0.63		μA

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 8-bit interval timer. It does not include the operating current of the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2 and ITMT when the 8-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 6.** This current only flows to the timer RJ. It does not include the operating current of the XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2 and ITMRJ when the timer RJ 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 7.** This current only flows to the 16-bit timers KB40, KB41, and KB42. It does not include the operating current of the high-speed on-chip oscillator. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2 and ITMKB when the 16-bit timers KB40, KB41, and KB42 are operating or in the HALT mode. When the low-speed on-chip oscillator is selected, IFIL should be included in the supply current.**Note 8.** This current only flows to the external signal sampler. It does not include the operating current of the low-speed on-chip oscillator, XT1 oscillator, or 8-bit interval timer. The supply current of the RL78 microcontrollers is the sum of either IDD1 or IDD2 and IEXSD when the external signal sampler 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 9.** 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 IWDW when the watchdog timer is operating.**Note 10.** 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 11.** This current flows into AVREFP.

- Note 12.** This current only flows to the D/A converter. The supply current of the RL78 microcontrollers is the sum of the values of either IDD1 or IDD2, and IDAC, when the D/A converter is operating or in the HALT mode.
- Note 13.** This current only flows to the comparator. The supply current of the RL78 microcontrollers is the sum of IDD1, IDD2 or IDD3 and ICMP when the comparator is in operation.
- Note 14.** 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 15.** This current only flows during self-programming.
- Note 16.** This current only flows while the data flash memory is being rewritten.
- Note 17.** 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 18.** For shift time to the SNOOZE mode, see **27.3.3 SNOOZE mode**.
- Note 19.** 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.
- Note 20.** 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 21.** This current is added to the supply current when the output voltage control port is set.
- Note 22.** This current does not include the current flowing into the I/O port pins.
- Note 23.** This current flows to VDD.
- Note 24.** This current only flows to the LCD controller/driver. The supply current of the RL78 microcontrollers is the sum of the supply current (IDD1 or IDD2) and LCD operating current (ILCD1, ILCD2, or ILCD3) when the LCD controller/driver is operating while the MCU is operating or in the HALT mode. It does not include the current flowing into the LCD panel. The values in the Typ. and Max. columns apply under the following conditions.
- Setting 20 pins as the segment function and blinking all
 - Selecting fsUB when LCD clock = 128 Hz (LCDC0 = 07H)
 - Setting four time slices and 1/3 bias
- Note 25.** This current does not include the current flowing into the external division resistor when the external resistance division method is in use.
- Note 26.** In addition to these conditions, $V_{L2}(\text{Max.}) + 0.1 \text{ V} \leq V_{DD}$ must be satisfied. For details, see **42.6.8.2 Internal voltage boosting method**.
- Remark 1.** f_{IL}: Low-speed on-chip oscillator clock frequency
- Remark 2.** fsx: 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.

42.4 AC Characteristics

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(1/2)

Item	Symbols	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			1.6 V ≤ VDD ≤ 5.5 V	26.041	30.5	31.3	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		20	MHz	
		1.6 V ≤ VDD < 1.8 V			1		4	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			24			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	
Timer RJ input cycle	tc	TRJIO0, TRJIO1		2.7 V ≤ VDD < 5.5 V	100			ns	
				1.8 V ≤ VDD < 2.7 V	300			ns	
				1.6 V ≤ VDD < 1.8 V	500			ns	
Timer RJ Input high-level width, low-level width	tTJIH, tTJIL	TRJIO0, TRJIO1		2.7 V ≤ VDD < 5.5 V	40			ns	
				1.8 V ≤ VDD < 2.7 V	120			ns	
				1.6 V ≤ VDD < 1.8 V	200			ns	
TO00 to TO07, TKBO00, TKBO01, TKBO10, TKBO11, TKBO20, TKBO21, TRJIO0, TRJIO1, TRJO0, and TRJO1 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 < 5.5 V			2	MHz	

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(2/2)

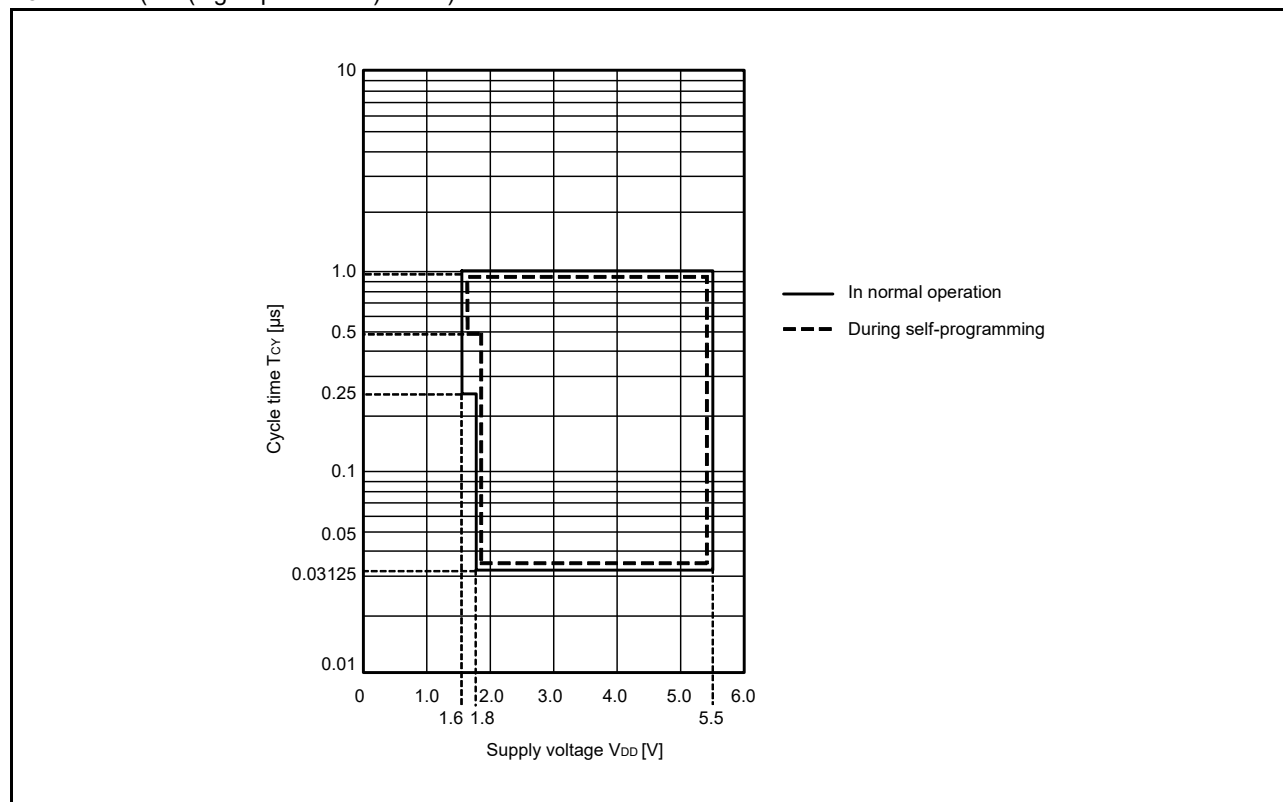
Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Interrupt input high-level width, low-level width	tINTH, tINTL	INTP0 to INTP7	1.6 V ≤ VDD ≤ 5.5 V	1			μs
Key-input high-level width, low-level width	tKRH, tKRL	KR0 to KR7	1.8 V ≤ VDD ≤ 5.5 V	250			ns
			1.6 V ≤ VDD < 1.8 V	1			μs
$\overline{\text{RESET}}$ low-level width	tRSL			10			μs

Remark fMCK: Timer array unit operating clock frequency

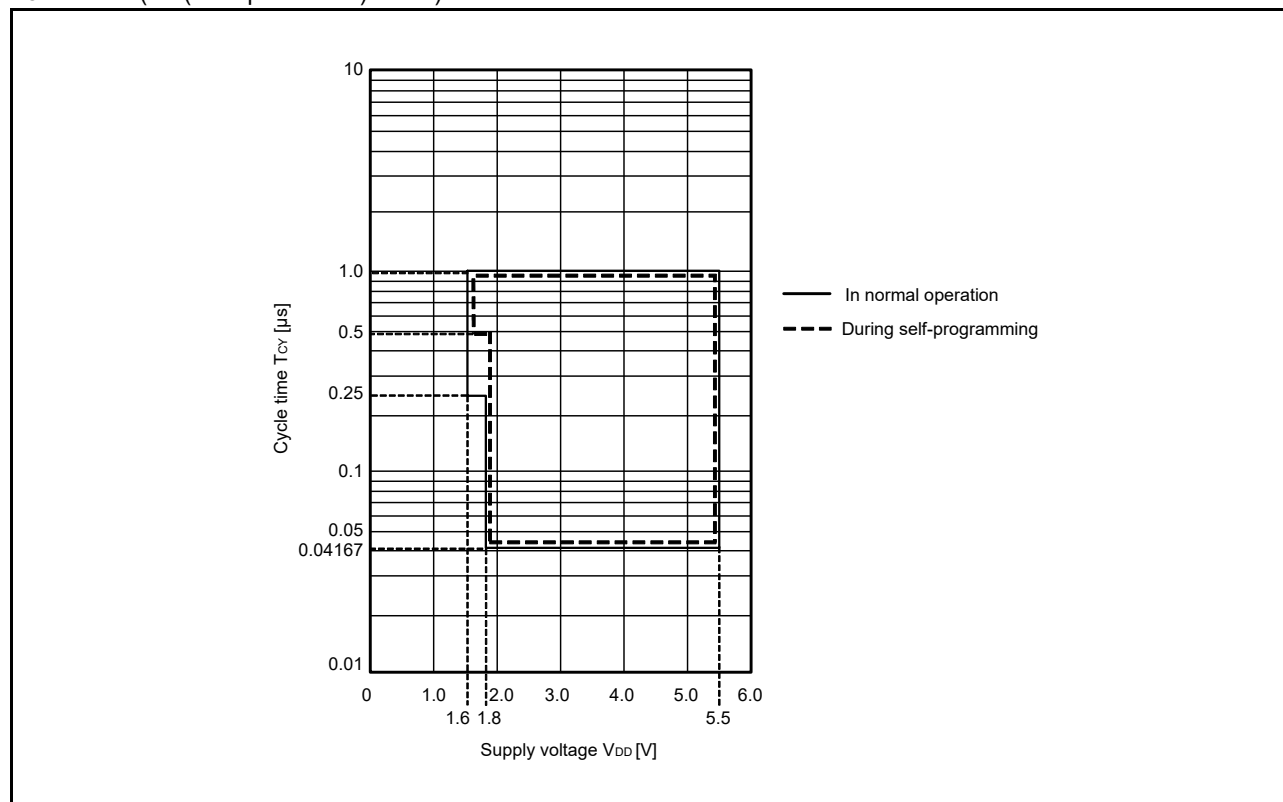
To set this operating clock, use the CKSmn0 and CKSmn1 bits of the timer mode register mn (TMRmn) (m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)).

Minimum Instruction Execution Time during Main System Clock Operation

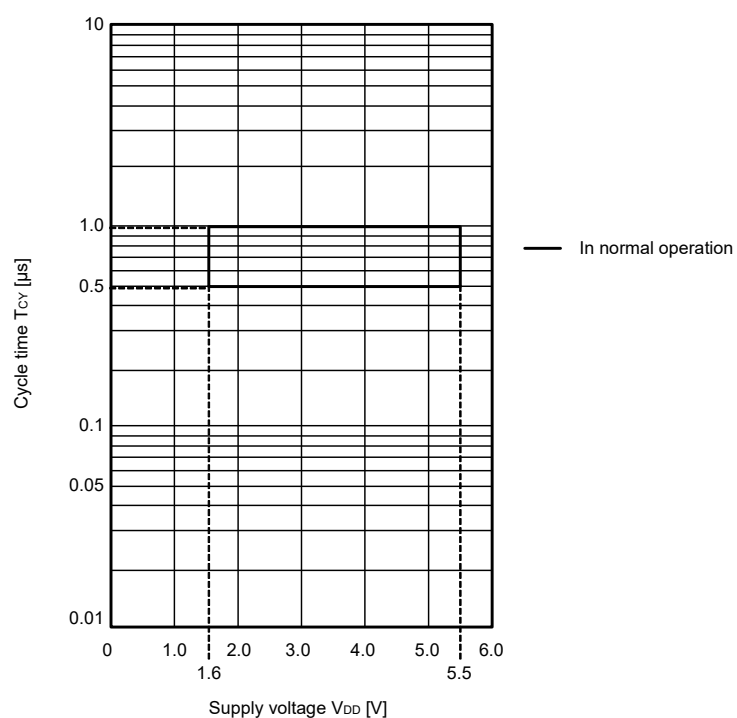
TCY vs VDD (HS (high-speed main) mode)



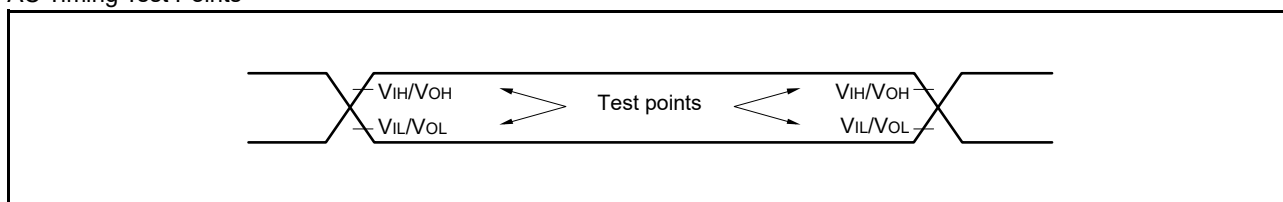
TCY vs VDD (LS (low-speed main) mode)



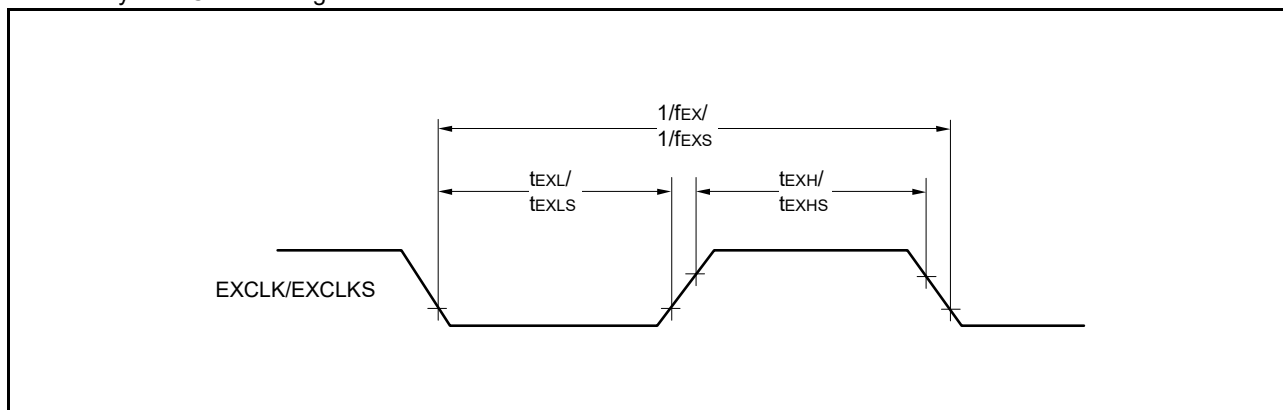
TCY vs VDD (LP (low-power main) mode)



