User's Manual

16

RL78/I1C (512 KB)

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).

Renesas Electronics
www.renesas.com

Rev. 1.11 Mar 2024

Notice

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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/I1C (512 KB) and design and develop application systems and programs for these devices.
The target products are as follows.

- 80-pin: R5F10NMLDFB
- 100-pin: R5F10NPLDFB

Purpose

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

Organization

The RL78/I1C (512 KB) manual is separated into two parts: this manual and the software edition (common to the RL78 Family).

RL78/I1C (512 KB) **User's Manual** **Hardware**	**RL78 Microcontroller** **User's Manual** **Software**
● Pin functions	● CPU functions
● Internal block functions	● Instruction set
● Interrupts	● Explanation of each instruction
● Other on-chip peripheral functions	
● Electrical specifications	

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/I1C (512 KB) Microcontroller instructions:
  → Refer to the separate document **RL78 Family Software User's Manual (R01US0015E)**.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representations: $\overline{\times\times\times}$ (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 ···×××× or ××××B

Decimal ···××××

Hexadecimal ···××××H

Related Documents

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

Documents Related to Devices

Document Name	Document No.
RL78/I1C (512 KB) User's Manual Hardware	This manual
RL78 Family Software User's Manual	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	R20UT4025E
E1, E20 Emulator User's Manual	R20UT0398E
E2 Emulator User's Manual	R20UT3538E
E2 Lite Emulator User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User’s Manual	R20UT4066E
Renesas Flash Development Toolkit User’s Manual	R20UT0508E

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 MICROCOMPUTER 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

CHAPTER 1 OUTLINE

1.1 Features

Target applications

- Power, gas, and water meters

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 1.6 to 5.5 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.03125 μs: @ 32 MHz selection with high-speed on-chip oscillator) to ultra-low speed (66.6 μs: @ 15 kHz operation with low-speed on-chip oscillator)
- 16-bit multiplication, 16-bit multiply-accumulation, and 32-bit division are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 32 KB

Code flash memory

- Code flash memory: 512 KB (256 KB × 2 banks)
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with flash shield window function)
- Bank programming function: Enables updating of the user program while it is running.

Data flash memory

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 1.8 to 5.5 V

PLL clock

- 32 MHz is selectable (ΔΣ A/D converter is operable even when the PLL clock is selected as a CPU clock.)

High-speed on-chip oscillator

- Select from 1 to 32 MHz (TYP.). However when it is used as a clock for the ΔΣ A/D converter, select from 24 MHz (TYP.), 12 MHz (TYP.), 6 MHz (TYP.), or 3 MHz (TYP.).
- High accuracy: ±1.0% (V_{DD} = 1.8 to 5.5 V, T_A = –20 to +85°C)
- On-chip high-speed on-chip oscillator clock frequency correction function

Middle-speed on-chip oscillator

- Select from 4 MHz/2 MHz/1 MHz (However ΔΣ A/D converter is disabled.)

Operating ambient temperature

- T_A = -40 to +85°C

Power management and reset function

- On-chip power-on-reset (POR) circuit for Internal V_{DD} power supply
- On-chip RTC power-on-reset (RTCPOR) circuit for VRTC power supply
- On-chip voltage detector (LVD) (Select interrupt and reset from 14 levels)

Voltage detective circuit

- Detective voltage for V_{DD} pin (Select interrupt from 6 levels)
- Detective voltage for LVDVBAT pin (Select interrupt from 7 levels)
- Detective voltage for VRTC pin (Select interrupt from 4 levels)
- Detective voltage for EXLVD pin (Select interrupt from 1 level)

Data transfer controller (DTC)

- Transfer mode: Normal mode, repeat mode, block mode
- Activation source: Start by interrupt sources
- Chain transfer function

Event link controller (ELC)

- Event signals of 30 types can be linked to the specified peripheral function.

On-chip 32-bit multiplier and multiply-accumulator

- 32 bits × 32 bits = 64 bits (Unsigned or signed)
- 32 bits × 32 bits + 64 bits = 64 bits (Unsigned or signed)
- The results of multiply-and-accumulate operations (cumulative values) can be retained in any of 24 selectable buffer channels.

Serial interface

● Simplified SPI (CSI[Note]):	2 to 3 channels
● UART/UART (LIN-bus supported):	2 or 4 channels
● UART/IrDA:	1 channel
● Simplified I^2C communication:	2 to 3 channels
● I^2C communication:	1 channel
● Serial interface UARTMG (9600 bps @ 38.4 kHz):	2 channels

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.

Timer

● 16-bit timer:	10 channels (timer array unit (TAU): 8 channels, timer RJ: 2 channels)
● 12-bit interval timer:	1 channel
● 8-bit interval timer:	8 channels
● Independent power supply RTC:	1 channel (calendar for 99 years, alarm function, and clock correction function)
● Watchdog timer:	1 channel
● Sampling output timer detector (SMOTD):	2 units (6 channels for input, 6 channels for output)
● Oscillation stop detection circuit:	1 channel

LCD controller/driver

- Internal voltage boosting method, capacitor split method, and external resistance division method are switchable
- Segment signal output: 34 (30)[Note] to 42 (38)[Note]
- Common signal output: 4 (8)[Note]

A/D converter

- 24-Bit ΔΣ A/D converter: 3 or 4 channels
- 12-bit resolution A/D converter (AV_{DD} = 1.8 to 5.5 V): 4 or 6 channels
  Simultaneous sample-and-hold function: Three sample-and-hold circuits are installed.
- Internal reference voltage (1.45 V) and temperature sensor
- The voltage reference output voltage can be used as the reference voltage for the 12-bit A/D converter.
- The voltage reference output voltage can be selected from among 1.5 V (typ.), 2.0 V (typ.), and 2.5 V (typ.).

I/O port

- I/O port: 60 or 76 (N-ch open drain I/O [6 V tolerance]: 6,
  N-ch open drain I/O [EV_{DD} tolerance]: 13 or 18)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5/3 V device
- On-chip clock output/buzzer output controller
- On-chip key interrupt function

AES circuit

- Cipher modes of operation: GCM/ECB/CBC
- Encryption key length: 128/192/256 bits

Others

- On-chip BCD (binary-coded decimal) correction circuit

Note The values in parentheses are the number of signal outputs when 8 com is used.