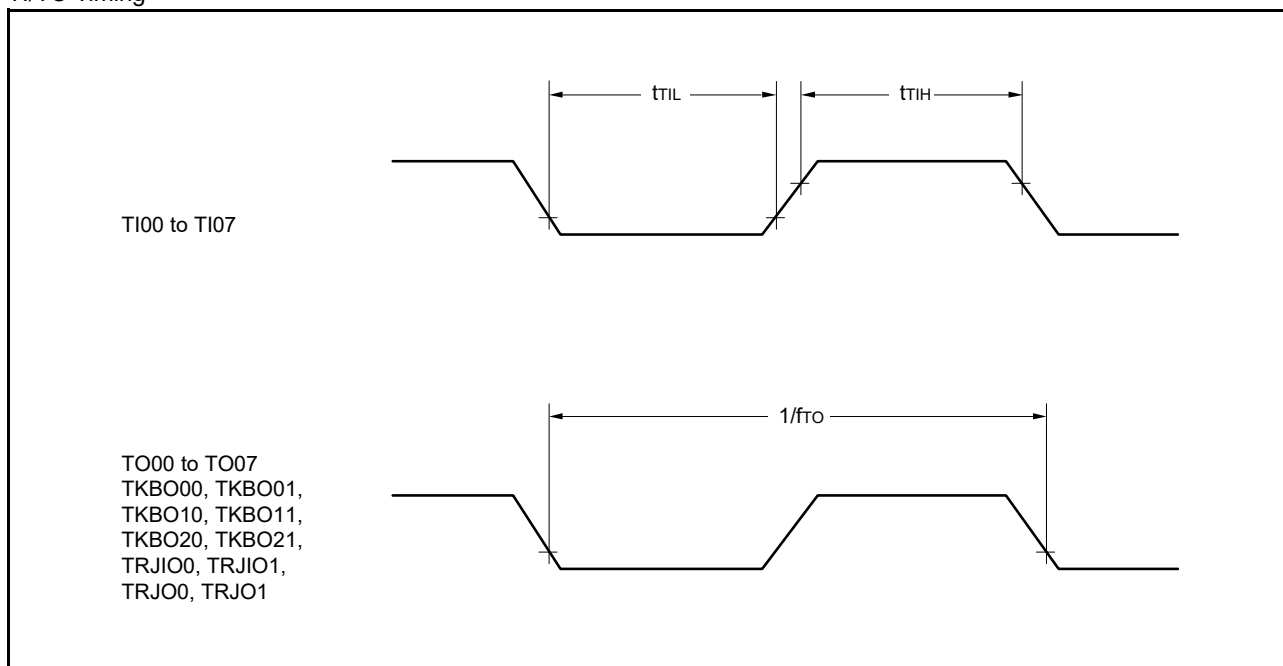
AC Timing Test Points



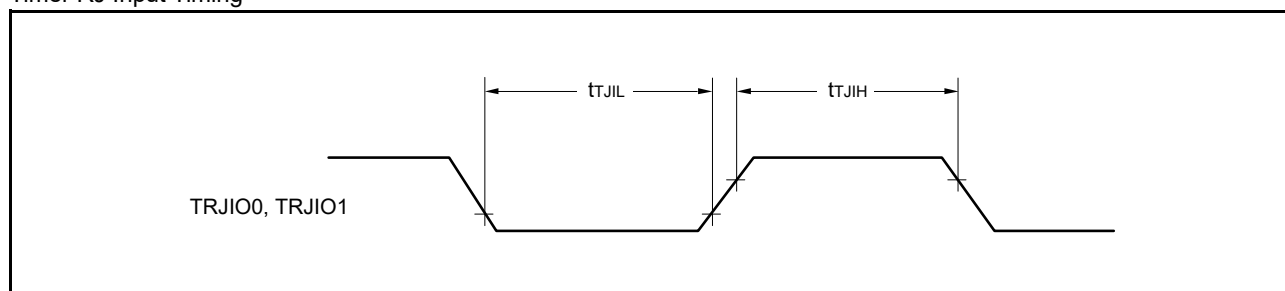
External System Clock Timing



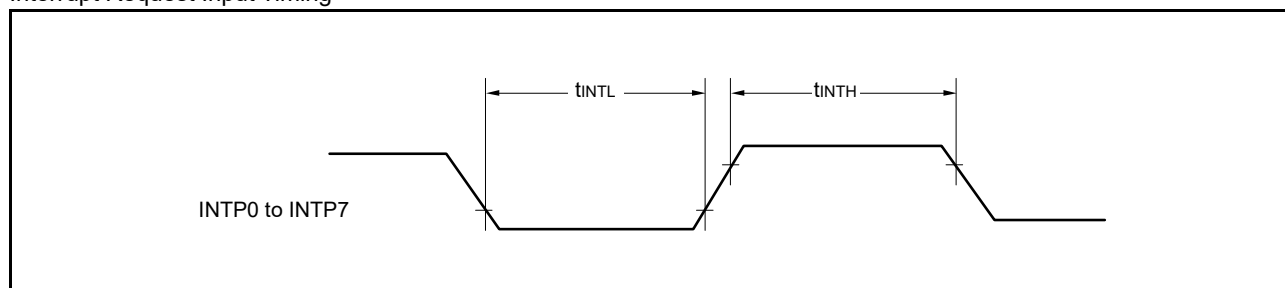
TI/TO Timing



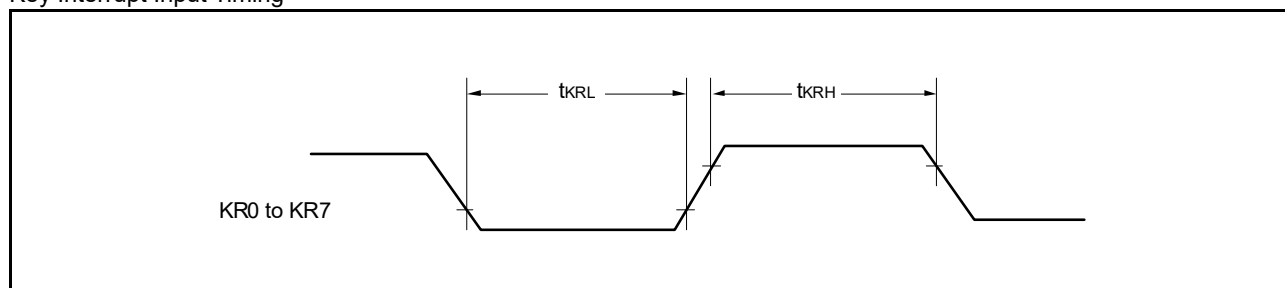
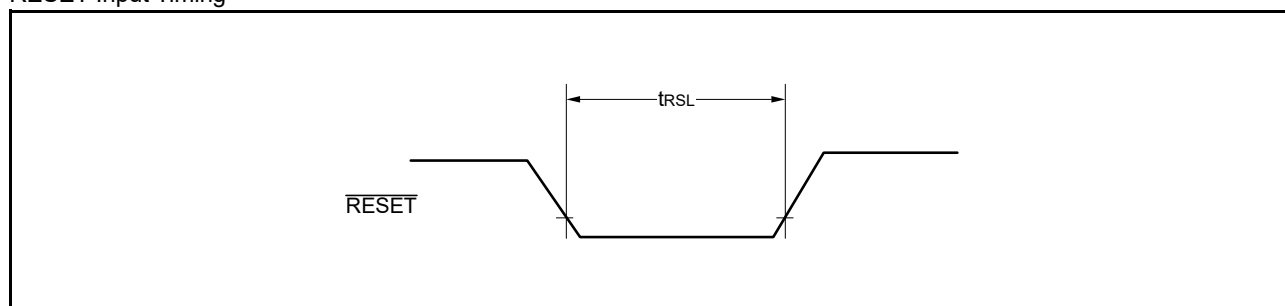
Timer RJ Input Timing



Interrupt Request Input Timing

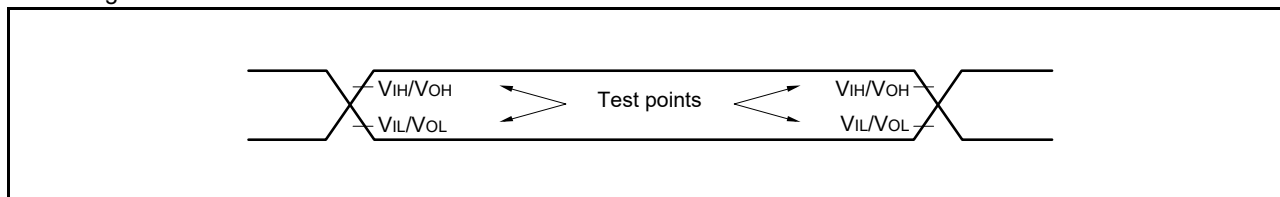


Key Interrupt Input Timing

 $\overline{\text{RESET}}$ Input Timing

42.5 Characteristics of the Peripheral Functions

AC Timing Test Points



42.5.1 Serial array unit

1. In UART communications with devices operating at same voltage levels

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

Item	Symbols	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		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		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 ($1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)

4 MHz ($1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)

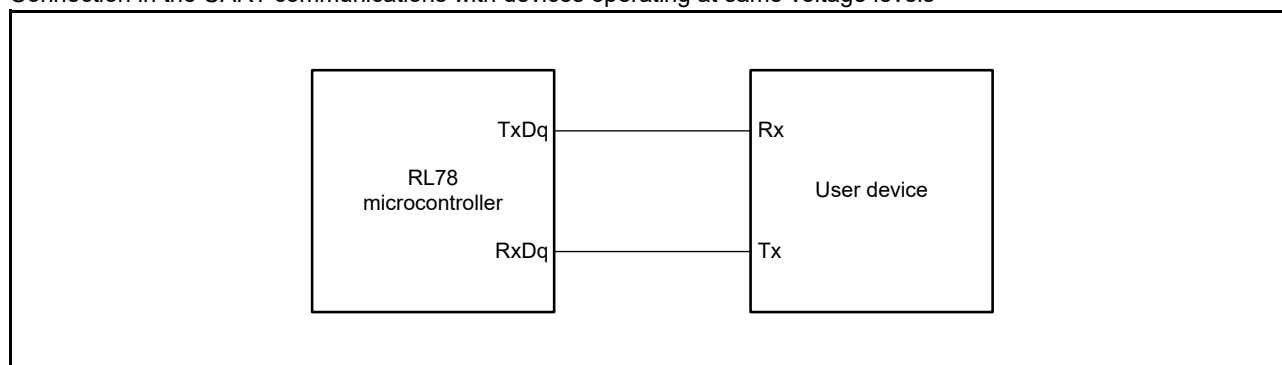
LS (low-speed main) mode : 24 MHz ($1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)

4 MHz ($1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)

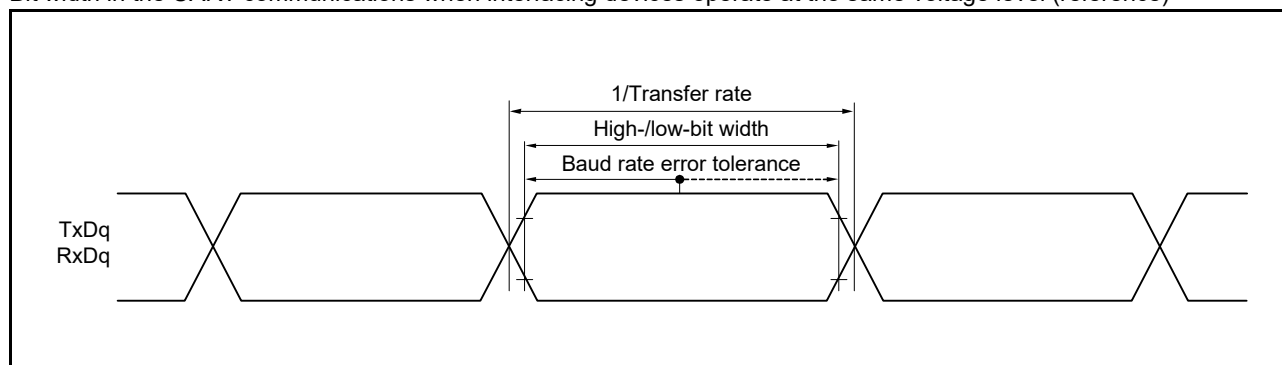
LP (low-power main) mode : 2 MHz ($1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$)

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).

Connection in the UART communications with devices operating at same voltage levels



Bit width in the UART communications when interfacing devices operate at the same voltage level (reference)



Remark 1. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 9, 12 to 14)

Remark 2. fMCK: 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 to 03, 10 to 13)).

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 = EVSS = 0 V)

Item	Symbols	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-level width, 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 the rising of 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 the rising of SCKp Note 1	tKS1	2.7 V ≤ VDD ≤ 5.5 V		10		10		10		ns
Delay time from the falling of SCKp to an SOp output Note 2	tKSO1	2.7 V ≤ VDD ≤ 5.5 V, C = 20 pF ^{Note 3}			10		10		10	ns

Note 1. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the setup time to or from the falling of SCKp.

Note 2. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the delay time to the rising of SCKp.

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), g: PIM and POM number (g = 1), m: Unit number (m = 0), n: Channel number (n = 0)

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

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

Item	Symbols	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	$tkCY1 \geq 4/f_{CLK}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	125		166		2000	ns
			$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	250		250		2000	ns
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	500		500		2000	ns
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1000		1000		2000	ns
SCKp high-level width, low-level width	tkH1, tkL1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	tkCY1/2 -12		tkCY1/2 -21		tkCY1/2 -50		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	tkCY1/2 -18		tkCY1/2 -25		tkCY1/2 -50		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	tkCY1/2 -38		tkCY1/2 -38		tkCY1/2 -50		ns
		$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	tkCY1/2 -50		tkCY1/2 -50		tkCY1/2 -50		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	tkCY1/2 -100		tkCY1/2 -100		tkCY1/2 -100		ns
Slp setup time to the rising of SCKp Note 1	tsIK1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	44		54		110		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	44		54		110		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	75		75		110		ns
		$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	110		110		110		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	220		220		220		ns
Slp hold time from the rising of SCKp Note 1	tkSI1	$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	19		19		19		ns
Delay time from the falling of SCKp to an SOP output Note 2	tkSO1	$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C = 30\text{ pF}$ Note 3		25		25		25	ns

Note 1. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the setup time to or from the falling of SCKp.

Note 2. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the delay time to the rising of SCKp.

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, 10, 11, 20, 21, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14),
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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 = 00 to 03, 10 to 13).

4. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the SCKp external clock

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(1/2)

Item	Symbols	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	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$20\text{ MHz} < f_{MCK}$	8/fMCK		8/fMCK		—		ns
			$f_{MCK} \leq 20\text{ MHz}$	6/fMCK		6/fMCK		6/fMCK		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$16\text{ MHz} < f_{MCK}$	8/fMCK		8/fMCK		—		ns
			$f_{MCK} \leq 16\text{ MHz}$	6/fMCK		6/fMCK		6/fMCK		ns
		$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		6/fMCK and 500		6/fMCK and 500		6/fMCK and 500		ns
		$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		6/fMCK and 750		6/fMCK and 750		6/fMCK and 750		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		6/fMCK and 1500		6/fMCK and 1500		6/fMCK and 1500		ns
SCKp high-level width, low-level width	tkH2, tkL2	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		tkCY2/2 -7		tkCY2/2 -7		tkCY2/2 -7		ns
		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		tkCY2/2 -8		tkCY2/2 -8		tkCY2/2 -8		ns
		$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		tkCY2/2 -18		tkCY2/2 -18		tkCY2/2 -18		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		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

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(2/2)

Item	Symbols	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 the rising of SCKp Note 2	tsIK2	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1/fMCK +20		1/fMCK +30		1/fMCK +30		ns
		$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1/fMCK +30		1/fMCK +30		1/fMCK +30		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1/fMCK +40		1/fMCK +40		1/fMCK +40		ns
Slp hold time from the rising of SCKp Note 2	tkSI2	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1/fMCK +31		1/fMCK +31		1/fMCK +31		ns
		$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		1/fMCK +250		1/fMCK +250		1/fMCK +250		ns
Delay time from the falling of SCKp to an SOp output Note 3	tkSO2	C = 30 pF Note 4	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2/fMCK +44		2/fMCK +110		2/fMCK +110	ns
			$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2/fMCK +75		2/fMCK +110		2/fMCK +110	ns
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2/fMCK +110		2/fMCK +110		2/fMCK +110	ns
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		2/fMCK +220		2/fMCK +220		2/fMCK +220	ns

Note 1. Transfer rate in the SNOOZE mode is 1 Mbps at the maximum.

Note 2. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the setup time to or from the falling of SCKp.

Note 3. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the delay time to the rising of SCKp.

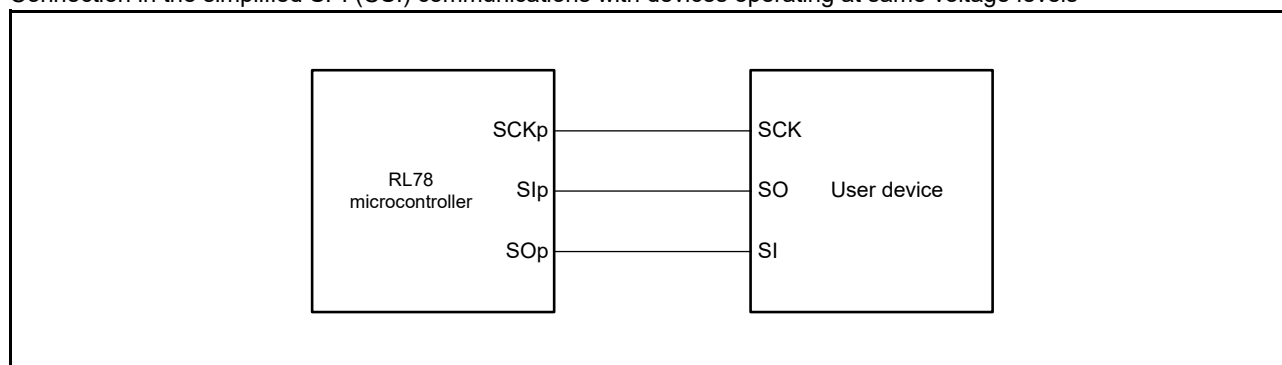
Note 4. C is the load capacitance of the SOp output line.

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).

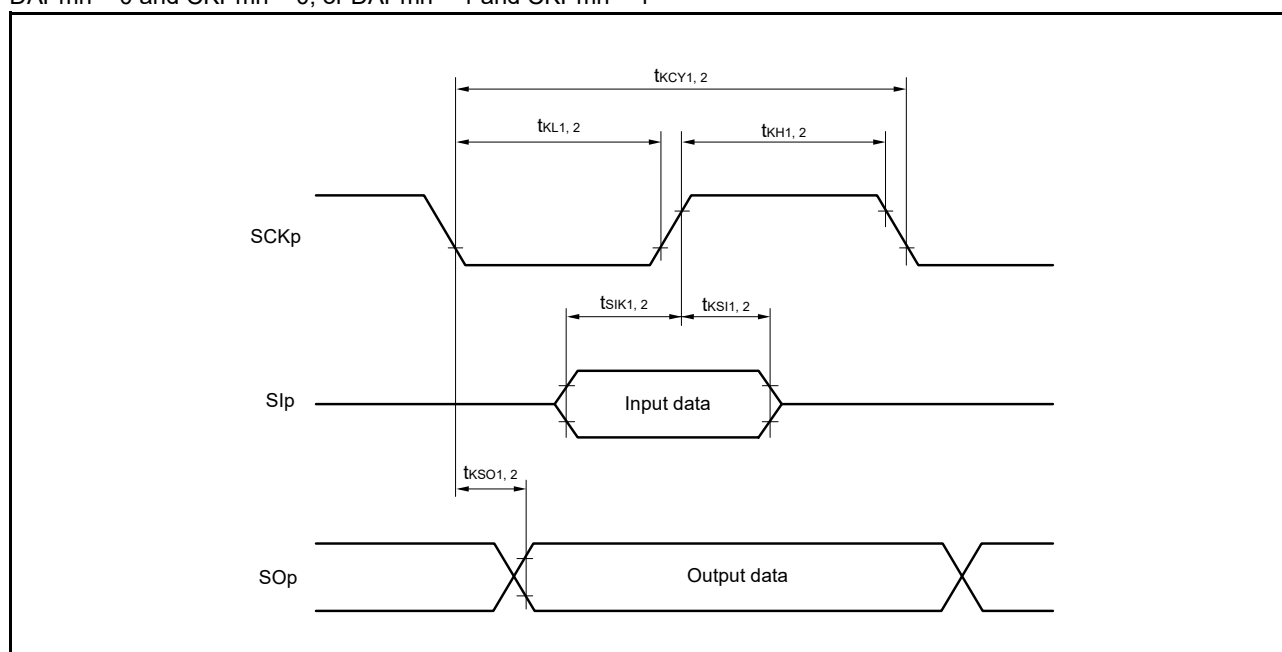
Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14),
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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 = 00 to 03, 10 to 13).

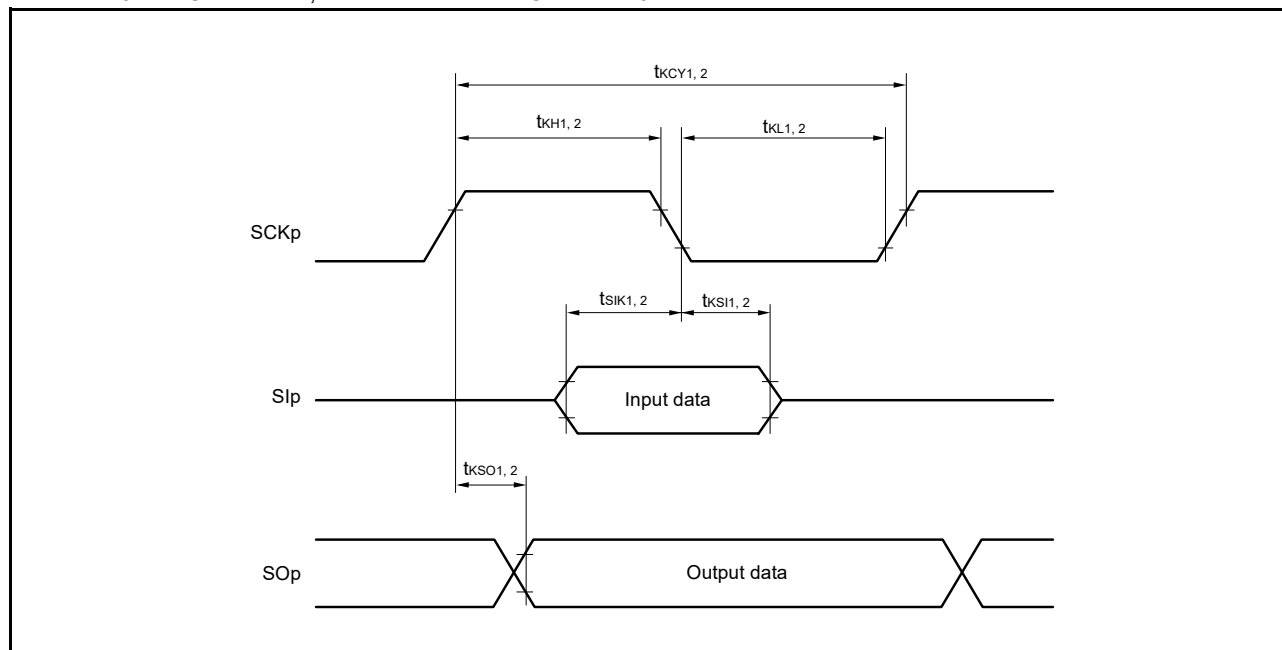
Connection in the simplified SPI (CSI) communications with devices operating at same voltage levels



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



Timing of serial transfer in the simplified SPI (CSI) communications with devices operating at same voltage levels when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0



Remark p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

5. In simplified I²C communications with devices operating at same voltage levels

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

(1/2)

Item	Symbols	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	fSCL	2.7 V ≤ VDD ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ		1000 Note 1		1000 Note 1		400Note 1	kHz
		1.8 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		400Note 1		400Note 1		400Note 1	kHz
		1.8 V ≤ VDD < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		300Note 1		300Note 1		300Note 1	kHz
		1.6 V ≤ VDD < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		250Note 1		250Note 1		250Note 1	kHz
Hold time when SCLr is low	tLOW	2.7 V ≤ VDD ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ VDD < 2.7 V, Cb = 100 pF, Rb = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ VDD < 1.8 V, Cb = 100 pF, Rb = 5 kΩ	1850		1850		1850		ns
Hold time when SCLr is high	tHIGH	2.7 V ≤ VDD ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ VDD < 2.7 V, Cb = 100 pF, Rb = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ VDD < 1.8 V, Cb = 100 pF, Rb = 5 kΩ	1850		1850		1850		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1729.)

5. In simplified I²C communications with devices operating at same voltage levels(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

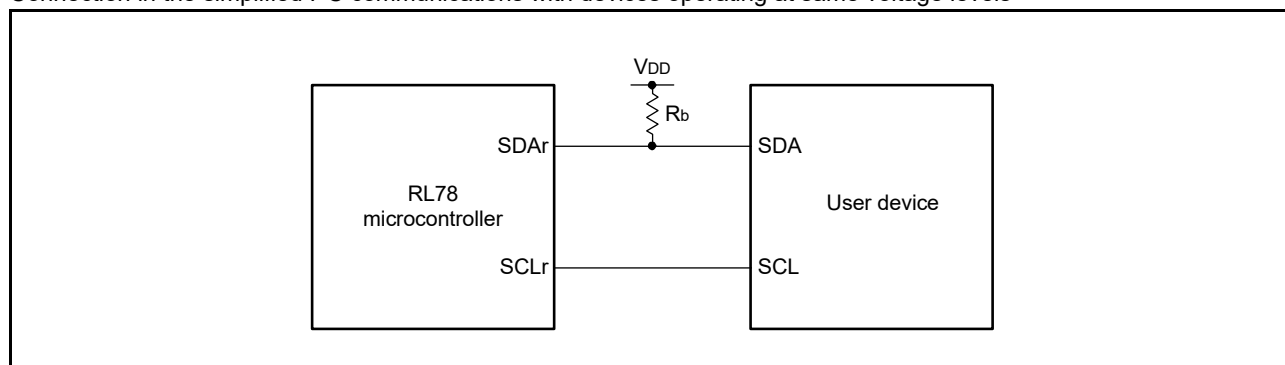
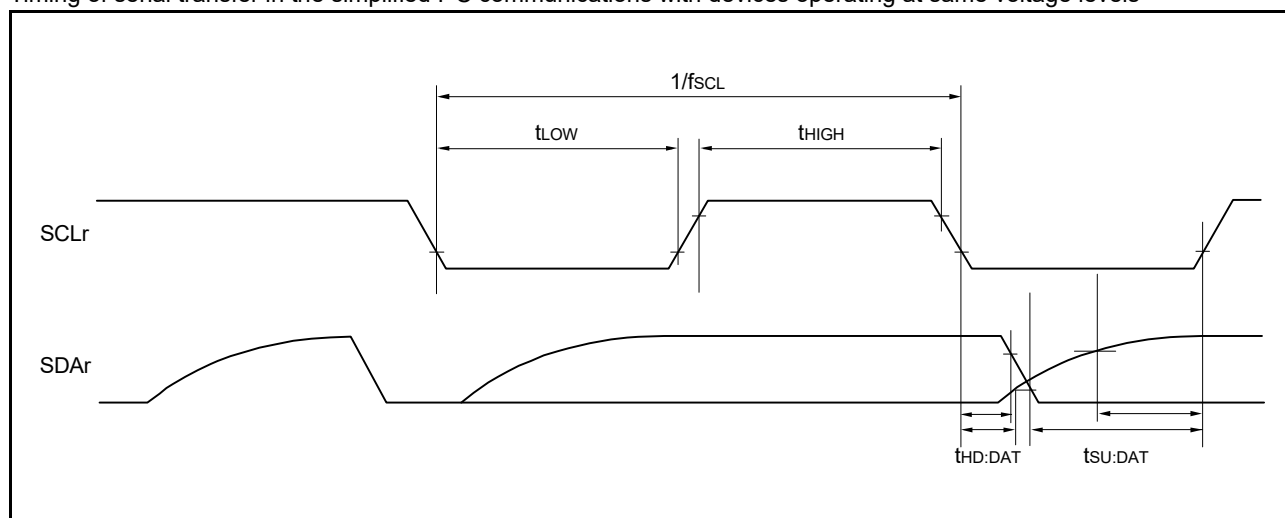
(2/2)

Item	Symbols	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 N-ch open-drain output (withstand voltage of V_{DD}) mode for the SDAr pin, and normal output mode for the SCLr pin by using the port input mode register g (PIMg) and port output mode register h (POMh).

(Remarks are listed on the next page.)

Connection in the simplified I²C communications with devices operating at same voltage levelsTiming of serial transfer in the simplified I²C communications with devices operating at same voltage levels

Remark 1. R_b[Ω]: 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, 30, 31), g: PIM number (g = 0 to 9, 12, 14), h: POM number (h = 0 to 9, 12 to 14)

Remark 3. f_{MCK}: Operation clock frequency of the serial array unit

To set this operating clock, use the CKSmn bit in the SMRmn register (m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)).

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 = EVSS = 0 V)

(1/2)

Item	Symbols	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
			Theoretical value of the maximum transfer rate fMCK = fCLKNote 2			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
			Theoretical value of the maximum transfer rate fMCK = fCLKNote 2			5.3		4	Mbps
			1.8 V ≤ VDD < 3.3 V, 1.6 V ≤ Vb ≤ 2.0 V			fMCK/6 Notes 1, 3		fMCK/6 Notes 1, 3	bps
			Theoretical value of the maximum transfer rate fMCK = fCLKNote 2			5.3		4	Mbps

Note 1. 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:

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)

Note 3. Use this rate with EVDD0 ≥ Vb.

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 the port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with the TTL input buffer selected.

Remark 1. Vb[V]: Communication line voltage**Remark 2.** q: UART number (q = 0 to 3), g: PIM and POM number (g = 0 to 9, 12 to 14)**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 = 00 to 03, 10 to 13).

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 = EVSS = 0 V)

(2/2)

Item	Symbols	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 and 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 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ and $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

* 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 $V_{DD} \geq V_b$.

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 $1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$ and $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

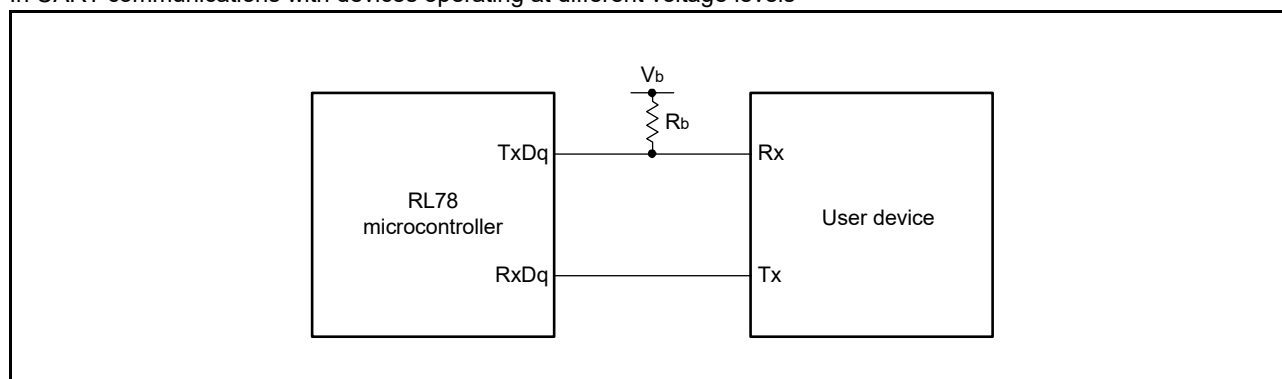
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

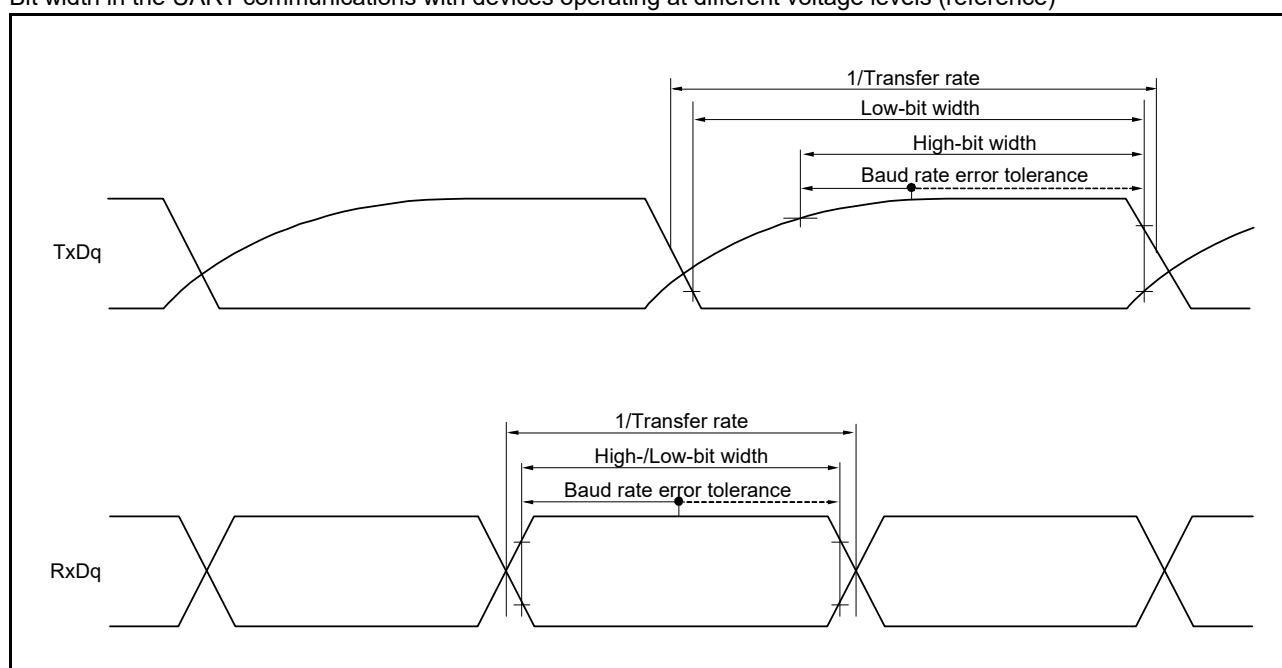
Note 7. This rate is calculated as an example when the conditions described in the Conditions column are met. See **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin, and the N-ch open-drain output (withstand voltage of V_{DD}) mode for the TxDq pin by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL} , see the DC characteristics with the TTL input buffer selected.

In UART communications with devices operating at different voltage levels



Bit width in the UART communications with devices operating at different voltage levels (reference)



Remark 1. $R_b[\Omega]$: 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 3), g: PIM and POM number (g = 0 to 9, 12 to 14)

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 = 00 to 03, 10 to 13).

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)

($T_A = -40$ to $+105^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(1/2)

Item	Symbols	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 $\geq 2/f_{CLK}$ 4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω	200		200		2300		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω	300		300		2300		ns
SCKp high-level width	tkH1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω	tkCY1/2 -50		tkCY1/2 -50		tkCY1/2 -50		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω	tkCY1/2 -120		tkCY1/2 -120		tkCY1/2 -120		ns
SCKp low-level width	tkL1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω	tkCY1/2 -7		tkCY1/2 -7		tkCY1/2 -50		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω	tkCY1/2 -10		tkCY1/2 -10		tkCY1/2 -50		ns
Slp setup time to the rising of SCKp Note 1	tSIK1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω	58		58		479		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω	121		121		479		ns
Slp hold time from the rising of SCKp Note 1	tKSI1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω	10		10		10		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω	10		10		10		ns
Delay time from the falling of SCKp to an SOp output Note 1	tkSO1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 20 pF, Rb = 1.4 k Ω		60		60		60	ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 20 pF, Rb = 2.7 k Ω		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)

($T_A = -40$ to $+105^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(2/2)

Item	Symbols	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 the falling of SCKp ^{Note 2}	tSIK1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	23		23		110		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	33		33		110		ns
Slp hold time from the falling of SCKp ^{Note 2}	tKSI1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	10		10		10		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	10		10		10		ns
Delay time from the rising of SCKp to an SOp output ^{Note 2}	tKSO1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 1.4\text{ k}\Omega$		10		10		10	ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 20\text{ pF}$, $R_b = 2.7\text{ k}\Omega$		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 V_{DD}) mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL} , see the DC characteristics with the TTL input buffer selected.

Remark 1. $R_b[\Omega]$: Communication line (SCKp, SOp) pull-up resistance, $C_b[\text{F}]$: Communication line (SCKp, SOp) load capacitance, $V_b[\text{V}]$: Communication line voltage

Remark 2. p: CSI number (p = 00), g: PIM and POM number (g = 1), m: Unit number (m = 0), n: Channel number (n = 0)

Remark 3. fMCK: Serial array unit operation clock frequency

To set this operating clock, use the CKS_{mn} bit in the serial mode register mn (SMR_{mn}) (m: Unit number, n: Channel number = 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

($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(1/3)

Item	Symbols	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 $\geq 4/f_{CLK}$ 4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 30 pF, Rb = 1.4 k Ω		300		300		2300		ns
			2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 30 pF, Rb = 2.7 k Ω	500		500		2300		ns
			1.8 V $\leq V_{DD} < 3.3\text{ V}$, 1.6 V $\leq V_b \leq 2.0\text{ V}$ Note , Cb = 30 pF, Rb = 5.5 k Ω	1150		1150		2300		ns
SCKp high-level width	tkH1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 30 pF, Rb = 1.4 k Ω		tkCY1/2 -75		tkCY1/2 -75		tkCY1/2 -75		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 30 pF, Rb = 2.7 k Ω		tkCY1/2 -170		tkCY1/2 -170		tkCY1/2 -170		ns
		1.8 V $\leq V_{DD} < 3.3\text{ V}$, 1.6 V $\leq V_b \leq 2.0\text{ V}$ Note , Cb = 30 pF, Rb = 5.5 k Ω		tkCY1/2 -458		tkCY1/2 -458		tkCY1/2 -458		ns
SCKp low-level width	tkL1	4.0 V $\leq V_{DD} \leq 5.5\text{ V}$, 2.7 V $\leq V_b \leq 4.0\text{ V}$, Cb = 30 pF, Rb = 1.4 k Ω		tkCY1/2 -12		tkCY1/2 -12		tkCY1/2 -50		ns
		2.7 V $\leq V_{DD} < 4.0\text{ V}$, 2.3 V $\leq V_b \leq 2.7\text{ V}$, Cb = 30 pF, Rb = 2.7 k Ω		tkCY1/2 -18		tkCY1/2 -18		tkCY1/2 -50		ns
		1.8 V $\leq V_{DD} < 3.3\text{ V}$, 1.6 V $\leq V_b \leq 2.0\text{ V}$ Note , Cb = 30 pF, Rb = 5.5 k Ω		tkCY1/2 -50		tkCY1/2 -50		tkCY1/2 -50		ns

Note Use this rate with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin, and the N-ch open-drain output (withstand voltage of V_{DD}) mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL} , see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on page 1739.)

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

($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(2/3)

Item	Symbols	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 the rising of SCKp ^{Note 1}	tSIK1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	81		81		479		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	177		177		479		ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$	479		479		479		ns
Slp hold time from the rising of SCKp ^{Note 1}	tKSI1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	19		19		19		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	19		19		19		ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$	19		19		19		ns
Delay time from the falling of SCKp to an SOp output ^{Note 1}	tKSO1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$		100		100		100	ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$		195		195		195	ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$		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 value with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the Slp pin, and the N-ch open-drain output (withstand voltage of V_{DD}) mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on page 1739.)

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

($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

(3/3)

Item	Symbols	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 the falling of SCKp ^{Note 1}	tSIK1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	44		44		110		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	44		44		110		ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$	110		110		110		ns
Slp hold time from the falling of SCKp ^{Note 1}	tKSI1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$	19		19		19		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$	19		19		19		ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$	19		19		19		ns
Delay time from the rising of SCKp to an SOp output ^{Note 1}	tKSO1	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 1.4\text{ k}\Omega$		25		25		25	ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$, $C_b = 30\text{ pF}$, $R_b = 2.7\text{ k}\Omega$		25		25		25	ns
		$1.8\text{ V} \leq V_{DD} < 3.3\text{ V}$, $1.6\text{ V} \leq V_b \leq 2.0\text{ V}$ ^{Note 2} , $C_b = 30\text{ pF}$, $R_b = 5.5\text{ k}\Omega$		25		25		25	ns

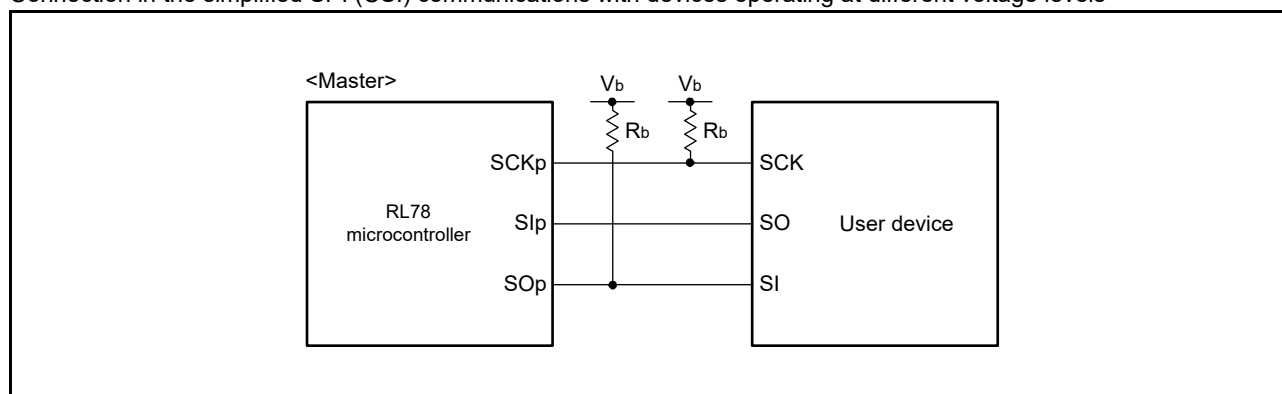
Note 1. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. Use this value with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the Slp pin, and the N-ch open-drain output (withstand voltage of V_{DD}) mode for the SOp and SCKp pins by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL} , see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



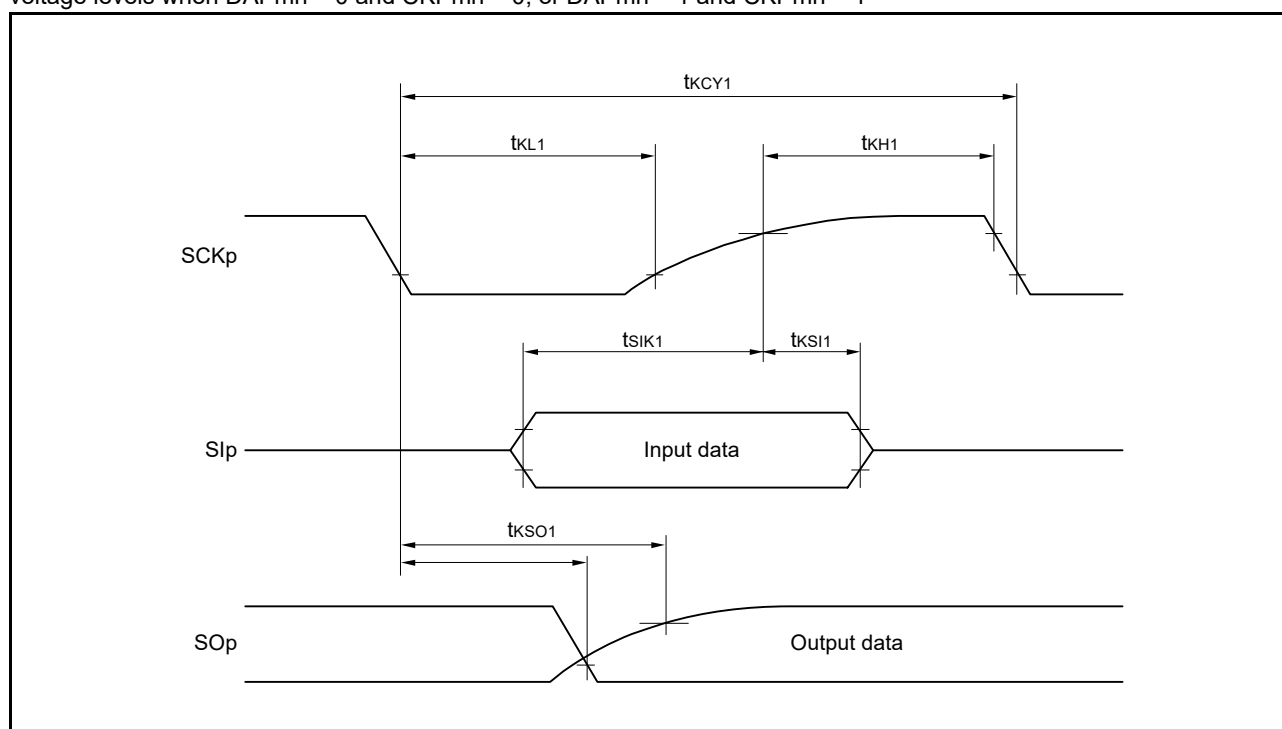
Remark 1. $R_b[\Omega]$: Communication line (SCKp, SOp) pull-up resistance, $C_b[F]$: Communication line (SCKp, SOp) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14), m: Unit number, n: Channel number (mn = 00 to 02, 10, 12, 13)