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

○ ROM, RAM capacities

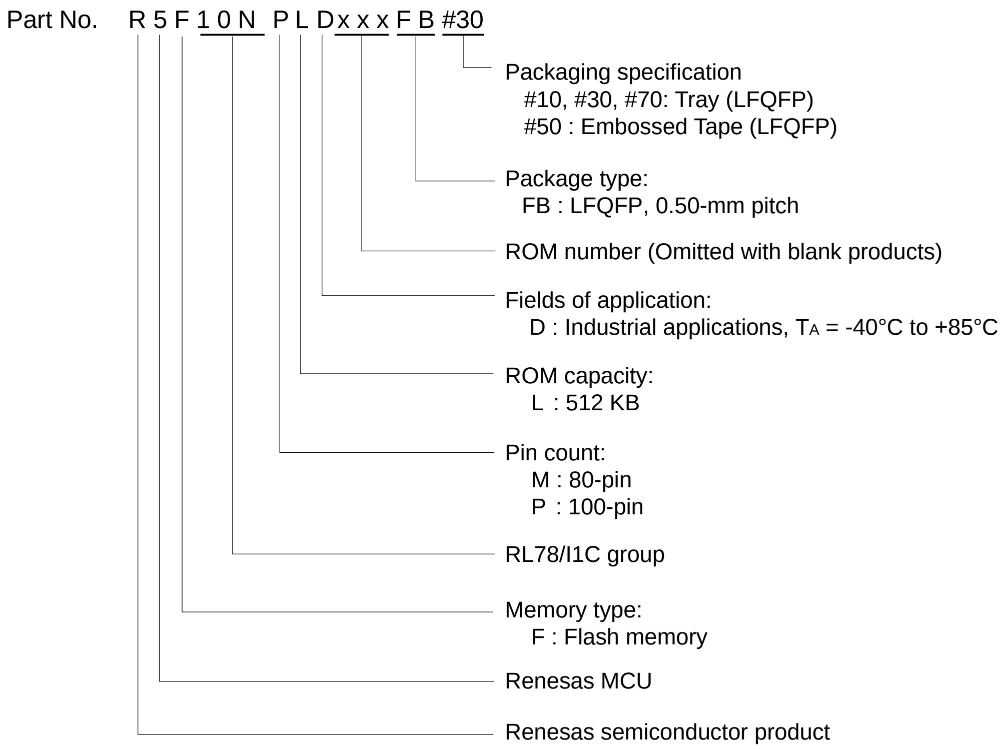
Code Flash	Data Flash	RAM	AES Function	RL78/I1C (512 KB)	
				80 pins	100 pins
512 KB	2 KB	32 KB[Note]	Mounted	R5F10NML	R5F10NPL

Note This is about 31 KB when the self-programming function is used. (For details, refer to **CHAPTER 3**.)

1.2 List of Part Numbers

<R>

Figure 1-1. Part Number, Memory Size, and Package of RL78/I1C (512 KB)



Part No. R 5 F 1 0 N P L D x x x F B #30

Packaging specification
#10, #30, #70: Tray (LFQFP)
#50 : Embossed Tape (LFQFP)

Package type:
FB : LFQFP, 0.50-mm pitch

ROM number (Omitted with blank products)

Fields of application:
D : Industrial applications, TA = -40°C to +85°C

ROM capacity:
L : 512 KB

Pin count:
M : 80-pin
P : 100-pin

RL78/I1C group

Memory type:
F : Flash memory

Renesas MCU

Renesas semiconductor product

<R>

Table 1-1. List of Ordering Part Numbers

Pin Count	Package	Data Flash	AES Function	Fields of Application[Note]	Ordering Part Number
80 pins	80-pin plastic LFQFP (12 × 12 mm, 0.5-mm pitch)	Mounted	Mounted	D	R5F10NMLDFB#10, R5F10NMLDFB#30, R5F10NMLDFB#50, R5F10NMLDFB#70
100 pins	100-pin plastic LFQFP (14 × 14 mm, 0.5-mm pitch)	Mounted	Mounted	D	R5F10NPLDFB#10, R5F10NPLDFB#30, R5F10NPLDFB#50, R5F10NPLDFB#70

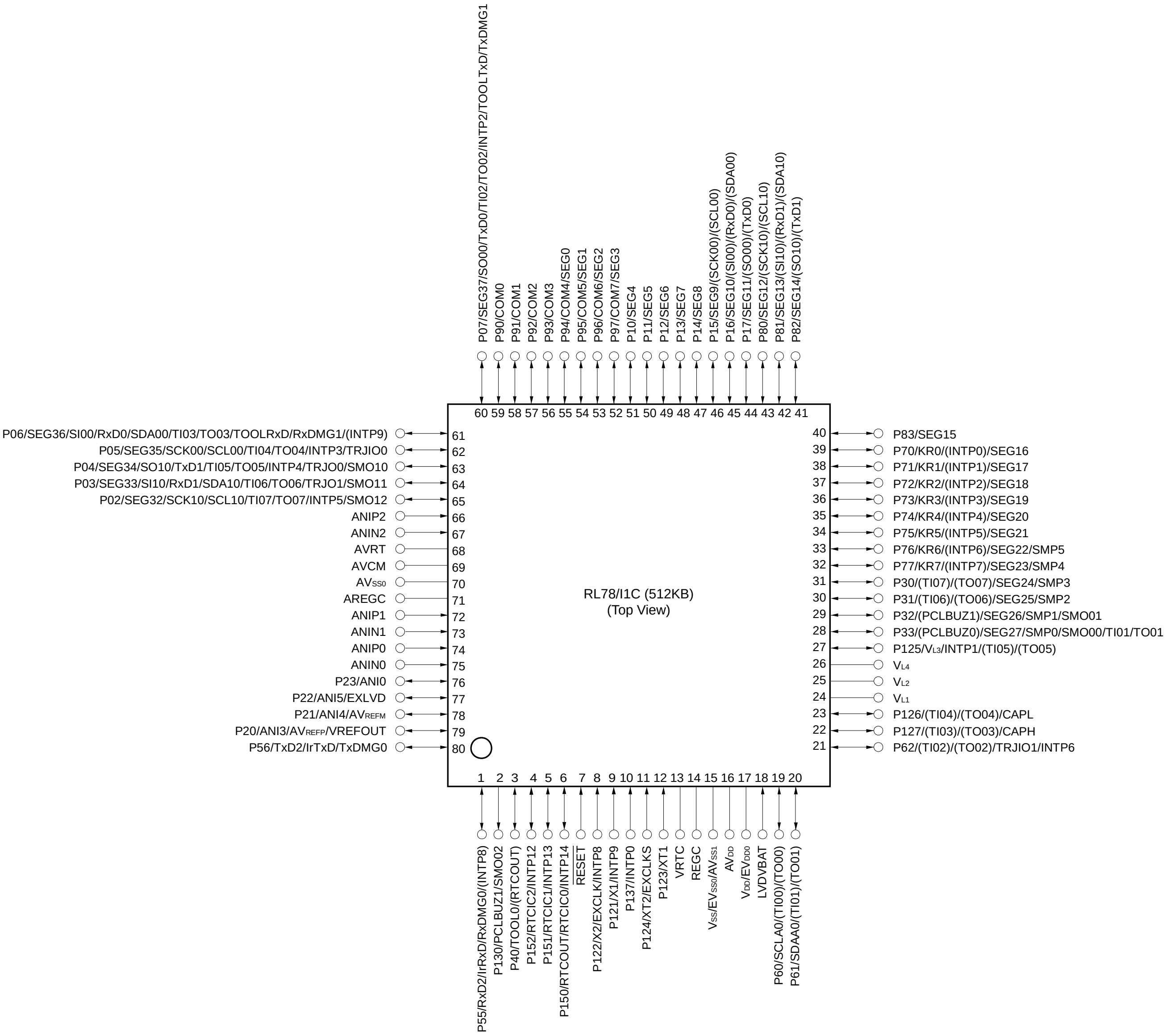
Note For the fields of application, see **Figure 1-1 Part Number, Memory Size, and Package of RL78/I1C (512 KB)**.

Caution The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.

1.3 Pin Configuration (Top View)

1.3.1 80-pin product

- 80-pin plastic LFQFP (12 × 12 mm, 0.5-mm pitch)


RL78/I1C (512KB)
(Top View)
1 P55/RxD2/IrRxD/RxDMG0/(INTP8)
2 P130/PCLBUZ1/SMO02
3 P40/TOOL0/(RTCOUT)
4 P152/RTCIC2/INTP12
5 P151/RTCIC1/INTP13
6 P150/RTCOUT/RTCIC0/INTP14
7 $\overline{\text{RESET}}$
8 P122/X2/EXCLK/INTP8
9 P121/X1/INTP9
10 P137/INTP0
11 P124/XT2/EXCLKS
12 P123/XT1
13 VRTC
14 REGC
15 $V_{SS}/EV_{SS0}/AV_{SS1}$
16 AV_{DD}
17 V_{DD}/EV_{DD0}
18 LVDVBAT
19 P60/SCLA0/(TI00)/(TO00)
20 P61/SDAA0/(TI01)/(TO01)
21 P62/(TI02)/(TO02)/TRJIO1/INTP6
22 P127/(TI03)/(TO03)/CAPH
23 P126/(TI04)/(TO04)/CAPL
24 V_{L1}
25 V_{L2}
26 V_{L4}
27 $P125/V_{L3}$/INTP1/(TI05)/(TO05)
28 P33/(PCLBUZ0)/SEG27/SMP0/SMO00/TI01/TO01
29 P32/(PCLBUZ1)/SEG26/SMP1/SMO01
30 P31/(TI06)/(TO06)/SEG25/SMP2
31 P30/(TI07)/(TO07)/SEG24/SMP3
32 P77/KR7/(INTP7)/SEG23/SMP4
33 P76/KR6/(INTP6)/SEG22/SMP5
34 P75/KR5/(INTP5)/SEG21
35 P74/KR4/(INTP4)/SEG20
36 P73/KR3/(INTP3)/SEG19
37 P72/KR2/(INTP2)/SEG18
38 P71/KR1/(INTP1)/SEG17
39 P70/KR0/(INTP0)/SEG16
40 P83/SEG15
41 P82/SEG14/(SO10)/(TxD1)
42 P81/SEG13/(SI10)/(RxD1)/(SDA10)
43 P80/SEG12/(SCK10)/(SCL10)
44 P17/SEG11/(SO00)/(TxD0)
45 P16/SEG10/(SI00)/(RxD0)/(SDA00)
46 P15/SEG9/(SCK00)/(SCL00)
47 P14/SEG8
48 P13/SEG7
49 P12/SEG6
50 P11/SEG5
51 P10/SEG4
52 P97/COM7/SEG3
53 P96/COM6/SEG2
54 P95/COM5/SEG1
55 P94/COM4/SEG0
56 P93/COM3
57 P92/COM2
58 P91/COM1
59 P90/COM0
60 P07/SEG37/SO00/TxD0/TI02/TO02/INTP2/TOOLTxD/TxDMG1
61 P06/SEG36/SI00/RxD0/SDA00/TI03/TO03/TOOLRxD/RxDMG1/(INTP9)
62 P05/SEG35/SCK00/SCL00/TI04/TO04/INTP3/TRJIO0
63 P04/SEG34/SO10/TxD1/TI05/TO05/INTP4/TRJO0/SMO10
64 P03/SEG33/SI10/RxD1/SDA10/TI06/TO06/TRJO1/SMO11
65 P02/SEG32/SCK10/SCL10/TI07/TO07/INTP5/SMO12
66 ANIP2
67 ANIN2
68 AVRT
69 AVCM
70 AV_{SS0}
71 AREGC
72 ANIP1
73 ANIN1
74 ANIP0
75 ANIN0
76 P23/ANI0
77 P22/ANI5/EXLVD
78 $P21/ANI4/AV_{REFM}$
79 $P20/ANI3/AV_{REFP}$/VREFOUT
80 P56/TxD2/IrTxD/TxDMG0

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

2. Make the voltage on the AV_{DD} pin the same as that on the V_{DD} and EV_{DD0} pins.

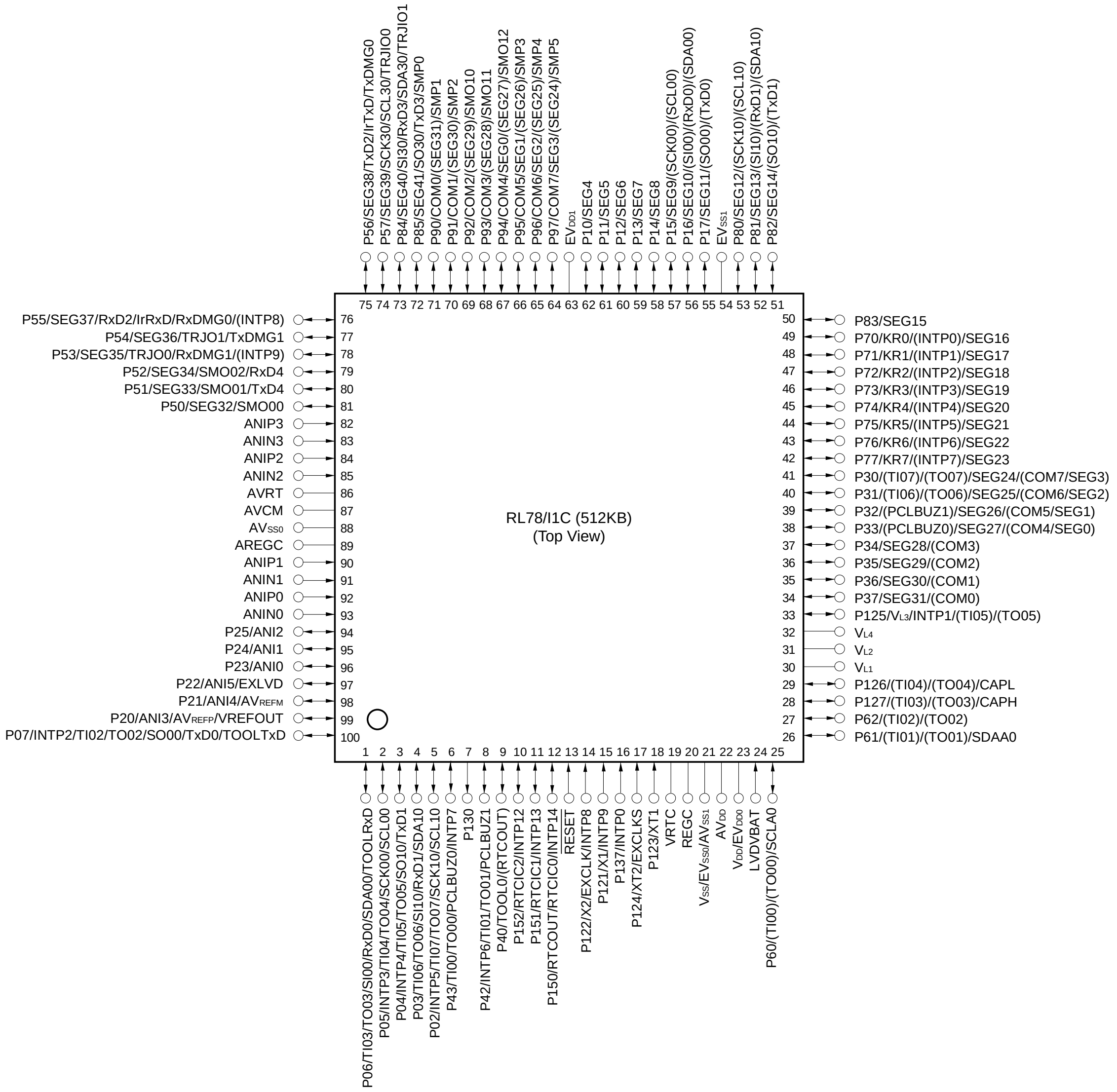
<R> **3. When a high-level signal is to be input to any pin among pins P150 to P152 (including if the pin is in use for a multiplexed pin function rather than for GPIO), the pin should always be individually connected via a resistor to V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V.**

Remarks 1. For pin identification, see **1.4 Pin Identification.**

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0).**

1.3.2 100-pin product

- 100-pin plastic LFQFP (14 × 14 mm, 0.5-mm pitch)



Cautions 1. Make the voltage on the EV$_{SS1}$ pin the same as that on the V$_{SS}$, EV$_{SS0}$, and AV$_{SS1}$ pins.