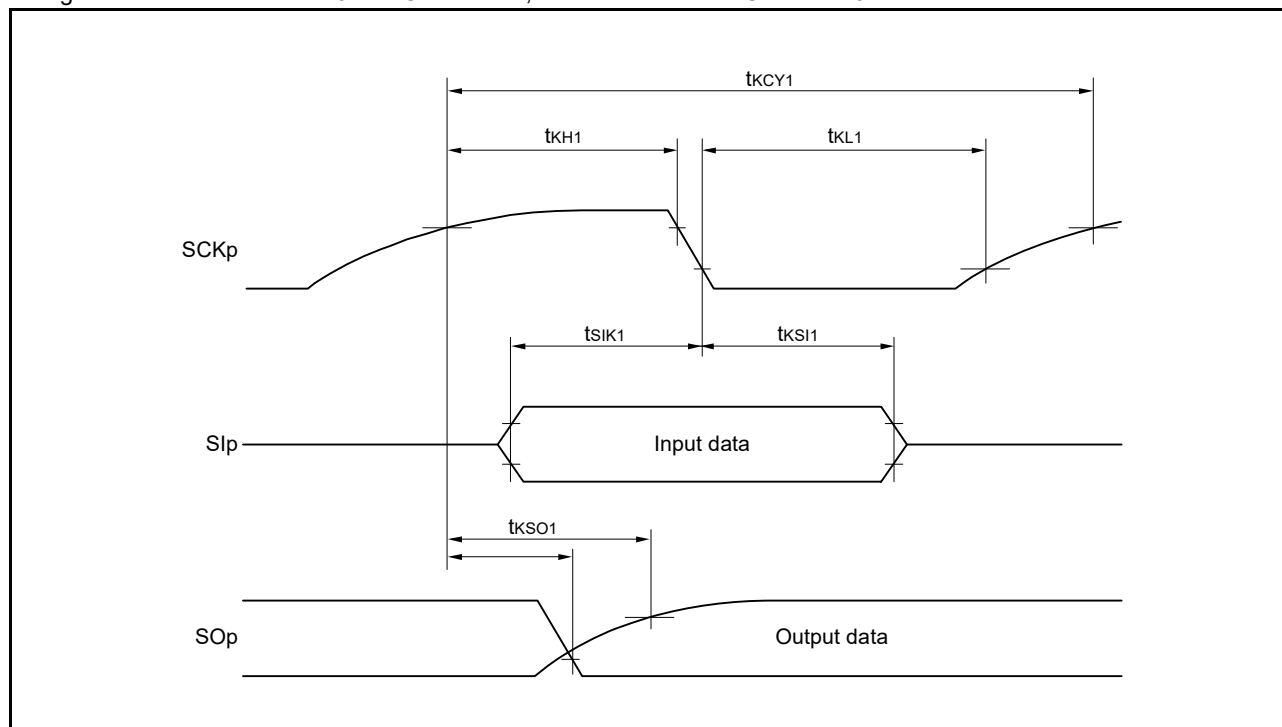
Remark 3. f_{MCK} : Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn) (m: Unit number, n: Channel number = 00).

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 $CKPmn = 1$



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 = 1, or DAPmn = 1 and CKPmn = 0



Remark p: CSI number (p = 00, 01, 10, 20, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14), m: Unit number, n: Channel number (mn = 00 to 02, 10, 12, 13)

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 = EVSS = 0 V)

(1/2)

Item	Symbols	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 page 1743.)

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 = EVSS = 0 V)

(2/2)

Item	Symbols	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 the rising of 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 the rising of SCKp Note 3	tkSI2		1/fMCK +31		1/fMCK +31		1/fMCK +31		ns
Delay time from the falling of SCKp to an 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 rate with VDD ≥ Vb.

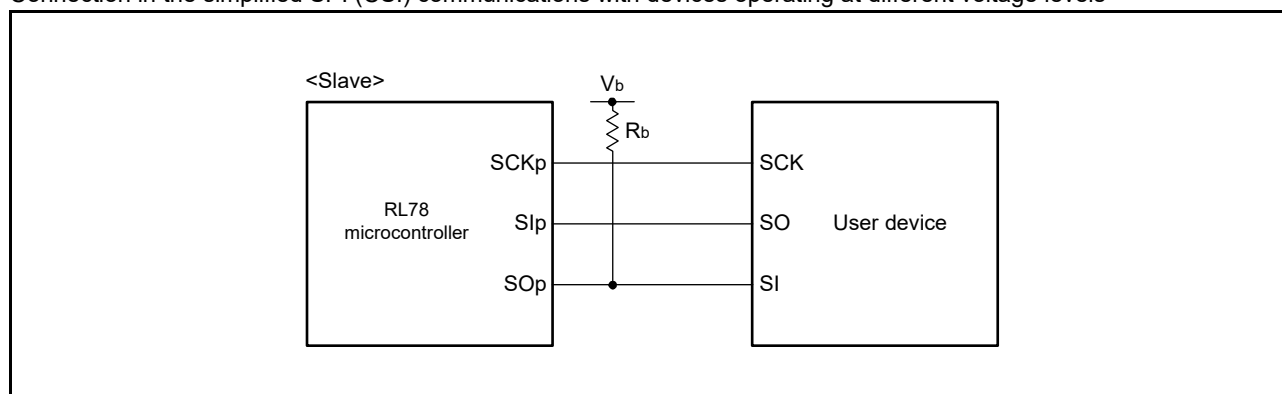
Note 3. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the setup time to or from the falling of SCKp.

Note 4. This is for when the settings are DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. When the settings are DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0, this is the delay time to the rising of SCKp.

Caution Select the TTL input buffer for the Slp and SCKp pins, and the N-ch open-drain output (withstand voltage of VDD) mode for the SOp pin by using the port input mode register g (PIMg) and port output mode register g (POMg). For VIH and VIL, see the DC characteristics with the TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



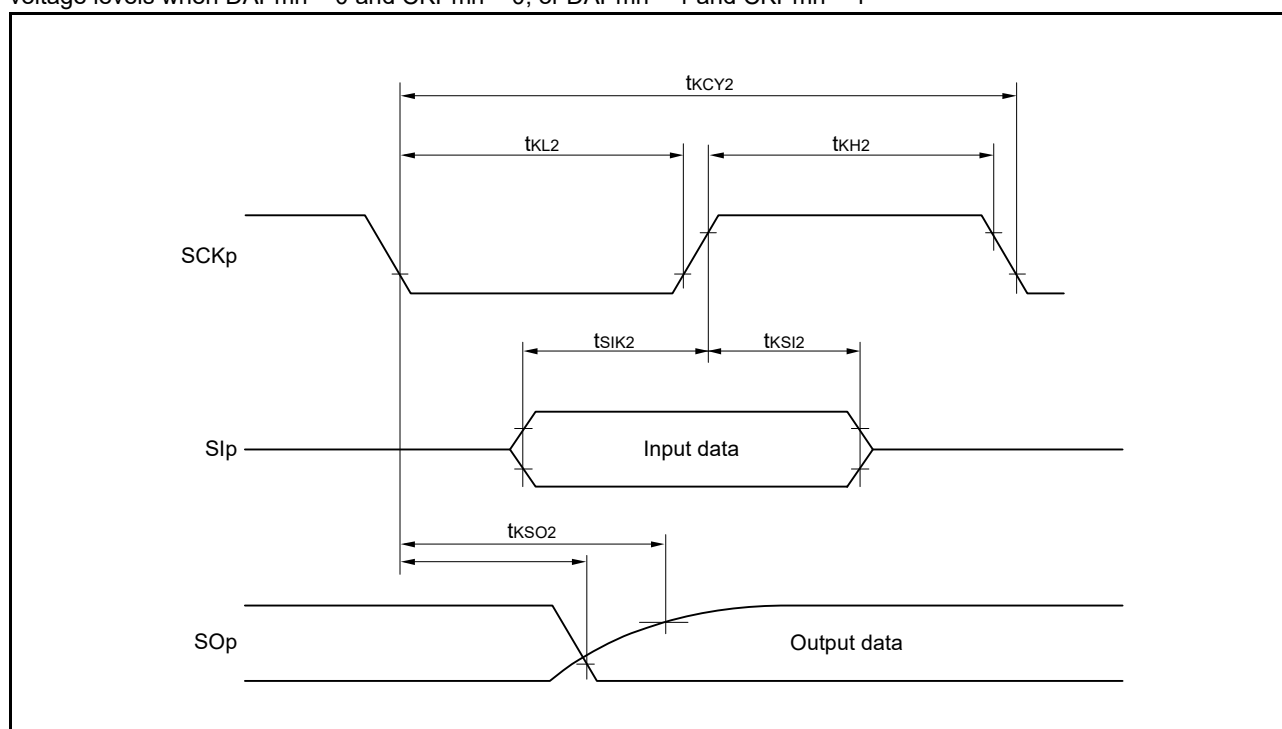
Remark 1. $R_b[\Omega]$: Communication line (SO_p) pull-up resistance, $C_b[F]$: Communication line (SO_p) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14), m: Unit number, n: Channel number (mn = 00 to 02, 10, 12, 13)

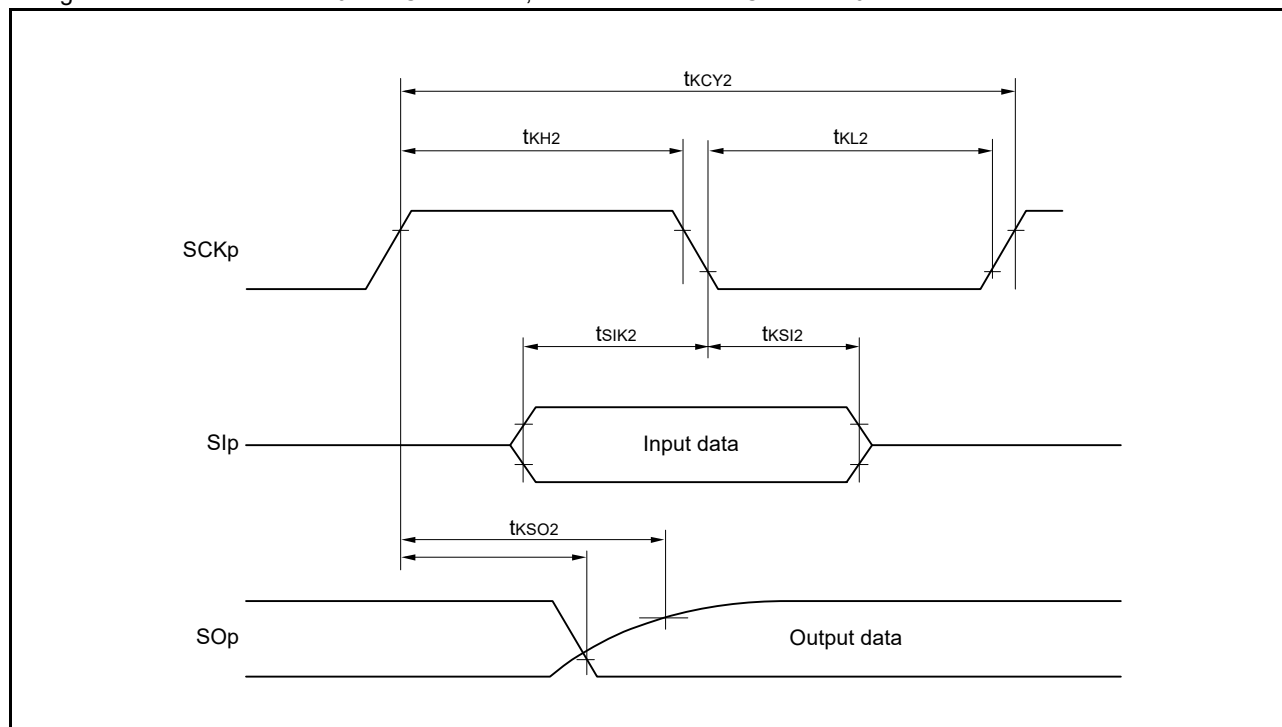
Remark 3. f_{MCK}: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn) (m: Unit number, n: Channel number = 00, 01, 02, 10, 12 and 13).

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



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 = 1, or DAPmn = 1 and CKPmn = 0



Remark p: CSI number (p = 00, 01, 10, 20, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14), m: Unit number,
n: Channel number (mn = 00 to 02, 10, 12, 13)

10. Simplified I²C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V)(T_A = -40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

(1/2)

Item	Symbols	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	fSCL	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	tLOW	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	tHIGH	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)(T_A = -40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

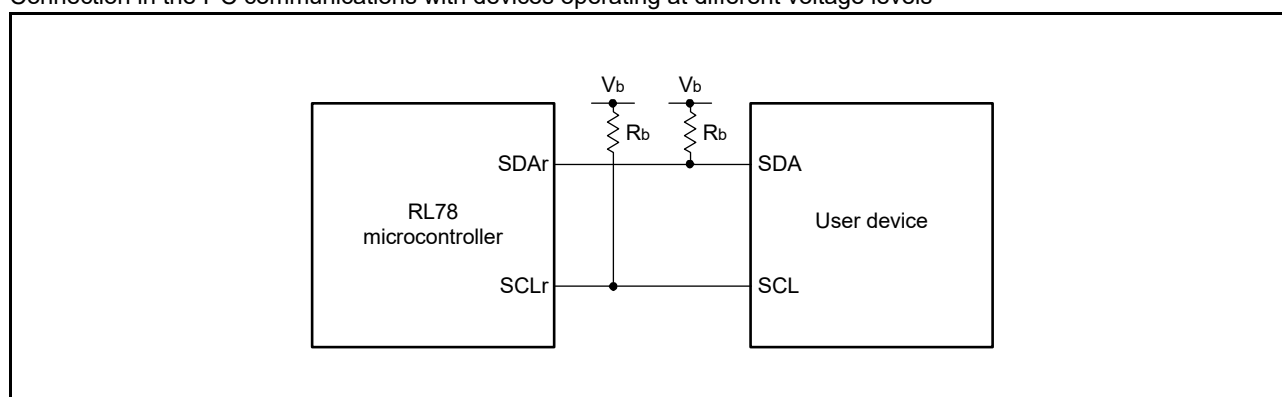
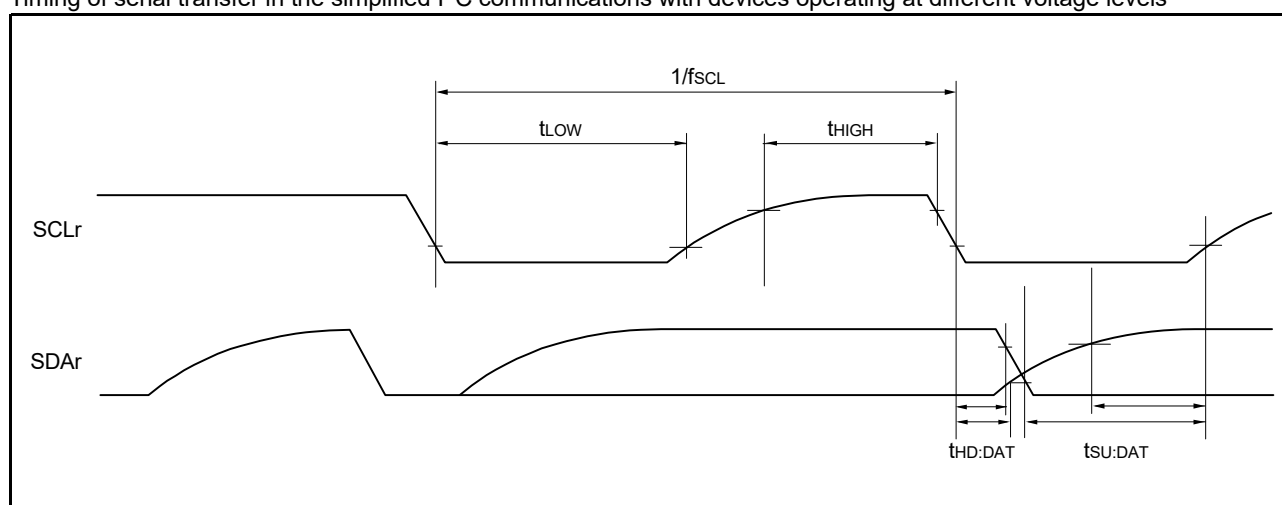
(2/2)

Item	Symbols	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 ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 2.7 kΩ	1/f _{MCK} +135 Note 3		1/f _{MCK} +135 Note 3		1/f _{MCK} +190 Note 3		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	1/f _{MCK} +135 Note 3		1/f _{MCK} +135 Note 3		1/f _{MCK} +190 Note 3		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	1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		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	1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		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	1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		1/f _{MCK} +190 Note 3		ns
Data hold time (transmission)	thd:DAT	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	0	305	0	305	0	305	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	0	305	0	305	0	305	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	0	355	0	355	0	355	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	0	355	0	355	0	355	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	0	405	0	405	0	405	ns

Note 1. The listed times must be no greater than f_{MCK}/4.**Note 2.** Use this rate with V_{DD} ≥ V_b.**Note 3.** Set f_{MCK} so that it will not exceed the hold time when SCLr is low or high.

Caution Select the TTL input buffer and N-ch open-drain output (withstand voltage of V_{DD}) mode for the SDAr pin, and the N-ch open-drain output (withstand voltage of V_{DD}) mode for the SCLr pin by using the port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the I²C communications with devices operating at different voltage levelsTiming 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, 10, 20, 30, 31), g: PIM and POM number (g = 0 to 9, 12 to 14)

Remark 3. f_{MCK} : Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the SMRmn register (m: Unit number, n: Channel number (mn = 00 to 02, 10, 12, 13)).

42.5.2 Serial interface UARTA

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Transfer rate			200	0	153600	bps

Caution Select the normal input buffer for the RxDAq pin, and the normal output mode for the TxDAq pin by using the port input mode register g (PIMg) and port output mode register h (POMh).

Remark 1. g: PIM number (g = 1, 2, 4, 5, 8, 14), h: POM number (h= 0 to 2, 4 to 6, 8, 13, 14)

Remark 2. q = 0 to 3

42.5.3 Serial interface IICA

1. I²C standard mode

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	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.