2. Make the voltage on the EV$_{DD1}$ pin the same as that on the V$_{DD}$ and EV$_{DD0}$ pins.

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

4. Make the voltage on the AV$_{DD}$ pin the same as that on the V$_{DD}$ and EV$_{DD0}$ pins.

<R> **5. When a high-level signal is to be input to any pin among pins P150 to P152 (including if the pin is in use for a multiplexed pin function rather than for GPIO), the pin should always be individually connected via a resistor to V$_{DD}$ or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V.**

Remarks 1. For pin identification, see **1.4 Pin Identification.**

2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the V$_{DD}$ and EV$_{DD1}$ pins and connect the V$_{SS}$ and EV$_{SS1}$ pins to separate ground lines.

3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0).**

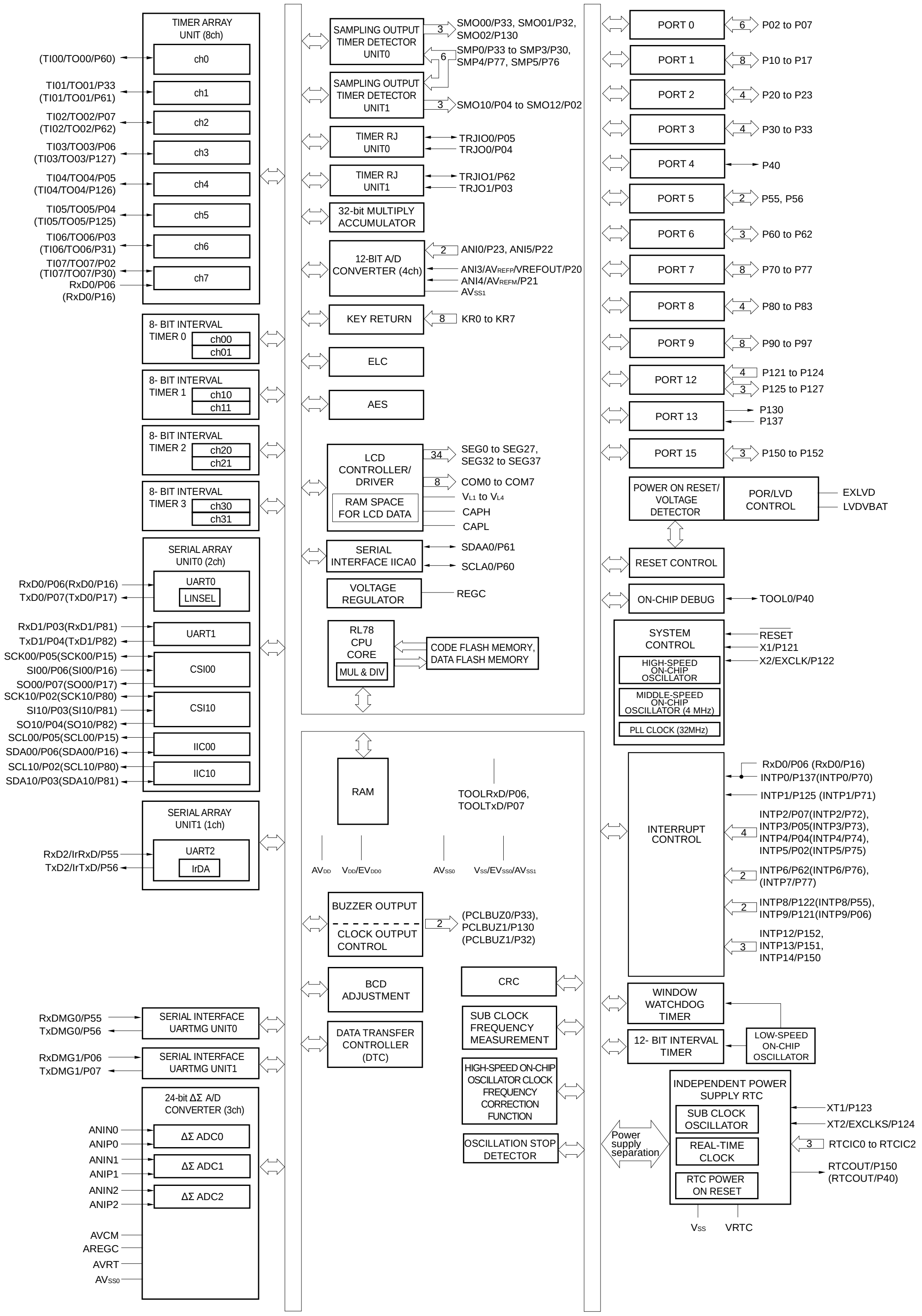
1.4 Pin Identification

ANI0 to ANI5:	Analog Input
ANIN0 to ANIN3, ANIP0 to ANIP3:	Analog Input for ΔΣ ADC
AREGC:	Regulator Capacitance for ΔΣ ADC
AVCM:	Control for ΔΣ ADC
AV$_{\mathrm{DD}}$:	Analog Power Supply
AV$_{\mathrm{REFM}}$:	12-bit A/D Converter Reference Potential (– side) Input
AV$_{\mathrm{REFP}}$:	12-bit A/D Converter Reference Potential (+ side) Input
AVRT:	Reference Potential for ΔΣ ADC
AV$_{\mathrm{SS0}}$:	Ground for ΔΣ ADC
AV$_{\mathrm{SS1}}$:	Ground for 12-bit A/D Converter
CAPH, CAPL:	Capacitor Connection for LCD Controller/Driver
COM0 to COM7:	Common Signal Output for LCD Controller/Driver
EV$_{\mathrm{DD0}}$, EV$_{\mathrm{DD1}}$:	Power Supply for Port
EV$_{\mathrm{SS0}}$, EV$_{\mathrm{SS1}}$:	Ground for Port
EXCLK:	External Clock Input (Main System Clock)
EXCLKS:	External Clock Input (Subsystem clock)
EXLVD:	External Input for Low Voltage Detector
INTP0 to INTP9, INTP12 to INTP14:	Interrupt Request From Peripheral
IrRxD:	Receive Data for IrDA
IrTxD:	Transmit Data for IrDA
KR0 to KR7:	Key Return
LVDVBAT:	Battery Backup Power Supply for Voltage Detector
P02 to P07:	Port 0
P10 to P17:	Port 1
P20 to P25:	Port 2
P30 to P37:	Port 3
P40, P42, P43:	Port 4
P50 to P57:	Port 5
P60 to P62:	Port 6
P70 to P77:	Port 7
P80 to P85:	Port 8
P90 to P97:	Port 9
P121 to P127:	Port 12