Remark 1. 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

Remark 2. n = 0, 1

2. I²C fast mode(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
SCLAn 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 SCLAn is low	t _{LOW}	1.8 V ≤ V _{DD} ≤ 5.5 V	1.3			μs
Hold time when SCLAn 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.**Remark 1.** The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.C_b = 320 pF, R_b = 1.1 kΩ**Remark 2.** n = 0, 1

3. I²C fast mode plus

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

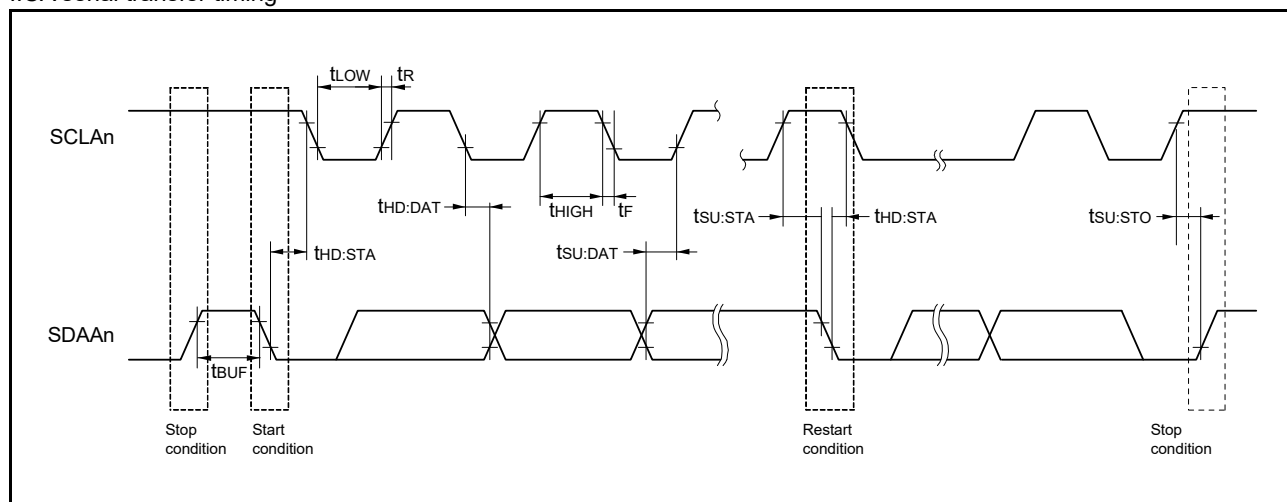
Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
SCLAn clock frequency	fSCL	Fast mode plus: fCLK ≥ 10 MHz 2.7 V ≤ VDD ≤ 5.5 V	0		1000	kHz
Setup time of restart condition	tSU:STA	2.7 V ≤ VDD ≤ 5.5 V	0.26			μs
Hold time ^{Note 1}	tHD:STA	2.7 V ≤ VDD ≤ 5.5 V	0.26			μs
Hold time when SCLAn is low	tLOW	2.7 V ≤ VDD ≤ 5.5 V	0.5			μs
Hold time when SCLAn is high	tHIGH	2.7 V ≤ VDD ≤ 5.5 V	0.26			μs
Data setup time (reception)	tSU:DAT	2.7 V ≤ VDD ≤ 5.5 V	50			ns
Data hold time (transmission) ^{Note 2}	tHD:DAT	2.7 V ≤ VDD ≤ 5.5 V	0		0.45	μs
Setup time of stop condition	tSU:STO	2.7 V ≤ VDD ≤ 5.5 V	0.26			μs
Bus-free time	tBUF	2.7 V ≤ VDD ≤ 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 tHD:DAT applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.**Remark 1.** 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

Remark 2. n = 0, 1

IICA serial transfer timing

**Remark** n = 0, 1

42.6 Analog Characteristics

42.6.1 A/D converter characteristics

1. Normal modes 1 and 2

($T_A = -40$ to $+105^\circ\text{C}$, $2.4\text{ V} \leq \text{AVREFP} \leq \text{VDD} \leq 5.5\text{ V}$, $\text{VSS} = 0\text{ V}$,
reference voltage (+) = AVREFP (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM (ADREFM = 1),
target pins: ANI16 to ANI26)

	Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
	Resolution	RES				12	bit
	Conversion clock	f _{AD}		1		32	MHz
<R>	Overall error ^{Notes 1, 2, 3, 4, 5}	AINL	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±1.8	±4.1	LSB
<R>			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±2.1	±4.2	LSB
<R>			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±2.4	±4.7	LSB
	Conversion time ^{Notes 5, 6}	t _{CONV}	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
<R>	Zero-scale error ^{Notes 1, 2, 3, 4, 5, 7}	E _{ZS}	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.01	±0.06	%FSR
<R>			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.02	±0.09	%FSR
<R>			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.03	±0.11	%FSR
<R>	Full-scale error ^{Notes 1, 2, 3, 4, 5, 7}	E _{FS}	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.02	±0.09	%FSR
<R>			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.03	±0.11	%FSR
<R>			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.03	±0.11	%FSR
	Analog input voltage	V _{AIN}		0		AVREFP	V

Note 1. This value does not include the quantization error ($\pm 1/2$ LSB).

Note 2. The values in the column Max. only apply in the case of a normal distribution with $\pm 3\sigma$ variation from the mean.

Note 3. We do not inspect the characteristics of the A/D converter before shipment. The listed values are only results of evaluation.

Note 4. When $\text{AVREFP} < \text{VDD}$, the maximum values are as follows.

Overall error/zero-scale error/full-scale error: Add ($\pm 0.75\text{ LSB} \times (\text{VDD voltage (V)} - \text{AVREFP voltage (V)})$) to the maximum value.

Integral linearity error: Add ($\pm 0.2\text{ LSB} \times (\text{VDD voltage (V)} - \text{AVREFP voltage (V)})$) to the maximum value.

Note 5. The listed values apply when the conversion resolution is set to 12 bits.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 μs. Accordingly, use standard mode 2 with the longer sampling time.

Note 7. This value is indicated as a ratio (%FSR) to the full-scale value.

($T_A = -40$ to $+105^\circ\text{C}$, $2.4\text{ V} \leq \text{AVREFP} \leq \text{VDD} \leq 5.5\text{ V}$, $\text{VSS} = 0\text{ V}$,
reference voltage (+) = AVREFP (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM (ADREFM = 1),
target pins: internal reference voltage and temperature sensor output voltage)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES			8		12	bit
Conversion clock	fAD			1		32	MHz
Overall error ^{Notes 1, 2, 3, 4}	AINL	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 7.5	LSB
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 9.0	LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 9.0	LSB
Conversion time ^{Note 5}	tCONV	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	2.0			μs
Zero-scale error ^{Notes 1, 2, 3, 4, 6}	Ezs	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.17	%FSR
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.21	%FSR
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.21	%FSR
Full-scale error ^{Notes 1, 2, 3, 4, 6}	Efs	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.17	%FSR
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.21	%FSR
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 0.21	%FSR
Integral linearity error ^{Notes 1, 3, 4}	ILE	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 3.0	LSB
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 3.0	LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$			± 3.0	LSB
Differential linearity error ^{Note 1}	DLE	12-bit resolution	$4.5\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		± 1.0		LSB
			$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		± 1.0		LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		± 1.0		LSB
Analog input voltage	VAIN			0		AVREFP	V

Note 1. This value does not include the quantization error ($\pm 1/2$ LSB).

Note 2. When pins ANI16 to ANI26 are selected as the target pins for conversion or the TSCAP voltage of the CTSU is to be A/D converted, the maximum values are as follows.

Overall error: Add ± 3 LSB to the maximum value.

Zero-scale/full-scale error: Add $\pm 0.04\%$ FSR to the maximum 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.

Zero-scale/full-scale error: Add $\pm 0.25\%$ FSR to the maximum value.

Integral linearity error: Add ± 4 LSB to the maximum value.

Note 4. When AVREFP < VDD, the maximum values are as follows.

Overall error/zero-scale error/full-scale error: Add $(\pm 0.75\text{ LSB} \times (\text{VDD voltage (V)} - \text{AVREFP voltage (V)}))$ to the maximum value.

Integral linearity error: Add $(\pm 0.2\text{ LSB} \times (\text{VDD voltage (V)} - \text{AVREFP voltage (V)}))$ to the maximum value.

Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 μs . Accordingly, use standard mode 2 with the longer sampling time.

Note 6. This value is indicated as a ratio (%FSR) to the full-scale value.

2. Low-voltage modes 1 and 2

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq \text{AVREFP} \leq \text{VDD} \leq 5.5\text{ V}$, $\text{VSS} = 0\text{ V}$,
reference voltage (+) = AVREFP ($\text{ADREFP1} = 0$, $\text{ADREFP0} = 1$), reference voltage (-) = AVREFM ($\text{ADREFM} = 1$),
target pins: internal reference voltage^{Note 1} and temperature sensor output voltage^{Note 1})

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8		12	bit
Conversion clock	f _{AD}		1		24	MHz
Overall error ^{Notes 2, 3, 4, 5}	AINL	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±9	LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±9	LSB
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±11.5	LSB
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±12.0	LSB
Conversion time ^{Note 6}	t _{CONV}	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	3.33		μs
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	5.0		μs
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	10.0		μs
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	20.0		μs
Zero-scale error ^{Notes 2, 3, 4, 5, 7}	E _{ZS}	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.21	%FSR
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.21	%FSR
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.27	%FSR
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.28	%FSR
Full-scale error ^{Notes 2, 3, 4, 5, 7}	E _{FS}	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.21	%FSR
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.21	%FSR
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.27	%FSR
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±0.28	%FSR
Integral linearity error ^{Notes 2, 4, 5}	ILE	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±4.0	LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±4.0	LSB
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±4.5	LSB
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$		±4.5	LSB
Differential linearity error ^{Note 2}	DLE	12-bit resolution	$2.7\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	±1.5		LSB
			$2.4\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	±1.5		LSB
			$1.8\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	±2.0		LSB
			$1.6\text{ V} \leq \text{AVREFP} = \text{VDD} \leq 5.5\text{ V}$	±2.0		LSB
Analog input voltage	V _{AIN}		0		AVREFP	V

<R> **Note 1.** If the internal reference voltage, temperature sensor output voltage, or TSCAP voltage of the CTSU is to be A/D converted, VDD must be at least 1.8 V.

Note 2. This value does not include the quantization error (±1/2 LSB).

Note 3. When pins ANI16 to ANI26 are selected as the target pins for conversion or the TSCAP voltage of the CTSU is to be A/D converted, the maximum values are as follows. If the CTSU TSCAP voltage is to be A/D converted, VDD must be at least 1.8 V.

Overall error: Add ±3 LSB to the maximum value.

Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. When reference voltage (+) = VDD and reference voltage (-) = VSS, the maximum values are as follows.

Overall error: Add ±10 LSB to the maximum value.

Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.

Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When $\text{AVREFP} < \text{VDD}$, the maximum values are as follows.

Overall error/zero-scale error/full-scale error: Add (±0.75 LSB × (VDD voltage (V) - AVREFP voltage (V))) to the maximum value.

Integral linearity error: Add (±0.2 LSB × (VDD voltage (V) - AVREFP voltage (V))) to the maximum value.

- Note 6.** When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 μ s. Accordingly, use standard mode 2 with the longer sampling time, and use the conversion clock (f_{AD}) of no more than 16 MHz.
- Note 7.** This value is indicated as a ratio (%FSR) to the full-scale value.

3. When the internal reference voltage is selected as reference voltage (+)

($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$, low-voltage modes 1 and 2,
 reference voltage (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0),
 reference voltage (-) = AVREFM (ADREFM = 1)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8			bit
Conversion clock	f _{AD}	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1		2	MHz
Zero-scale error ^{Notes 1, 2, 3}	E _{ZS}	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			±0.6	%FSR
Integral linearity error ^{Notes 1, 3}	I _{LE}	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			±2.0	LSB
Differential linearity error ^{Note 1}	D _{LE}	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		±1.0		LSB
Analog input voltage	V _{AIN}		0		V _{BGR} Note 4	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 reference voltage (-) is selected as V_{SS} , the maximum values are as follows.

Zero-scale error: Add $\pm 0.35\%$ FSR to the maximum value.

Integral linearity error: Add ± 0.5 LSB to the maximum value.

Note 4. See **42.6.2 Temperature sensor/internal reference voltage characteristics**.

42.6.2 Temperature sensor/internal reference voltage characteristics

($T_A = -40$ to $+105^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Temperature sensor output voltage	VTMPS25	Setting ADS register = 80H, $T_A = +25^{\circ}\text{C}$		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/ $^{\circ}\text{C}$
Operation stabilization wait time	tAMP		5			μs

42.6.3 D/A converter characteristics

($T_A = -40$ to $+105^{\circ}\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = \text{EVSS} = 0\text{ V}$)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 8 M Ω	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			± 2.5	LSB
		Rload = 4 M Ω	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			± 2.5	LSB
Settling time	tSET	Cload = 20 pF	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			3	μs
			$1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			6	μs

42.6.4 Comparator characteristics

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
Input voltage range	IVREF	Input to the IVREF0 and IVREF1 pins C0LVL = 0, C1LVL = 0		0		VDD -1.4	V
		Input to the IVREF0 and IVREF1 pins C0LVL = 1, C1LVL = 1		1.4		VDD	V
	IVCMP	Input to the IVCMP0 and IVCMP1 pins		-0.3		VDD +0.3	V
Output delay	td	VDD = 3.0 V, Input slew rate > 1 V/μs	High-speed mode			1.5	μs
			Low-speed mode		3.0		μs
Offset voltage	VIOCMP	High-speed mode				50	mV
		Low-speed mode				40	mV
Operation stabilization wait time	tCMP			30			μs
Internal reference voltage ^{Note}	VBGR2	Internal reference voltage		1.4		1.6	V
	VDA2	D/A converter 2 output + comparator offset voltage	1.8 V ≤ VDD ≤ 5.5 V			VDD × (DACS2)/ 256 + VIOCMP	V
D/A converter 2 settling time	tSET2	When changing the DACS2 settings	2.7 V ≤ VDD ≤ 5.5			3	μs
			1.6 V ≤ VDD ≤ 5.5 V			6	μs

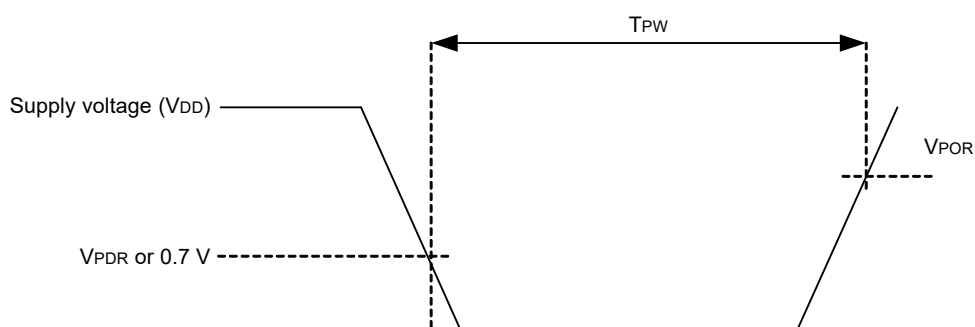
Note The internal reference voltage can be selected as comparator reference voltage only when 1.8 V ≤ VDD ≤ 5.5 V.

42.6.5 POR circuit characteristics

(TA = -40 to +105°C, VSS = 0 V)

Item	Symbols	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 (HIOSSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



42.6.6 LVD circuit characteristics

1. LVD0 Detection Voltage in the Reset Mode and Interrupt Mode

($T_A = -40$ to $+105^\circ\text{C}$, $V_{PDR} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

Item		Symbols	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		t _{LW}		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		Symbols	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	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	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	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

42.6.7 Power supply voltage rising slope characteristics

(TA = -40 to +105°C, Vss = 0 V)

Item	Symbols	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 VDD reaches the operating voltage range shown in AC characteristics.

42.6.8 LCD characteristics

42.6.8.1 External resistance division method

1. Static display mode

($T_A = -40$ to $+105^\circ\text{C}$, $V_{L4} (\text{Min.}) \leq V_{DD} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = 0 \text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
LCD drive voltage	V_{L4}		2.0		V_{DD}	V

2. 1/2 bias method, 1/4 bias method

($T_A = -40$ to $+105^\circ\text{C}$, $V_{L4} (\text{Min.}) \leq V_{DD} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = 0 \text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
LCD drive voltage	V_{L4}		2.7		V_{DD}	V

3. 1/3 bias method

($T_A = -40$ to $+105^\circ\text{C}$, $V_{L4} (\text{Min.}) \leq V_{DD} \leq 5.5 \text{ V}$, $V_{SS} = EV_{SS} = 0 \text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
LCD drive voltage	V_{L4}		2.5		V_{DD}	V

42.6.8.2 Internal voltage boosting method

1. 1/3 bias method, VL1 reference (MDSET2 to MDSET0 = 001B, LBAS1 to LBAS0 = 01B)

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
VL1 output voltage	VL1	C1 to C4 Note 1 = 0.47 μF	VLCD4-0 = 04H	0.97	1.01	1.04	V
			VLCD4-0 = 05H	1.00	1.04	1.08	V
			VLCD4-0 = 06H	1.04	1.07	1.11	V
			VLCD4-0 = 07H	1.07	1.11	1.14	V
			VLCD4-0 = 08H	1.10	1.14	1.18	V
			VLCD4-0 = 09H	1.13	1.17	1.21	V
			VLCD4-0 = 0AH	1.16	1.21	1.25	V
			VLCD4-0 = 0BH	1.20	1.24	1.28	V
			VLCD4-0 = 0CH	1.23	1.27	1.32	V
			VLCD4-0 = 0DH	1.26	1.31	1.35	V
			VLCD4-0 = 0EH	1.29	1.34	1.38	V
			VLCD4-0 = 0FH	1.33	1.37	1.42	V
			VLCD4-0 = 10H	1.36	1.40	1.45	V
			VLCD4-0 = 11H	1.39	1.44	1.49	V
			VLCD4-0 = 12H	1.42	1.47	1.52	V
			VLCD4-0 = 13H	1.45	1.50	1.55	V
			VLCD4-0 = 14H	1.49	1.54	1.59	V
			VLCD4-0 = 15H	1.52	1.57	1.62	V
			VLCD4-0 = 16H	1.55	1.60	1.66	V
			VLCD4-0 = 17H	1.58	1.64	1.69	V
			VLCD4-0 = 18H	1.61	1.67	1.73	V
			VLCD4-0 = 19H	1.65	1.70	1.76	V
			VLCD4-0 = 1AH	1.68	1.74	1.79	V
VL2 output voltage	VL2	C1 to C4 Note 1 = 0.47 μF	2VL1 -5%	2VL1	2VL1	V	
VL4 output voltage	VL4	C1 to C4 Note 1 = 0.47 μF	3VL1 -6%	3VL1	3VL1	V	
Reference voltage setup time Note 2	tvWAIT1		10			ms	
Voltage boost wait time Note 3	tvWAIT2	C1 to C4 Note 1 = 0.47 μF	500			ms	

Note 1. C1 to C4 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL4 and GND.

C1 = C2 = C3 = C4 = 0.47 μF ±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (MDSET2 to MDSET0 = 001B) if the default reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

2. 1/3 bias method, VL2 reference (MDSET2 to MDSET0 = 101B, LBAS1 to LBAS0 = 01B)

(TA = -40 to +105°C, VL2 (Max.) + 0.1 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
VL1 output voltage	VL1	C1 to C4 Note 1 = 0.47 μF		1/2VL2 -5%	1/2VL2	1/2VL2	V
VL2 output voltage	VL2	C1 to C4 Note 1 = 0.47 μF	VLCD4-0 = 04H	1.94	2.02	2.11	V
			VLCD4-0 = 05H	2.00	2.09	2.18	V
			VLCD4-0 = 06H	2.07	2.16	2.25	V
			VLCD4-0 = 07H	2.13	2.22	2.32	V
			VLCD4-0 = 08H	2.19	2.29	2.39	V
			VLCD4-0 = 09H	2.26	2.36	2.46	V
			VLCD4-0 = 0AH	2.32	2.42	2.53	V
			VLCD4-0 = 0BH	2.39	2.49	2.59	V
			VLCD4-0 = 0CH	2.45	2.56	2.66	V
			VLCD4-0 = 0DH	2.51	2.62	2.73	V
			VLCD4-0 = 0EH	2.58	2.69	2.80	V
			VLCD4-0 = 0FH	2.64	2.76	2.87	V
			VLCD4-0 = 10H	2.70	2.82	2.94	V
			VLCD4-0 = 11H	2.77	2.89	3.01	V
			VLCD4-0 = 12H	2.83	2.96	3.08	V
			VLCD4-0 = 13H	2.90	3.02	3.15	V
			VLCD4-0 = 14H	2.96	3.09	3.22	V
			VLCD4-0 = 15H	3.02	3.15	3.29	V
			VLCD4-0 = 16H	3.09	3.22	3.35	V
			VLCD4-0 = 17H	3.15	3.29	3.42	V
			VLCD4-0 = 18H	3.21	3.35	3.49	V
			VLCD4-0 = 19H	3.28	3.42	3.56	V
			VLCD4-0 = 1AH	3.34	3.49	3.63	V
VL4 output voltage	VL4	C1 to C4 Note 1 = 0.47 μF		3/2VL2 -6%	3/2VL2	3/2VL2	V
Reference voltage setup time Note 2	tvWAIT1			10			ms
Voltage boost wait time Note 3	tvWAIT2	C1 to C4 Note 1 = 0.47 μF		500			ms

Note 1. C1 to C4 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL4 and GND.