P130, P137:	Port 13
P150 to P152:	Port 15
PCLBUZ0, PCLBUZ1:	Programmable Clock Output/Buzzer Output
REGC:	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RTCOUT:	Real-time Clock Correction Clock (1 Hz/64 Hz) Output
RTCIC0 to RTCIC2:	RTC Time Capture Event Input
RxD0 to RxD4:	Receive Data for UART
RxDMG0, RxDMG1:	Receive Data for UARTMG
SCL00, SCL10, SCL30:	Serial Clock Output for Simplified IIC
SDA00, SDA10, SDA30:	Serial Data Input/Output for Simplified IIC
SCLA0 :	Serial Clock Input/Output for IICA0
SDAA0:	Serial Data Input/Output for IICA0
SCK00, SCK10, SCK30:	Serial Clock Input/Output for CSI
SEG0 to SEG41:	Segment Signal Output for LCD Controller/Driver
SI00, SI10, SI30:	Serial Data Input for CSI
SMP0 to SMP5:	Sampling Input
SMO00 to SMO02, SMO10 to SMO12:	Sampling Clock Output
SO00, SO10, SO30:	Serial Data Output for CSI
TI00 to TI07:	Timer Input
TO00 to TO07, TRJO0, TRJO1:	Timer Output
TOOL0:	Data Input/Output for Tool
TOOLRxD, TOOLTxD:	Data Input/Output for External Device
TRJIO0, TRJIO1:	Timer Input/Output
TxD0 to TxD4:	Transmit Data for UART
TxDMG0, TxDMG1:	Transmit Data for UARTMG
V$_{\mathrm{DD}}$:	Power Supply
V$_{\mathrm{L1}}$ to V$_{\mathrm{L4}}$:	Voltage for Driving LCD
VREFOUT:	Analog Reference Voltage Output
VRTC:	RTC Power Supply
V$_{\mathrm{SS}}$:	Ground
X1, X2:	Crystal Oscillator (Main System Clock)
XT1, XT2:	Crystal Oscillator (Subsystem Clock)

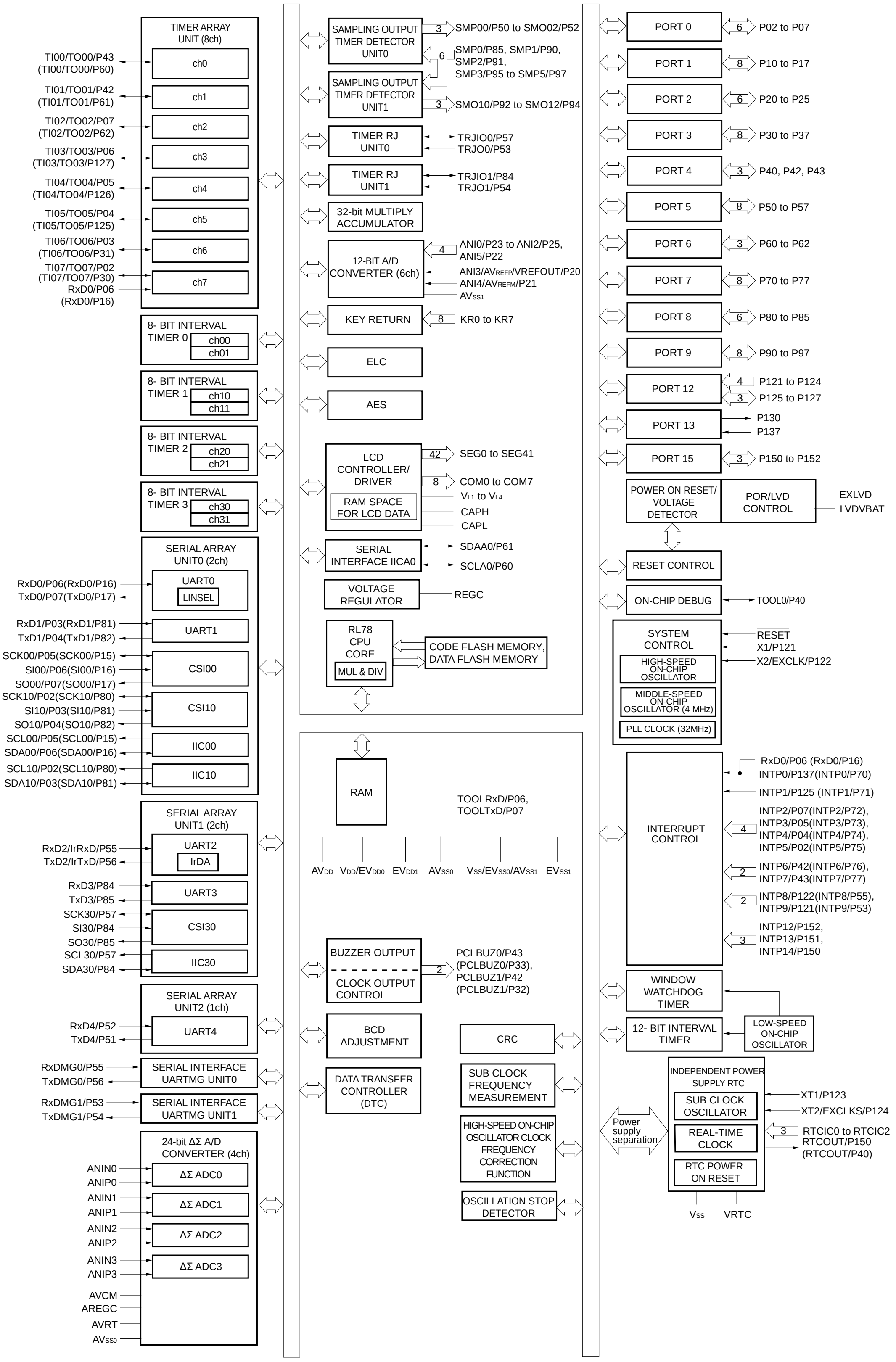
1.5 Block Diagram

1.5.1 80-pin product



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

1.5.2 100-pin product



Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0).**

1.6 Outline of Functions

(1/3)

Item		80-pin	100-pin
		R5F10NMLDFB	R5F10NPLDFB
Code flash memory		512 KB (256 KB × 2 banks)	
Data flash memory		2 KB	
RAM		32 KB[Note]	
Address space		1 MB	
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (high-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V), HS (high-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), HS (high-speed main) mode: 1 to 6 MHz (V_{DD} = 2.1 to 5.5 V), LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V), LV (low-voltage main) mode: 1 to 4 MHz (V_{DD} = 1.6 to 5.5 V), LP (low-power main) mode: 1 MHz (V_{DD} = 1.8 to 5.5 V)	
	High -speed on-chip oscillator clock (f_{IH}) Max.: 32 MHz	HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 2.7 to 5.5 V), HS (high-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), HS (high-speed main) mode: 1 to 6 MHz (V_{DD} = 2.1 to 5.5 V), LS (low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V), LV (low-voltage main) mode: 1 to 4 MHz (V_{DD} = 1.6 to 5.5 V), LP (low-power main) mode: 1 MHz (V_{DD} = 1.8 to 5.5 V)	
	Middle -speed on-chip oscillator clock (f_{IM}) Max.: 4 MHz		
	PLL clock (f_{PLL})	HS (high-speed main) mode: 32 MHz (V_{DD} = 2.7 to 5.5 V)	
Subsystem clock	Subsystem clock oscillator clock (f_{SX})	XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (typ.): V_{DD} = 1.6 to 5.5 V 38.4 kHz (typ.): V_{DD} = 1.6 to 5.5 V	
	Low-speed on-chip oscillator clock (f_{IL})	15 kHz (typ.): V_{DD} = 1.6 to 5.5 V	
High-speed on-chip oscillator clock frequency correction function		Correct the frequency of the high-speed on-chip oscillator clock by the subsystem clock.	
General-purpose register		8 bits × 8 registers × 4 banks	
Minimum instruction execution time		0.03125 μs (PLL clock: f_{PLL} = 32 MHz selection)	
		0.03125 μs (High-speed on-chip oscillator clock: f_{IH} = 32 MHz operation)	
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)	
		66.6 μs (Low-speed on-chip oscillator: f_{IL} = 15 kHz operation)	
Instruction set		● Data transfer (8/16 bits) ● Adder and subtractor/logical operation (8/16 bits) ● Multiplication (16 bits × 16 bits), division (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	60	76
	CMOS I/O	48	64
	CMOS input	5	5
	CMOS output	1	1
	N-ch O.D I/O (6 V tolerance)	6	6

Note This is about 31 KB when the self-programming function is used.