C1 = C2 = C3 = C4 = 0.47 μF ±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (MDSET2 to MDSET0 = 101B) if the default reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

3. 1/4 bias method, VL1 reference (MDSET2 to MDSET0 = 001B, LBAS1 to LBAS0 = 10B)

(TA = -40 to +105°C, 1.8 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions		Min.	Typ.	Max.	Unit
VL1 output voltage	VL1	C1 to C5 Note 1 = 0.47 μF	VLCD4-0 = 04H	0.97	1.01	1.04	V
			VLCD4-0 = 05H	1.00	1.04	1.08	V
			VLCD4-0 = 06H	1.04	1.07	1.11	V
			VLCD4-0 = 07H	1.07	1.11	1.14	V
			VLCD4-0 = 08H	1.10	1.14	1.18	V
			VLCD4-0 = 09H	1.13	1.17	1.21	V
			VLCD4-0 = 0AH	1.16	1.21	1.25	V
			VLCD4-0 = 0BH	1.20	1.24	1.28	V
			VLCD4-0 = 0CH	1.23	1.27	1.32	V
			VLCD4-0 = 0DH	1.26	1.31	1.35	V
VL2 output voltage	VL2	C1 to C5 Note 1 = 0.47 μF	2VL1 -5%	2VL1	2VL1	V	
VL3 output voltage	VL3	C1 to C5 Note 1 = 0.47 μF	3VL1 -6%	3VL1	3VL1	V	
VL4 output voltage	VL4	C1 to C5 Note 1 = 0.47 μF	4VL1 -6%	4VL1	4VL1	V	
Reference voltage setup time Note 2	tVWAIT1		10			ms	
Voltage boost wait time Note 3	tVWAIT2	C1 to C5 Note 1 = 0.47 μF	500			ms	

Note 1. C1 to C5 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL3 and GND.

C5 is to be connected between VL4 and GND.

C1 = C2 = C3 = C4 = C5 = 0.47 μF ±30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (MDSET2 to MDSET0 = 001B) if the default reference voltage is used) until voltage boosting starts (VLCON = 1).**Note 3.** This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

42.6.8.3 Capacitor split method

1. 1/3 bias method, VDD reference (MDSET2 to MDSET0 = 010B, LBAS1 to LBAS0 = 01B)

(TA = -40 to +105°C, 2.2 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
VL4 output voltage	VL4	C1 to C4 = 0.47 μF ^{Note 1}		VDD		V
VL2 output voltage	VL2	C1 to C4 = 0.47 μF ^{Note 1}	2/3VL4 -3%	2/3VL4	2/3VL4 +3%	V
VL1 output voltage	VL1	C1 to C4 = 0.47 μF ^{Note 1}	1/3VL4 -3%	1/3VL4	1/3VL4 +3%	V
Capacitor split wait time ^{Note 2}	tVWAIT		100			ms

Note 1. C1 to C4 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL4 and GND. This is required when the voltage generation method is switched to the internal voltage boosting method.

C1 = C2 = C3 = C4 = 0.47 μF ±30%

Note 2. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

2. 1/3 bias method, VL4 reference (MDSET2 to MDSET0 = 110B, LBAS1 to LBAS0 = 01B)

(TA = -40 to +105°C, 3.2 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
VL4 output voltage	VL4	C1 to C4 = 0.47 μF ^{Note 1}	2.89	3.04	3.20	V
VL2 output voltage	VL2	C1 to C4 = 0.47 μF ^{Note 1}	1.89	2.03	2.17	V
VL1 output voltage	VL1	C1 to C4 = 0.47 μF ^{Note 1}	0.94	1.01	1.08	V
Reference voltage setup time ^{Note 2}	tVWAIT1		10			ms
Capacitor split wait time ^{Note 3}	tVWAIT		100			ms

Note 1. C1 to C4 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL4 and GND.

C1 = C2 = C3 = C4 = 0.47 μF ±30%

Note 2. The time required to wait from when the capacitor split method is selected (MDSET2 to MDSET0 = 110B) until voltage bucking starts (VLCON = 1).

Note 3. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

3. 1/4 bias method, VDD reference (MDSET2 to MDSET0 = 010B, LBAS1 to LBAS0 = 10B)

(TA = -40 to +105°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = EVSS = 0 V)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
VL4 output voltage	VL4	C1 to C5 = 0.47 μF ^{Note 1}		VDD		V
VL3 output voltage	VL3	C1 to C5 = 0.47 μF ^{Note 1}	3/4VL4 -3%	3/4VL4	3/4VL4 +3%	V
VL2 output voltage	VL2	C1 to C5 = 0.47 μF ^{Note 1}	1/2VL4 -3%	1/2VL4	1/2VL4 +3%	V
VL1 output voltage	VL1	C1 to C5 = 0.47 μF ^{Note 1}	1/4VL4 -3%	1/4VL4	1/4VL4 +3%	V
Capacitor split wait time ^{Note 2}	tVWAIT		100			ms

Note 1. C1 to C5 are the capacitors that are to be connected to the LCD drive voltage pins of the MCU.

C1 is to be connected between CAPH and CAPL.

C2 is to be connected between VL1 and GND.

C3 is to be connected between VL2 and GND.

C4 is to be connected between VL3 and GND. This is required when the voltage generation method is switched to the internal voltage boosting method.

C5 is to be connected between VL4 and GND.

C1 = C2 = C3 = C4 = C5 = 0.47 μF ±30%

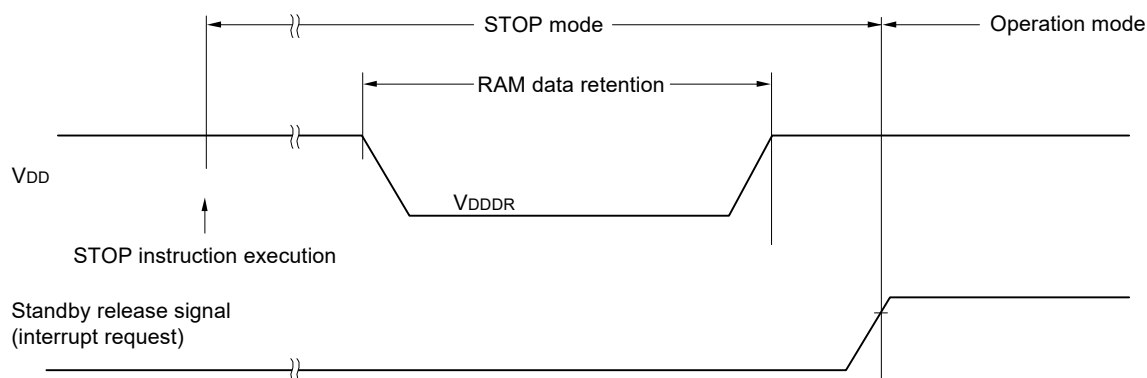
Note 2. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

42.7 RAM Data Retention Characteristics

($T_A = -40$ to $+105^\circ\text{C}$, $V_{SS} = 0\text{V}$)

Item	Symbols	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.



42.8 Flash Memory Programming Characteristics

($T_A = -40$ to $+105^\circ\text{C}$, $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
CPU/peripheral hardware clock frequency	fCLK		1		32	MHz
Number of code flash rewrites Notes 1, 2, 3	Cerwr	Retained for 10 years T _A = +85°C Note 4	10,000			Times
		Retained for 20 years T _A = +85°C Note 4	1,000			
Number of data flash rewrites Notes 1, 2, 3			1,000,000			
Retained for 5 years T _A = +85°C Note 4		100,000				
Retained for 20 years T _A = +85°C Note 4		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 the flash memory programmer or self-programming function is in use.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Note 4. This temperature is the average value at which data are retained.

1. Code flash memory

(TA = -40 to +105°C, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbols	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
Erase 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, 1.6 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

Item		Symbols	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
Erase 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.

42.9 Dedicated Flash Memory Programmer Communication (UART)

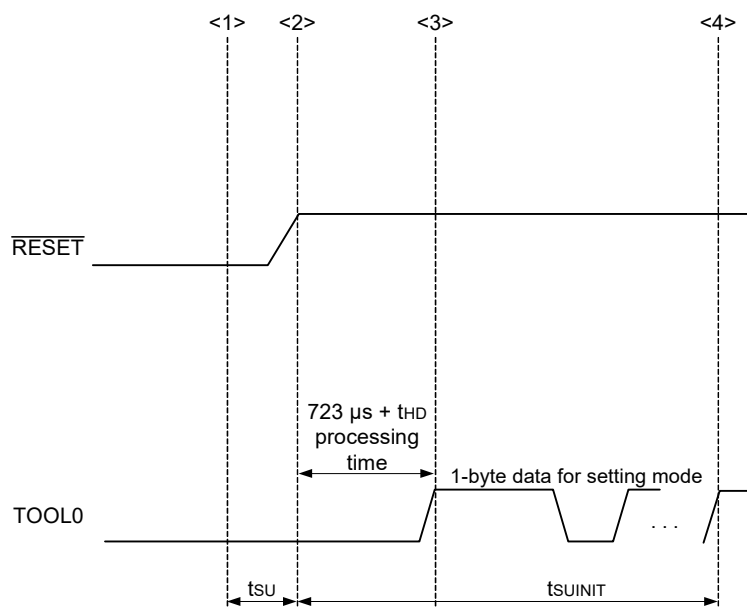
($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

Item	Symbols	Conditions	Min.	Typ.	Max.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

42.10 Timing of Entry to Flash Memory Programming Modes

($T_A = -40$ to $+105^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS} = 0\text{ V}$)

Item	Symbols	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 t_{suINIT} : The time during which the communications for the initial setting must be completed within 100 ms after the external reset is released.

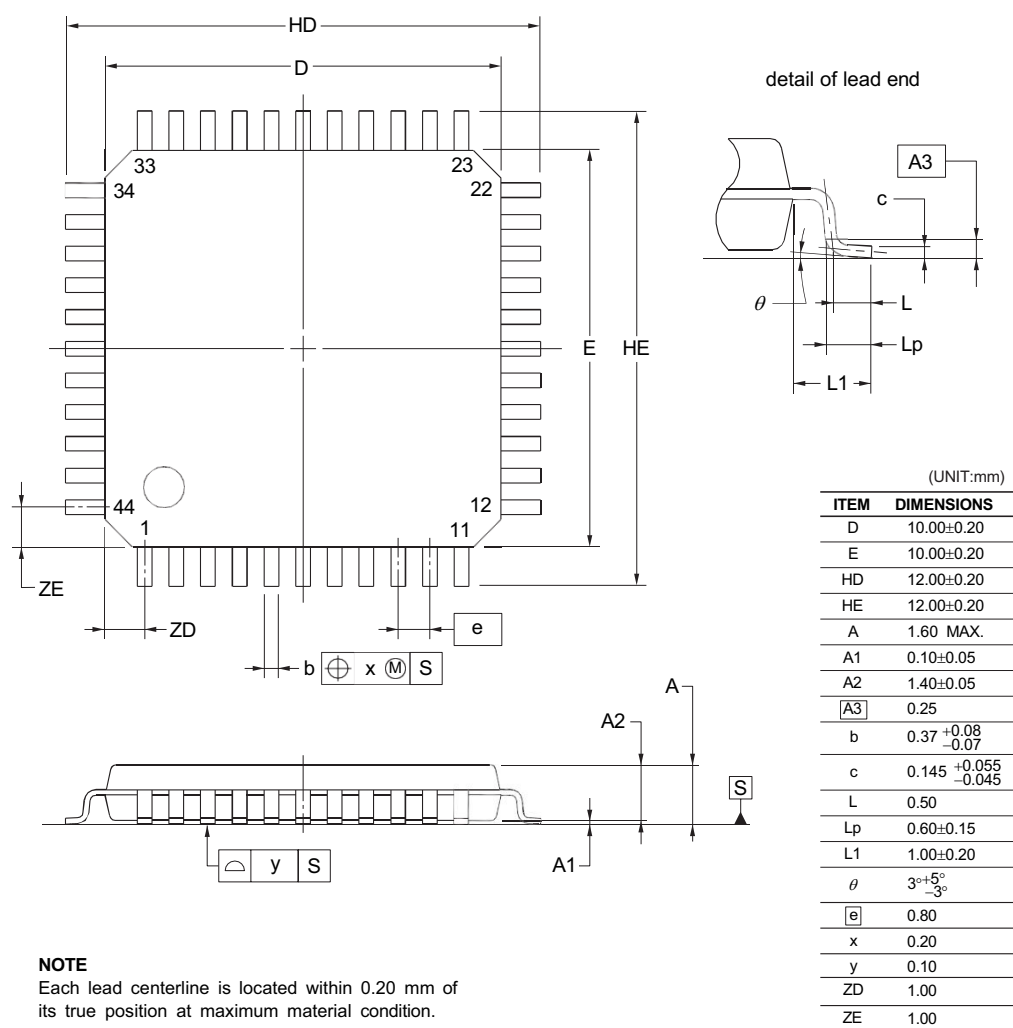
t_{su} : Time to release the external reset after the TOOL0 pin is set to the low level

t_{HD} : 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 43 Package Drawings

43.1 44-pin Products

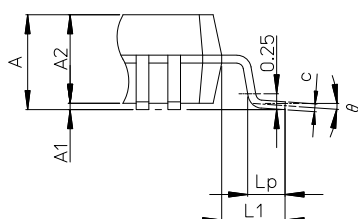
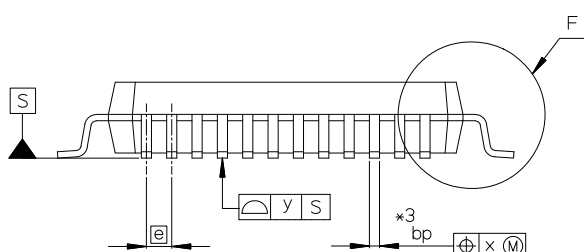
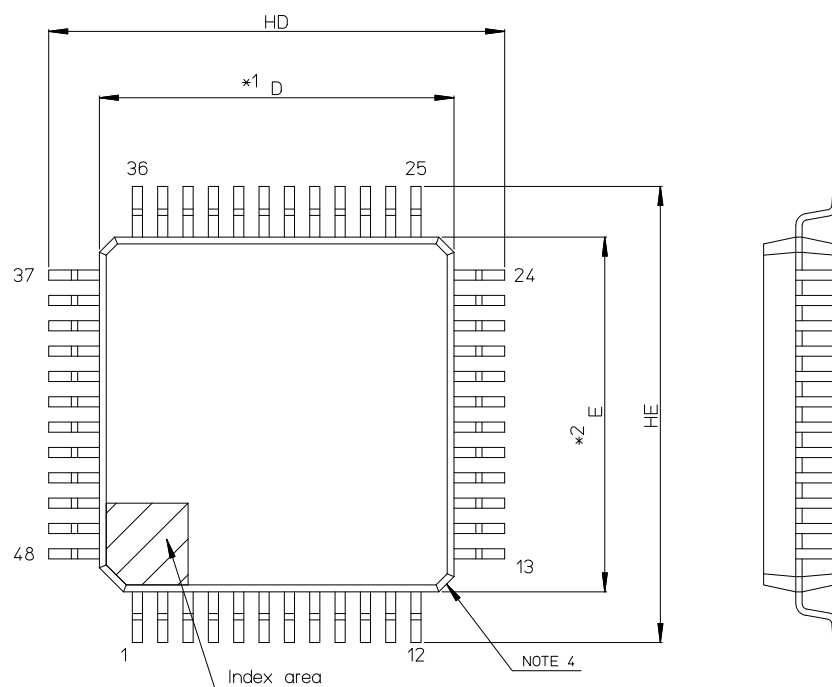
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP44-10x10-0.80	PLQP0044GC-A	P44GB-80-UES-2	0.36



©2012 Renesas Electronics Corporation. All rights reserved.

43.2 48-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2g



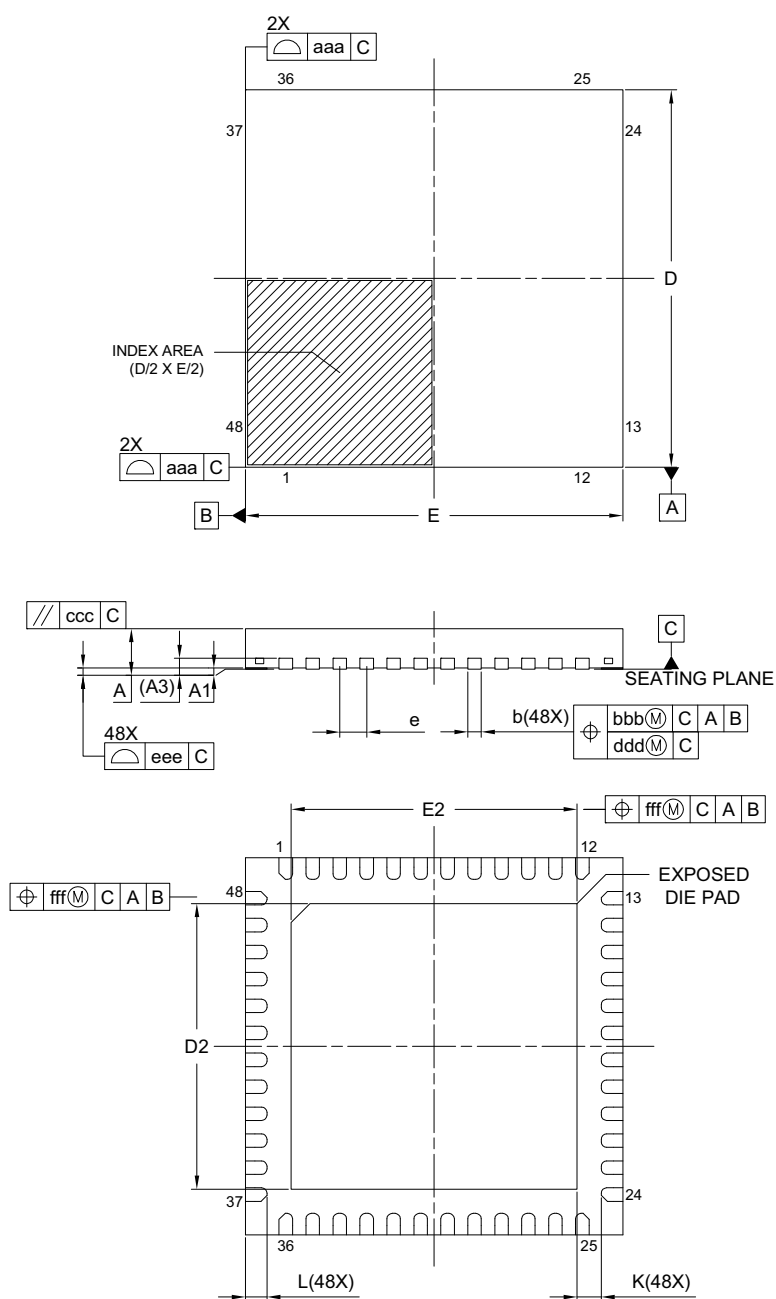
Detail F

NOTE)

1. DIMENSIONS *1* AND *2* DO NOT INCLUDE MOLD FLASH.
2. DIMENSION *3* DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	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	—

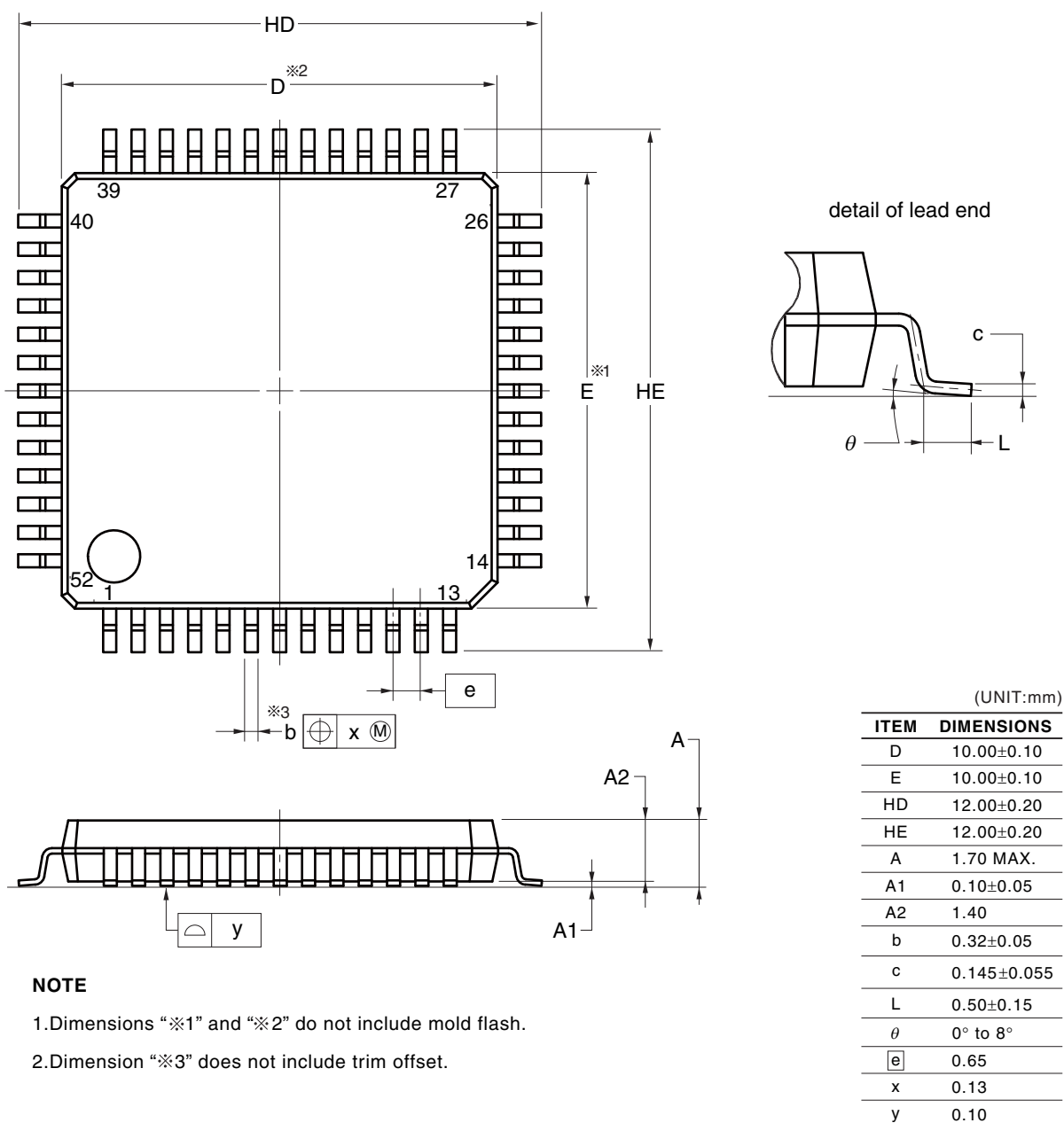
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



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		

43.3 52-pin Products

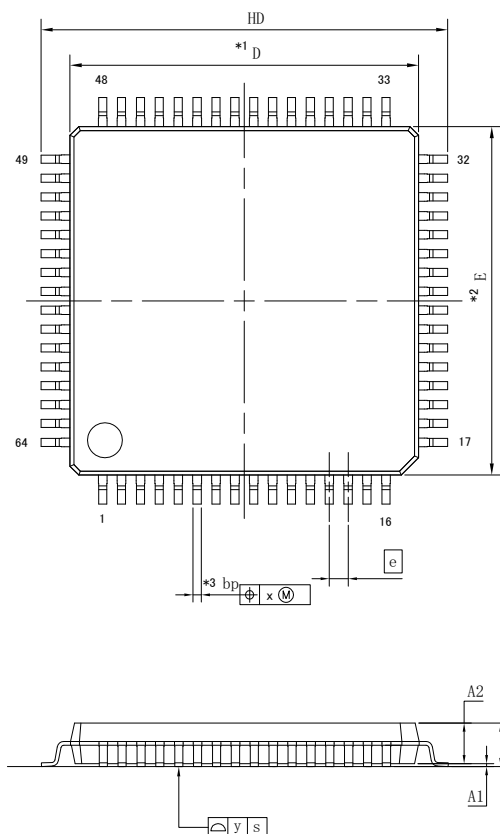
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP52-10x10-0.65	PLQP0052JA-A	P52GB-65-GBS-1	0.3



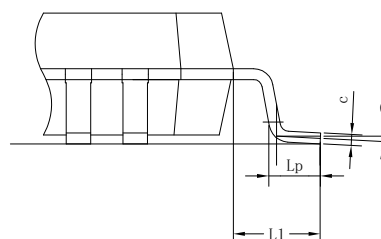
© 2012 Renesas Electronics Corporation. All rights reserved.

43.4 64-pin Products

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LQFP64-12x12-0.65	PLQP0064JB-A	0.50



detail of lead end

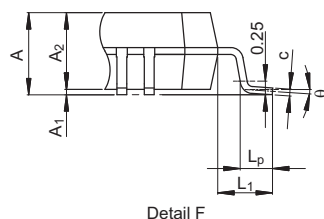
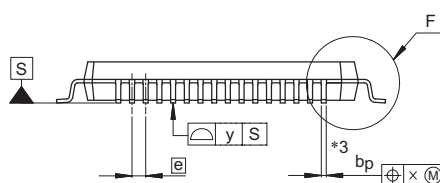
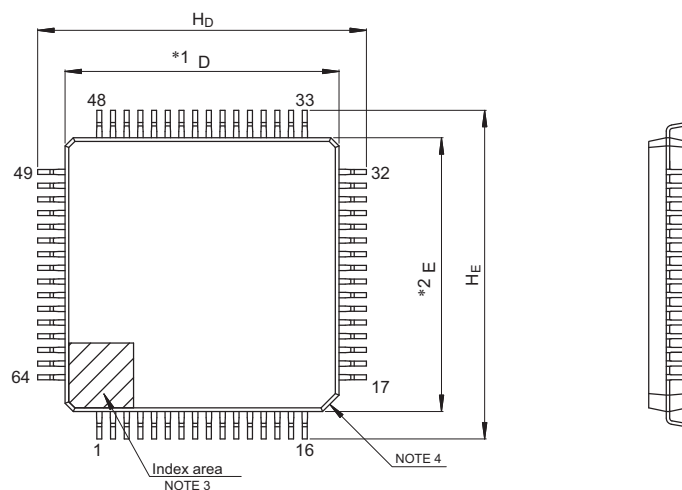


NOTE
 1.DIMENSIONS "*1" AND "*2"DO NOT INCLUDE MOLD FLASH.
 2.DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
E	11.90	12.00	12.10
D	11.90	12.00	12.10
A ₂	—	1.40	—
H _D	13.80	14.00	14.20
H _E	13.80	14.00	14.20
A	—	—	1.70
A ₁	0.05	—	0.15
L _p	0.45	0.60	0.75
L ₁	—	1.00	—
b _p	0.27	0.32	0.37
c	0.09	—	0.20
e	—	0.65	—
θ	0.00	3.50	8.00
x	—	—	0.08
y	—	—	0.08

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm



NOTE)

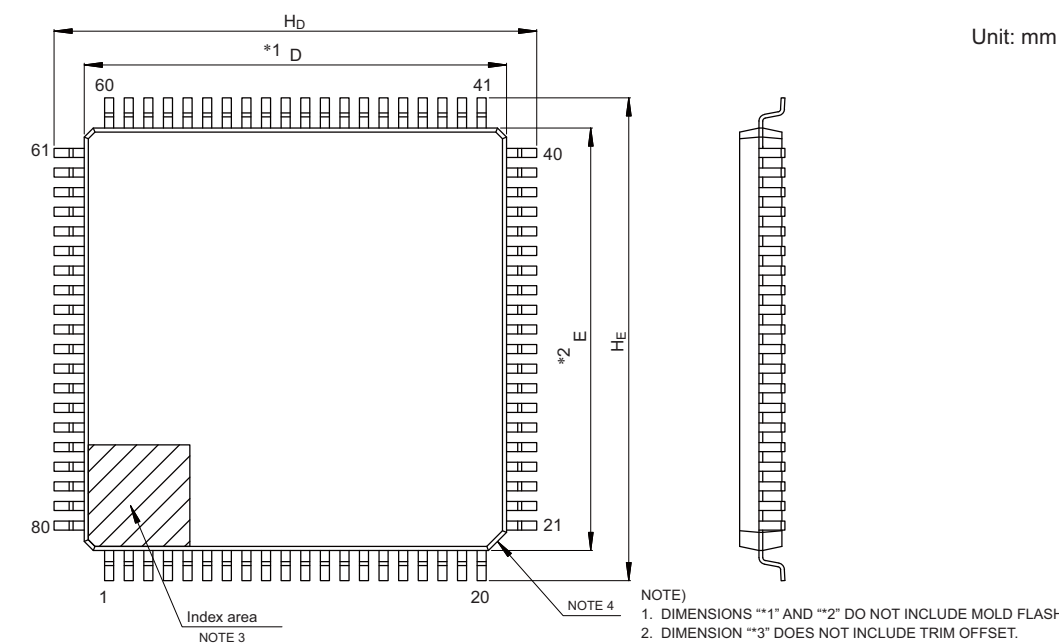
1. DIMENSIONS "**1" AND "**2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "**3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A ₂	—	1.4	—
H _D	11.8	12.0	12.2
H _E	11.8	12.0	12.2
A	—	—	1.7
A ₁	0.05	—	0.15
b _p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L _p	0.45	0.6	0.75
L ₁	—	1.0	—

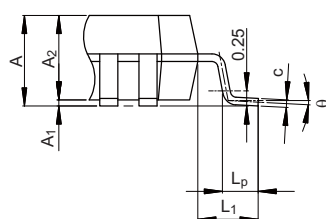
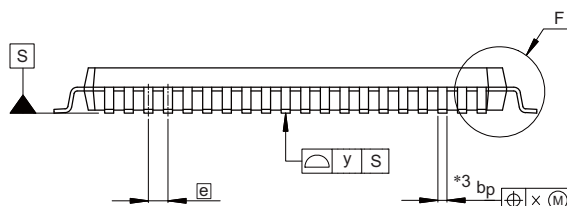
© 2015 Renesas Electronics Corporation. All rights reserved.

43.5 80-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LQFP80-14x14-0.65	PLQP0080JA-B	—	0.6



- 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.

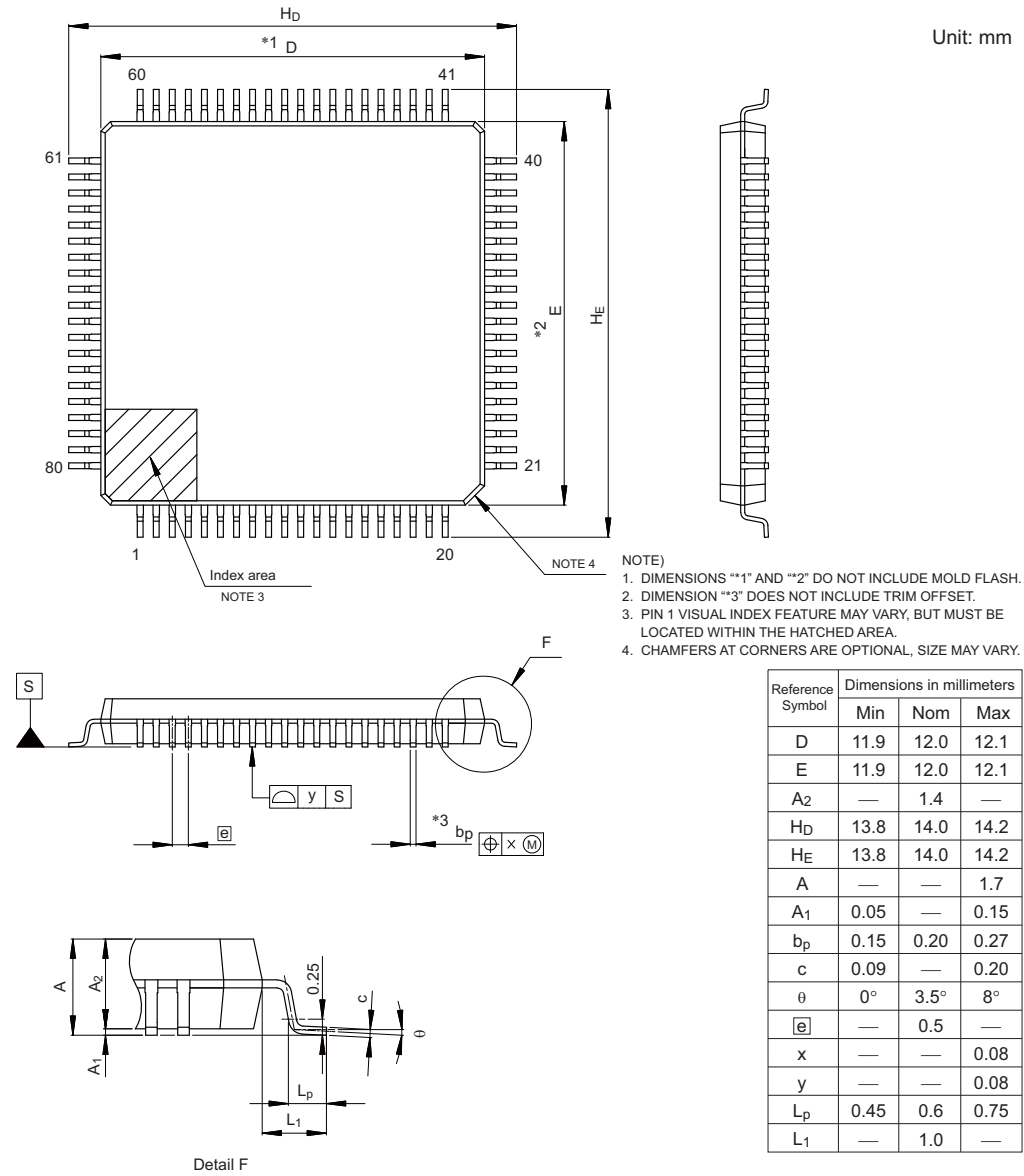


Detail F

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A ₂	—	1.4	—
H _D	15.8	16.0	16.2
H _E	15.8	16.0	16.2
A	—	—	1.7
A ₁	0.05	—	0.15
b _p	0.22	0.30	0.38
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
L _p	0.45	0.6	0.75
L ₁	—	1.0	—

© 2016 Renesas Electronics Corporation. All rights reserved.

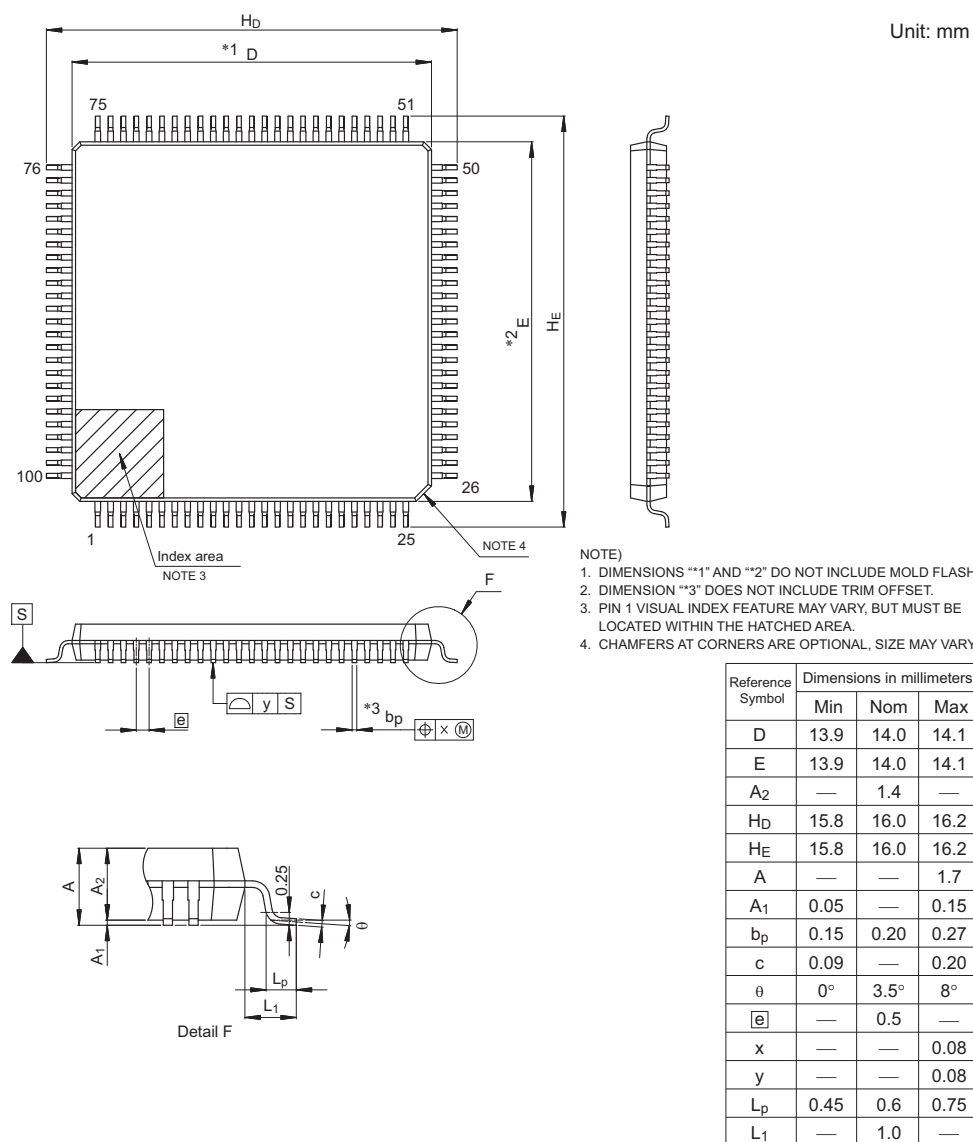
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5



© 2017 Renesas Electronics Corporation. All rights reserved.