(2/3)

Item		80-pin	100-pin
		R5F10NMLDFB	R5F10NPLDFB
Timer	16-bit timer TAU	8 channels	
	Watchdog timer	1 channel	
	12-bit interval timer	1 channel	
	8/16-bit interval timer	8 channels (8-bit)/4 channels (16-bit)	
	Independent power supply realtime clock (RTC)	1 channel	
	16-bit timer RJ	2 channels	
	Sampling output timer detector (SMOTD)	6 channels for input, 6 channels for output	
	Oscillation stop detection circuit	1 channel	
	Timer output	Timer outputs: 8 channels, PWM outputs: 7[Note]	
	RTC output	1 channel ● 1 Hz/64 Hz (sub clock: f_{SX} = 32.768 kHz)	
	RTC time capture input	3 channels	
Clock output/buzzer output		2	
		● 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation) ● 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Sub clock: f_{SX} = 32.768 kHz operation)	
12-bit resolution A/D converter		4 channels	6 channels
	Simultaneous sampling for 3 channels	—	3 channels
	Voltage reference voltage output	1.5 V/2.0 V/2.5 V	
24-bit ΔΣ A/D converter		3 channels	4 channels
	SNDR	Typ. 80 dB (gain ×1) Min. 69 dB (gain ×16) Min. 65 dB (gain ×32)	
	Sampling frequency	3.906 kHz/1.953 kHz	
	PGA	×1, ×2, ×4, ×8, ×16, ×32	
Serial interface	Simplified SPI (CSI)/ UART/simplified I^{2}C	2 channels	3 channels
	UART/IrDA	1 channel	
	UART	—	1 channel
	I^{2}C bus	1 channel	
	UARTMG	2 channels	
32-bit multiplier and multiply-accumulator		32 bits × 32 bits = 64 bits (Unsigned or signed) (5 clock) 32 bits × 32 bits + 64 bits = 64 bits (Unsigned or signed) (5 clock) 24 cumulative buffer channels	
Data transfer controller (DTC)		46 sources	50 sources

Note The number of outputs varies, depending on the setting of channels in use and the number of the master (see **8.9.3 Operation as multiple PWM output function)**.

(3/3)

Item		80-pin	100-pin
		R5F10NMLDFB	R5F10NPLDFB
Event link controller (ELC)	Event input	7	
	Event trigger input	30	
LCD controller/driver		Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.	
	Segment signal output	34 (30)[Note 1]	42 (38)[Note 1]
	Common signal output	4 (8)[Note 1]	
Vectored interrupt sources	Internal	41	47
	External	14	14
Key interrupt input		8	
AES circuit		Cipher modes of operation: GCM/ECB/CBC Encryption key length: 128/192/256-bit	
Reset	MCU	● Reset by $\overline{\text{RESET}}$ pin ● Internal reset by watchdog timer ● Internal reset by power-on-reset of V_{DD} power supply ● Internal reset by voltage detector of V_{DD} power supply ● Internal reset by illegal instruction execution[Note 2] ● Internal reset by RAM parity error ● Internal reset by illegal-memory access	
	RTC	● RTC circuit reset by RTC Power-on-reset	
Power-on-reset circuit	V_{DD}	● Power-on-reset: 1.51 V (typ.) ● Power-down-reset: 1.50 V (typ.)	
	VRTC	● RTC Power-on-reset: 1.52 V (typ.) ● RTC Power-down-reset: 1.50 V (typ.)	
Voltage detector	V_{DD}	● Rising edge: 1.67 V to 4.06 V (14 stages) ● Falling edge: 1.63 V to 3.98 V (14 stages)	
	V_{DD}	● Rising edge: 2.53 V to 3.77 V (6 stages) ● Falling edge: 2.46 V to 3.70 V (6 stages)	
	LVDVBAT	● Rising edge: 2.23 V to 3.13 V (7 stages) ● Falling edge: 2.17 V to 3.07 V (7 stages)	
	VRTC	● Rising edge: 2.22 V to 2.84 V (4 stages) ● Falling edge: 2.16 V to 2.78 V (4 stages)	
	EXLVD	● Rising edge: 1.33 V ● Falling edge: 1.28 V	
On-chip debug function		Provided	
Bank programming function		Provided	
Power supply voltage		V_{DD} = 1.6 to 5.5 V	
Operating ambient temperature		T_A = –40 to +85°C	

Notes 1. The values in parentheses are the number of signal outputs when 8 com is used.

2. This reset occurs when instruction code FFH is executed.
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

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

Table 2-1. Pin I/O Buffer Power Supplies

(1) 80-pin product

Power Supply	Corresponding Pins
EV_{DD0}	Port pins other than P20 to P23, P121 to P124, P137, and P150 to P152
V_{DD}	● P121, P122, P137, and P150 to P152 ● $\overline{\text{RESET}}$, REGC
AV_{DD}	● P20 to P23 ● ANIP0 to ANIP2, ANIN0 to ANIN2
VRTC	● P123, P124 ● RTCIC0 to RTCIC2

(2) 100-pin product

Power Supply	Corresponding Pins
EV_{DD0}, EV_{DD1}	Port pins other than P20 to P25, P121 to P124, P137, and P150 to P152
V_{DD}	● P121, P122, P137 and P150 to P152 ● $\overline{\text{RESET}}$, REGC
AV_{DD}	● P20 to P25 ● ANIP0 to ANIP3, ANIN0 to ANIN3
VRTC	● P123, P124 ● RTCIC0 to RTCIC2

Caution Make the voltage on the EV_{DD1} pin the same as that on the V_{DD} and EV_{DD0} pins.

Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.

2.1.1 80-pin product

(1/3)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P02	8-5-10	I/O	Digital input invalid[Note 1]	SEG32/SCK10/SCL10/TI07/TO07/INTP5/SMO12	Port 0. 6-bit I/O port. Input/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. Input of P02, P03, P05, and P06 can be set to TTL input buffer. Output of P02 to P07 can be set to N-ch open-drain output (EV$_{DD}$ tolerance).
P03				SEG33/SI10/RxD1/TI06/TO06/SDA10/TRJO1/SMO11	
P04	7-5-10			SEG34/SO10/TxD1/TI05/TO05/INTP4/TRJO0/SMO10	
P05	8-5-10			SEG35/SCK00/SCL00/TI04/TO04/INTP3/TRJIO0	
P06				SEG36/SI00/RxD0/TI03/TO03/SDA00/TOOLRxD/RxDMG1/(INTP9)	
P07	7-5-10			SEG37/SO00/TxD0/TI02/TO02/INTP2/TOOLTxD/TxDMG1	
P10	7-5-4	I/O	Digital input invalid[Note 1]	SEG4	Port 1. 8-bit I/O port. Input/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. Input of P15 and P16 can be set to TTL input buffer. Output of P15 to P17 can be set to N-ch open-drain output (EV$_{DD}$ tolerance). Can be set to LCD output[Note 2].
P11				SEG5	
P12				SEG6	
P13				SEG7	
P14				SEG8	
P15	8-5-10			SEG9/(SCK00)/(SCL00)	
P16				SEG10/(SI00)/(RxD0)/(SDA00)	
P17	7-5-10			SEG11/(SO00)/(TxD0)	
P20	4-3-5	I/O	Analog input port	AV$_{REFP}$/ANI3/VREFOUT	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input[Note 3].
P21				AV$_{REFM}$/ANI4	
P22	4-15-3			ANI5/EXVLD	
P23	4-3-5			ANI0	
P30	7-5-29	I/O	Digital input invalid[Note 1]	SEG24/(TI07)/(TO07)/SMP3	Port 3. 4-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P31				SEG25/(TI06)/(TO06)/SMP2	
P32				SEG26/(PCLBUZ1)/SMP1/SMO01	
P33				SEG27/(PCLBUZ0)/SMP0/SMO00/TI01/TO01	
P40	7-1-3	I/O	Input port	TOOL0/(RTCOUT)	Port 4. 1-bit I/O port. Input/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.