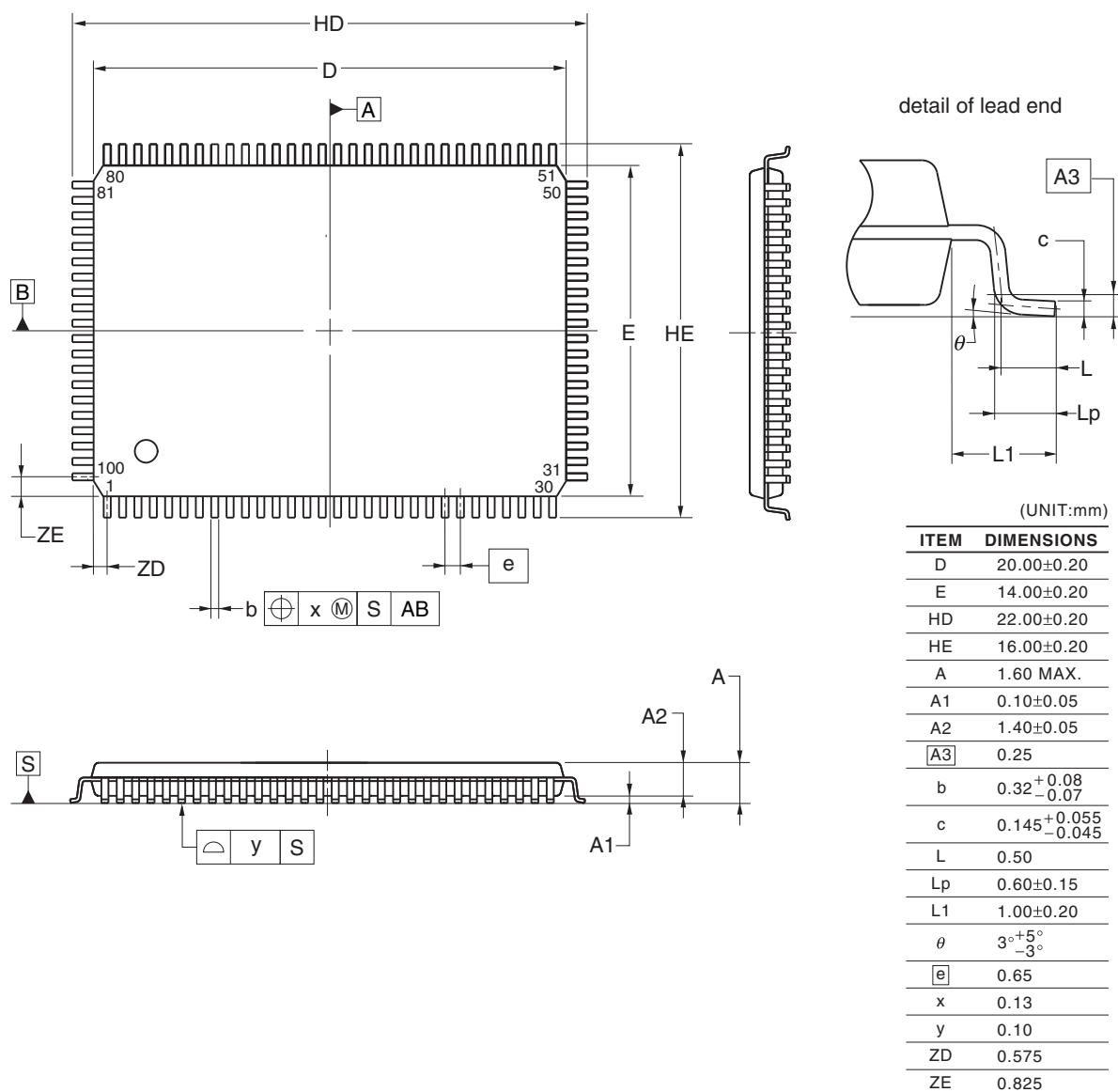
43.6 100-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6



© 2015 Renesas Electronics Corporation. All rights reserved.

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP100-14x20-0.65	PLQP0100JC-A	P100GF-65-GBN-1	0.92



© 2012 Renesas Electronics Corporation. All rights reserved.

Appendix A Revision History

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
Section 2 Pin Functions		
p.81	2.2.3.2 Connecting the VBATn pin to the battery for use in backing up (n = 0, 1): The descriptions were modified.	(c)
p.81	Figure 2 - 1 Example of the Connection of the VBATn Pin (n = 0, 1) was modified.	(c)
p.82	2.2.3.3 Using the VBATn Pin (n = 0, 1) 1. Making the initial settings for the VBATn pins: The description was modified.	(a)
p.95	Figure 2 - 13 Pin Block Diagram for Pin Type 7-5-32 was modified.	(a)
p.96	Figure 2 - 14 Pin Block Diagram for Pin Type 7-5-33: Caution was modified.	(a)
p.103	Figure 2 - 21 Pin Block Diagram for Pin Type 7-44-4 was modified.	(a)
p.104	Figure 2 - 22 Pin Block Diagram for Pin Type 7-45-1 was modified.	(a)
p.105	Figure 2 - 23 Pin Block Diagram for Pin Type 7-45-2 was modified.	(a)
p.116	Figure 2 - 29 Pin Block Diagram for Pin Type 8-5-20 was modified.	(a)
p.131	Figure 2 - 37 Pin Block Diagram for Pin Type 8-44-4 was modified.	(a)
p.133	Figure 2 - 38 Pin Block Diagram for Pin Type 8-45-1 was modified.	(a)
p.135	Figure 2 - 39 Pin Block Diagram for Pin Type 8-45-2 was modified.	(a)
p.137	Figure 2 - 40 Pin Block Diagram for Pin Type 12-38-2 was modified.	(a)
p.138	Figure 2 - 41 Pin Block Diagram for Pin Type 12-38-3 was modified.	(a)
Section 3 CPU Architecture		
p.154	3.1.3 Internal data memory space: Caution 2 was modified.	(c)
Section 4 Port Functions		
p.236	Figure 4 - 17 Format of LCD Input Switch Control Register (ISCLCD) was modified.	(c)
p.286	Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions was modified.	(a)
Section 13 External Signal Sampler		
p.738	Figure 13 - 4 Format of External Signal Sampler Control Register 0 (EXSDM0) was modified.	(a), (c)
Section 16 A/D Converter (ADC)		
p.784	Figure 16 - 9 Format of A/D Converter Mode Register 2 (ADM2) was modified.	(c)
Section 17 D/A Converter (DAC)		
p.827	Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1) was modified.	(a)
Section 18 Comparator (CMP)		
p.840	Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1) was modified.	(a)
Section 19 Serial Array Unit (SAU)		
p.886	Figure 19 - 21 Format of Input Switch Control Register (ISC) was modified.	(a)

Remark "Classification" in the above table classifies revisions as follows.

- (a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/2)

Page	Description	Classification
Section 20 Serial Interface IICA (IICA)		
p.1047	20.4.2 Setting transfer clock by using IICWLn and IICWHn registers: The descriptions were modified.	(a)
Section 21 Serial Interface UARTA (UARTA)		
p.1152	21.3.4 Baud rate generator d) Permissible baud rate range during reception: The descriptions were modified.	(a)
p.1152	Table 21 - 5 Maximum/Minimum Permissible Baud Rate Error was modified.	(a)
Section 22 LCD Controller/Driver		
p.1184	Figure 22 - 8 Format of LCD Input Switch Control Register (ISCLCD) was modified.	(c)
p.1202, p.1203	Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs c) Six-time-slice and eight-time-slice (100 to 64 pins): The title was modified.	(a)
p.1204, p.1205	Table 22 - 14 Relationship between LCD Display Data Register Contents and Segment and Common Outputs d) Six-time-slice and eight-time-slice (52 to 44 pins): The title was modified.	(a)
Section 24 Logic and Event Link Controller (ELCL)		
p.1300	Figure 24 - 5 Format of Input Signal Select Registers n (ELISELn): Notes 9 and 10 were modified.	(a)
Section 35 Capacitive Sensing Unit (CTSU2La)		
p.1520, p.1521	35.1.2 Measurement states was modified.	(a)
p.1528	Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) was modified.	(a)
p.1548	Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) was modified.	(a)
p.1563	35.3.4 Scan modes: 2. Multi-scan mode was modified.	(a)
p.1563	35.3.6 Multiple electrode connection was modified.	(a)
Section 38 Flash Memory		
p.1655	Table 38 - 19 Settings of the BTBLS and BAPR Bits: Note 1 was modified.	(c)
Section 42 Electrical Characteristics		
p.1691	42.1 Absolute Maximum Ratings (2/2): Note was modified.	(a)
p.1693	42.2.3 Characteristics of the On-chip Oscillators was modified.	(b)
p.1696	42.3.1 Pin characteristics (2/7): Note 3 was modified.	(a)
p.1699	42.3.1 Pin characteristics (5/7): Note was modified.	(a)
p.1702, p.1704, p.1705, p.1707	42.3.2 Supply current characteristics: 1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products was modified.	(b)
p.1708	42.3.2 Supply current characteristics: What used to be 2. 44- to 64-pin products with 64 to 128 Kbytes of flash ROM was deleted.	(b)
p.1709 to p.1711	42.3.2 Supply current characteristics: 2. Peripheral functions (common to all products) was modified.	(b)
p.1752	42.6.1 A/D converter characteristics: 1. Normal modes 1 and 2 was modified.	(b)
p.1754	42.6.1 A/D converter characteristics 2. Low-voltage modes 1 and 2: Note 1 was modified.	(c)

Remark "Classification" in the above table classifies revisions as follows.

- (a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Section indicates the section of each edition.

(1/4)

Edition	Description	Section
Rev.0.50	First edition issued.	Throughout
Rev.0.80	2.1.6 100-pin products: Table was modified.	Section 2 Pin Functions
	2.2.2 Description of pin functions: Table was modified.	
	2.2.3.3 Using the VBATn Pin (n = 0, 1): The description was modified.	
	Figure 2 – 2 State Transitions in Switching the Power Supply Pin between the VDD and VBATn Pins (n = 0, 1): Caution 3 was modified.	
	Figure 2 - 11 Pin Block Diagram for Pin Type 7-5-33 was modified.	
	Figure 2 - 14 Pin Block Diagram for Pin Type 7-5-32 was modified.	
	Figure 2 - 24 Pin Block Diagram for Pin Type 8-1-10 was modified.	
	Figure 2 - 25 Pin Block Diagram for Pin Type 8-5-4 was modified.	
	Figure 2 - 26 Pin Block Diagram for Pin Type 8-5-17 was modified.	
	Figure 2 - 27 Pin Block Diagram for Pin Type 8-5-18 was modified.	
	Figure 2 - 28 Pin Block Diagram for Pin Type 8-5-19 was modified.	
	Figure 2 - 29 Pin Block Diagram for Pin Type 8-5-20 was modified.	
	Figure 2 - 30 Pin Block Diagram for Pin Type 8-6-10 was modified.	
	Figure 2 - 31 Pin Block Diagram for Pin Type 8-10-1 was modified.	
	Figure 2 - 32 Pin Block Diagram for Pin Type 8-10-2 was modified.	
	Figure 2 - 33 Pin Block Diagram for Pin Type 8-43-1 was modified.	
	Figure 2 - 34 Pin Block Diagram for Pin Type 8-44-1 was modified.	
	Figure 2 - 35 Pin Block Diagram for Pin Type 8-44-2 was modified.	
	Figure 2 - 36 Pin Block Diagram for Pin Type 8-44-3 was modified.	
	Figure 2 - 37 Pin Block Diagram for Pin Type 8-44-4 was modified.	
	Figure 2 - 38 Pin Block Diagram for Pin Type 8-45-1 was modified.	
	Figure 2 - 39 Pin Block Diagram for Pin Type 8-45-2 was modified.	
	Table 3 - 3 Vector Table: Table and Notes 1 and 2 were modified.	Section 3 CPU Architecture
	Table 3 - 6 List of 2nd Special Function Registers was modified.	
	Figure 4 - 1 Format of Port Mode Registers (PMxx): Note was added.	Section 4 Port Functions
	Figure 4 - 2 Format of Port Registers (Pxx): Note 1 was added.	
	Table 4 - 3 Products Having 40-mA Output Port Pins and the Pins Supporting 40-mA Output was modified.	
	Figure 4 - 13 Format of 40-mA Port Output Control Register (PTDC): Notes 1 to 6 were modified.	
	Figure 4 - 14 Format of Port Function Output Enable Register 0 (PFOE0) was modified.	
	Table 4 - 6 Examples of Register and Output Latch Settings for Alternate Functions was modified.	
	Table 5 - 2 Features of Each Flash Operation Mode: Note 3 was modified.	Section 5 Operation State Control
	Figure 5 - 2 Format of Flash Operating Mode Select Register (FLMODE) was modified and Note 2 was added.	

(2/4)

Edition	Description	Section
Rev.0.80	5.5.3 Details of LP (low-power main) mode: The description was modified.	Section 5 Operation State Control
	Figure 6 - 1 Block Diagram of Clock Generator: Remark was modified.	Section 6 Clock Generator
	Figure 6 - 4 Format of System Clock Control Register (CKC): Note 3 was added and Caution 2 was modified.	
	6.6.2 Example of setting the X1 oscillator clock: The descriptions were modified.	
	Figure 7 - 1 Entire Configuration of Timer Array Unit 0 (Example: 64-pin Products) was modified.	Section 7 Timer Array Unit (TAU)
	Figure 7 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit 0 was modified.	
	7.10.2 Point for caution when a timer output is to be used as an event input for the ELCL: The description was modified.	
	8.5.8 Procedures for enabling and disabling operation in the event counter mode while the CPU is in the STOP mode: The descriptions were modified.	Section 8 Timer RJ
	Section 9 was entirely modified.	Section 9 16-bit Timers KB40, KB41, and KB42
	Figure 13 - 6 Example of Waveforms of Output Signal from the EXSDO0 and EXSDO1 Pins: Caution 1 was modified.	Section 13 External Signal Sampler
	13.6 Cautions: The description of 4. was modified.	
	Figure 19 - 1 Block Diagram of Serial Array Unit 0 was modified.	Section 19 Serial Array Unit (SAU)
	Table 20 - 2 INTIICAn Generation Timing and Control of Clock Stretching: Note 3 was added.	Section 20 Serial Interface IICA (IICA)
	20.5.8 Timing of Generation of the Interrupt Request Signal (INTIICAn) and Control of Clock Stretching: 1. During address transmission/reception was modified.	
	Figure 21 - 10 Format of UARTA Clock Select Register n (UTAnCK) was modified.	Section 21 Serial Interface UARTA (UARTA)
	Figure 22 - 2 Format of LCD Mode Register 0 (LCDM0) was modified.	Section 22 LCD Controller/Driver
	Figure 22 - 5 Format of Subsystem Clock Supply Mode Control Register (OSMC): Caution 2 was modified.	
	22.3.8 LCD port function registers (PFSEGx), a) Operation of ports that alternately function as COMxx/SEGxx pins: The description was modified.	
	22.3.8 LCD port function registers (PFSEGx), a) Operation of ports that alternately function as COMxx/SEGxx pins: The descriptions were added.	
	Figure 22 - 28 Examples of LCD Drive Power Connections (Capacitor Split Method) was modified and Note 1 was added.	
	Figure 24 - 5 Format of Input Signal Select Registers n (ELISELn): Note 3 was modified.	Section 24 Logic and Event Link Controller (ELCL)
	Table 25 - 1 Interrupt Source List and Note 8 were modified. What used to be Note 7 was deleted.	Section 25 Interrupt Functions
	Table 25 - 2 Flags Corresponding to Interrupt Request Sources, Notes 2, and 3 were modified.	
	Figure 25 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) was modified.	
	Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) was modified.	
	Figure 25 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) was modified.	

(3/4)

Edition	Description	Section
Rev.0.80	Table 27 - 1 Operating States in the HALT Mode (1): Note was added.	Section 27 Standby Function
	Table 27 - 2 Operating States in the HALT Mode (2) was modified, and Note was added.	
	Table 27 - 3 Operating States in the STOP Mode: Note was added.	
	Table 27 - 4 Operating States in the SNOOZE Mode: Note was added.	
	30.5 Points for Caution when the Voltage Detector Is to be Used: 5. Procedure for setting the LVD1 detection voltage was modified.	Section 30 Voltage Detector (LVD)
	31.1 Functions of Oscillation Stop Detector was modified.	Section 31 Oscillation Stop Detector
	Figure 31 - 4 Format of Oscillation Stop Detection Control Register (OSDC): Note 2 was modified.	
	Figure 31 - 5 Timing of Oscillation Stop Detection by Oscillation Stop Detector was modified.	
	The descriptions under Figure 35-2 were modified.	Section 35 Capacitive Sensing Unit (CTS2La)
	Table 35 - 1 Functional Overview of the CTSU was modified.	
	35.1.1 Functional overview of the CTSU was modified.	
	Figure 35 - 3 Block Diagram of the CTSU was modified.	
	Figure 35 - 4 Configuration of Clock Signals for Sensor Driving Pulse Output was modified.	
	Table 35 - 2 Input and Output Pins of the CTSU was modified.	
	35.1.2 Measurement states was added.	
	Figure 35 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) was modified.	
	Figure 35 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) was modified.	
	Figure 35 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUSMCHL, CTSUSMCHH) was modified.	
	35.2.6 CTSU channel enable control registers AL, AH, and BL (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL) was modified.	
	Figure 35 - 11 Format of CTSU Channel Enable Control Registers AL, AH, and BL (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL): Caution 1 was modified. What used to be Caution 2 was deleted.	
	35.2.7 CTSU channel transmit/receive control registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL) was modified.	
	Figure 35 - 12 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, and BL (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL): Caution 1 was modified. What used to be Caution 2 was deleted.	
	Figure 35 - 13 Format of CTSU Status Register L (CTSUSURL) was modified.	
	35.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1): Descriptions were added.	
	Figure 35 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) and Note 1 were modified.	
	35.2.10 CTSU sensor counter registers (CTSUSC, CTSUUC): Descriptions were added.	
	Figure 35 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) was modified.	
	Figure 35 - 17 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) was modified.	
	35.3 Operation was added.	
	Figure 35 - 27 Control Procedure in the Self-capacitance Method was modified.	

(4/4)

Edition	Description	Section
Rev.0.80	Figure 35 - 28 Control Procedure in the Mutual Capacitance Method was modified.	Section 35 Capacitive Sensing Unit (CTSU2La)
	35.4 Usage Notes on the Capacitive Sensing Unit: 35.4.1 was modified, and 35.4.2 to 35.4.11 were added.	
	Figure 37 - 3 Format of User Option Byte (000C2H or address of the area to be used in swapping): Note 2	Section 37 Option Bytes
	Table 38 - 8 Signature Data List was modified.	Section 38 Flash Memory
	Table 38 - 9 Example of Signature Data was modified.	
	Figure 38 - 37 Procedure for Updating the Program (Bank Swapping after a Reset): Note 1 was modified.	
	Figure 38 - 38 Procedure for Updating the Program (Active Bank Swapping): Note 1 was modified.	
	39.5 On-chip Debugging in the Bank Programming Mode was modified.	Section 39 On-chip Debugging
	42.2.3 Characteristics of the On-chip Oscillators: Note 3 was modified.	Section 42 Electrical Characteristics
	42.3.1 Pin characteristics was modified.	
	42.3.2 Supply current characteristics 1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products (1/4): Note 1 was modified.	
	42.3.2 Supply current characteristics 1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products (3/4): Note 1 was modified.	
	42.3.2 Supply current characteristics 1. 44- to 64-pin products with 256 to 512 Kbytes of flash ROM and 80- to 100-pin products (4/4): Note 1 was modified.	
	42.3.2 Supply current characteristics 2. 44- to 64-pin products with 64 to 128 Kbytes of flash ROM (1/4): Note 1 was modified.	
	42.3.2 Supply current characteristics 2. 44- to 64-pin products with 64 to 128 Kbytes of flash ROM (3/4): Note 1 was modified.	
	42.3.2 Supply current characteristics 2. 44- to 64-pin products with 64 to 128 Kbytes of flash ROM (4/4): Note 1 was modified	
	42.3.2 Supply current characteristics 3. Peripheral functions (common to all products) (3/3) and Note 5 were modified.	
	42.6.8.2 Internal voltage boosting method: 1. 1/3 bias method, VL1 reference (MDSET2 to MDSET0 = 001B, LBAS1 to LBAS0 = 01B) was modified.	
	42.6.8.2 Internal voltage boosting method: 2. 1/3 bias method, VL1 reference (MDSET2 to MDSET0 = 001B, LBAS1 to LBAS0 = 01B) was modified.	
	42.6.8.2 Internal voltage boosting method: 3. 1/3 bias method, VL1 reference (MDSET2 to MDSET0 = 001B, LBAS1 to LBAS0 = 01B) was modified.	
	42.6.8.3 Capacitor split method 1. 1/3 bias method, VDD reference (MDSET2 to MDSET0 = 010B, LBAS1 to LBAS0 = 01B): Note 1 was modified.	
	42.6.8.3 Capacitor split method 1. 1/3 bias method, VDD reference (MDSET2 to MDSET0 = 010B, LBAS1 to LBAS0 = 01B): Note 1 was modified.	

RL78/L23 User's Manual: Hardware

Publication Date: Rev.0.50 Sep 30, 2024
Rev.1.00 May 30, 2025

Published by: Renesas Electronics Corporation

RL78/L23