Notes 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

3. Setting digital or analog to each pin can be done in the port mode control register (PMC2).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P55	8-5-10	I/O	Input port	RxD2/IrRxD/RxDMG0/(INTP8)	Port 5. 2-bit I/O port. Input/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. Input of P55 can be set to TTL input buffer. Output of P56 can be set to N-ch open-drain output (EV_{DD} tolerance).
P56	7-5-10			TxD2/IrTxD/TxDMG0	
P60	12-1-2	I/O	Input port	SCLA0/(TI00)/(TO00)	Port 6. 3-bit I/O port. Input/output can be specified in 1-bit units. Outputs from P60 to P62 are N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI01)/(TO01)	
P62				(TI02)/(TO02)/TRJIO1/INTP6	
P70	7-5-4	I/O	Digital input invalid[Note 1]	SEG16/KR0/(INTP0)	Port 7. 8-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P71				SEG17/KR1/(INTP1)	
P72				SEG18/KR2/(INTP2)	
P73				SEG19/KR3/(INTP3)	
P74				SEG20/KR4/(INTP4)	
P75				SEG21/KR5/(INTP5)	
P76				SEG22/KR6/(INTP6)/SMP5	
P77				SEG23/KR7/(INTP7)/SMP4	
P80	8-5-10	I/O	Digital input invalid[Note 1]	SEG12/(SCL10)/(SCK10)	Port 8. 4-bit I/O port. Input/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. Input of P80 and P81 can be set to TTL input buffer. Output of P80 to P82 can be set to N-ch open-drain output (EV_{DD} tolerance). Can be set to LCD output[Note 2].
P81				SEG13/(RxD1)/(SDA10)/(SI10)	
P82	7-5-10			SEG14/(TxD1) /(SO10)	
P83	7-5-4			SEG15	
P90	7-5-29	I/O	Digital input invalid[Note 1]	COM0	Port 9. 8-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P91				COM1	
P92				COM2	
P93				COM3	
P94				COM4/SEG0	
P95				COM5/SEG1	
P96				COM6/SEG2	
P97				COM7/SEG3	

Notes 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P121	2-2-1	Input	Input port	X1/INTP9	Port 12. 3-bit I/O port and 4-bit input-only port. For pins P125 to P127, input/output can be specified in 1-bit units. For pins P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. For pins P125 to P127, can be set to LCD output[Note 2].
P122				X2/EXCLK/INTP8	
P123	2-2-2			XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid[Note 1]	V_{L3}/INTP1/(TI05)/(TO05)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)	
P130	1-1-4	Output	Output port	PCLBUZ1/SMO02	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-2	Input	Input port	INTP0	
P150	12-1-6	I/O	Input port	RTCOUT/RTCIC0/ INTP14	Port 15. 3-bit I/O port. Input/output can be specified in 1-bit units. Outputs from P150 to P152 are N-ch open-drain output (6 V tolerance).
P151				RTCIC1/INTP13	
P152				RTCIC2/INTP12	
$\overline{\text{RESET}}$	3-1-1	Input	–	–	Input only pin for external reset. Use of an on-chip pull-up resistor can be always specified. When external reset is not used, leave open.

Notes 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

2.1.2 100-pin product

(1/4)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P02	8-5-10	I/O	Input port	SCK10/SCL10/TI07/TO07/INTP5	Port 0. 6-bit I/O port. Input/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. Input of P02, P03, P05, and P06 can be set to TTL input buffer. Output of P02 to P07 can be set to N-ch open-drain output (EV_{DD} tolerance).
P03				SI10/RxD1/TI06/TO06/SDA10	
P04	7-5-10			SO10/TxD1/TI05/TO05/INTP4	
P05	8-5-10			SCK00/SCL00/TI04/TO04/INTP3	
P06				SI00/RxD0/TI03/TO03/SDA00/TOOLRxD	
P07	7-5-10			SO00/TxD0/TI02/TO02/INTP2/TOOLTxD	
P10	7-5-4	I/O	Digital input invalid[Note 1]	SEG4	Port 1. 8-bit I/O port. Input/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. Input of P15 and P16 can be set to TTL input buffer. Output of P15 to P17 can be set to N-ch open-drain output (EV_{DD} tolerance). Can be set to LCD output[Note 2].
P11				SEG5	
P12				SEG6	
P13				SEG7	
P14				SEG8	
P15	8-5-10			SEG9/(SCK00)/(SCL00)	
P16				SEG10/(SI00)/(RxD0)/(SDA00)	
P17	7-5-10			SEG11/(SO00)/(TxD0)	
P20	4-3-5	I/O	Analog input	AV_{REFP}/ANI3/VREFOUT	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input[Note 3].
P21				AV_{REFM}/ANI4	
P22	4-15-3			ANI5/EXLVD	
P23	4-3-5			ANI0	
P24				ANI1	
P25				ANI2	

Notes 1. "Digital input invalid" refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

3. Setting digital or analog to each pin can be done in the port mode control register (PMC2).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(2/4)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P30	7-5-29	I/O	Digital input invalid[Note 1]	SEG24/(TI07)/(TO07)/(COM7/SEG3)	Port 3. 8-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P31				SEG25/(TI06)/(TO06)/(COM6/SEG2)	
P32				SEG26/(PCLBUZ1)/(COM5/SEG1)	
P33				SEG27/(PCLBUZ0)/(COM4/SEG0)	
P34				SEG28/(COM3)	
P35				SEG29/(COM2)	
P36				SEG30/(COM1)	
P37				SEG31/(COM0)	
P40	7-1-3	I/O	Input port	TOOL0/(RTCOUT)	Port 4. 3-bit I/O port. Input/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. Input of P42 and P43 can be set to TTL input buffer.
P42	8-1-3			TI01/TO01/PCLBUZ1/INTP6	
P43				TI00/TO00/PCLBUZ0/INTP7	
P50	7-5-4	I/O	Digital input invalid[Note 1]	SEG32/SMO00	Port 5. 8/-bit I/O port. Input/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. Input of P52, P53, P55, and P57 can be set to TTL input buffer. Output of P51, P54, P56, and P57 can be set to N-ch open-drain output (EV_{DD} tolerance). Can be set to LCD output[Note 2].
P51	7-5-10			SEG33/SMO01/TxD4	
P52	8-5-10			SEG34/SMO02/RxD4	
P53				SEG35/TRJO0/RxDMG1/(INTP9)	
P54	7-5-10			SEG36/TRJO1/TxDMG1	
P55	8-5-10			SEG37/RxD2/IrRxD/RxDMG0/(INTP8)	
P56	7-5-10			SEG38/TxD2/IrTxD/TxDMG0	
P57	8-5-10			SEG39/SCK30/SCL30/TRJIO0	
P60	12-1-2	I/O	Input port	SCLA0/(TI00)/(TO00)	Port 6. 3-bit I/O port. Input/output can be specified in 1-bit units. Outputs from P60 to P62 are N-ch open-drain output (6 V tolerance).
P61				SDAA0/(TI01)/(TO01)	
P62				(TI02)/(TO02)	

Notes 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(3/4)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P70	7-5-4	I/O	Digital input invalid[Note 1]	SEG16/KR0/(INTP0)	Port 7. 8-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P71				SEG17/KR1/(INTP1)	
P72				SEG18/KR2/(INTP2)	
P73				SEG19/KR3/(INTP3)	
P74				SEG20/KR4/(INTP4)	
P75				SEG21/KR5/(INTP5)	
P76				SEG22/KR6/(INTP6)	
P77				SEG23/KR7/(INTP7)	
P80	8-5-10	I/O	Digital input invalid[Note 1]	SEG12/(SCL10)/(SCK10)	Port 8. 6-bit I/O port. Input/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. Input of P80, P81, and P84 can be set to TTL input buffer. Output of P80 to P82, P84, and P85 can be set to N-ch open-drain output (EV_{DD} tolerance). Can be set to LCD output [Note 2].
P81				SEG13/(RxD1)/(SDA10)/(SI10)	
P82	7-5-10			SEG14/(TxD1) /(SO10)	
P83	7-5-4			SEG15	
P84	8-5-10			SI30/RxD3/SDA30/SEG40/TRJIO1	
P85	7-5-10			SO30/TxD3/SEG41/SMP0	
P90	7-5-29	I/O	Digital input invalid[Note 1]	COM0/(SEG31)/SMP1	Port 9. 8-bit I/O port. Input/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. Can be set to LCD output[Note 2].
P91				COM1/(SEG30)/SMP2	
P92				COM2/(SEG29)/SMO10	
P93				COM3/(SEG28)/SMO11	
P94				COM4/SEG0/(SEG27)/SMO12	
P95				COM5/SEG1/(SEG26)/SMP3	
P96				COM6/SEG2/(SEG25)/SMP4	
P97				COM7/SEG3/(SEG24)/SMP5	
P121	2-2-1	Input	Input port	X1/INTP9	Port 12. 3-bit I/O port and 4-bit input only port. For only P125 to P127, input/output can be specified in 1-bit units. For only P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. P125 to P127 can be set to LCD output [Note 2].
P122				X2/EXCLK/INTP8	
P123	2-2-2			XT1	
P124				XT2/EXCLKS	
P125	7-5-6	I/O	Digital input invalid[Note 1]	V_{L3}/INTP1/(TI05)/(TO05)	
P126	7-5-5			CAPL/(TI04)/(TO04)	
P127				CAPH/(TI03)/(TO03)	

Notes 1. “Digital input invalid” refers to the state in which all the digital outputs, digital inputs, and LCD outputs are disabled.

2. Digital or LCD for each pin can be selected with the port mode register x (PMx) and the LCD port function register x (PFSEGx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(4/4)

Function Name	Pin Type	I/O	After Reset Released	Alternate Function	Function
P130	1-1-4	Output	Output port	–	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-2	Input	Input port	INTP0	
P150	12-1-6	I/O	Input port	RTCOUT/RTCIC0/ INTP14	Port 15. 3-bit I/O port. Input/output can be specified in 1-bit units. Outputs from P150 to P152 are N-ch open-drain output (6 V tolerance).
P151				RTCIC1/INTP13	
P152				RTCIC2/INTP12	
$\overline{\text{RESET}}$	3-1-1	Input	–	–	Input only pin for external reset. Use of an on-chip pull-up resistor can be always specified. When external reset is not used, leave open.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

2.2 Functions Other than Port Functions

2.2.1 Functions mounted on the products with different numbers of pins

(1/2)

Function Name	100-pin	80-pin	Function Name	100-pin	80-pin	Function Name	100-pin	80-pin
ANI0	√	√	RTCOUT	√	√	SMO01	√	√
ANI1	√	–	REGC	√	√	SMO02	√	√
ANI2	√	–	$\overline{\text{RESET}}$	√	√	SMO10	√	√
ANI3	√	√	RxD0	√	√	SMO11	√	√
ANI4	√	√	RxD1	√	√	SMO12	√	√
ANI5	√	√	RxD2	√	√	IrRxD	√	√
ANIN0	√	√	RxD3	√	–	IrTxD	√	√
ANIN1	√	√	RxD4	√	–	TI00	√	√
ANIN2	√	√	RxDMG0	√	√	TI01	√	√
ANIN3	√	–	RxDMG1	√	√	TI02	√	√
ANIP0	√	√	TxD0	√	√	TI03	√	√
ANIP1	√	√	TxD1	√	√	TI04	√	√
ANIP2	√	√	TxD2	√	√	TI05	√	√
ANIP3	√	–	TxD3	√	–	TI06	√	√
INTP0	√	√	TxD4	√	–	TI07	√	√
INTP1	√	√	TxDMG0	√	√	TO00	√	√
INTP2	√	√	TxDMG1	√	√	TO01	√	√
INTP3	√	√	SCK00	√	√	TO02	√	√
INTP4	√	√	SCK10	√	√	TO03	√	√
INTP5	√	√	SCK30	√	–	TO04	√	√
INTP6	√	√	SI00	√	√	TO05	√	√
INTP7	√	√	SI10	√	√	TO06	√	√
INTP8	√	√	SI30	√	–	TO07	√	√
INTP9	√	√	SO00	√	√	TRJIO0	√	√
INTP12	√	√	SO10	√	√	TRJIO1	√	√
INTP13	√	√	SO30	√	–	TRJO0	√	√
INTP14	√	√	SCL00	√	√	TRJO1	√	√
KR0	√	√	SCL10	√	√	V_{L1}	√	√
KR1	√	√	SCL30	√	–	V_{L2}	√	√
KR2	√	√	SDA00	√	√	V_{L3}	√	√
KR3	√	√	SDA10	√	√	V_{L4}	√	√
KR4	√	√	SDA30	√	–	CAPH	√	√
KR5	√	√	SDAA0	√	√	CAPL	√	√
KR6	√	√	SCLA0	√	√	X1	√	√
KR7	√	√	SMP0	√	√	X2	√	√
RTCIC0	√	√	SMP1	√	√	EXCLK	√	√
RTCIC1	√	√	SMP2	√	√	XT1	√	√
RTCIC2	√	√	SMP3	√	√	XT2	√	√
EXLVD	√	√	SMP4	√	√	EXCLKS	√	√
PCLBUZ0	√	√	SMP5	√	√	V_{DD}	√	√
PCLBUZ1	√	√	SMO00	√	√	EV_{DD0}	√	√

(2/2)

Function Name	100-pin	80-pin	Function Name	100-pin	80-pin	Function Name	100-pin	80-pin
EV$_{DD1}$	√	–	COM5	√	√	SEG20	√	√
LVDVBAT	√	√	COM6	√	√	SEG21	√	√
VREFOUT	√	√	COM7	√	√	SEG22	√	√
VRTC	√	√	SEG0	√	√	SEG23	√	√
AV$_{REFP}$	√	√	SEG1	√	√	SEG24	√	√
AV$_{REFM}$	√	√	SEG2	√	√	SEG25	√	√
V$_{SS}$	√	√	SEG3	√	√	SEG26	√	√
EV$_{SS0}$	√	√	SEG4	√	√	SEG27	√	√
EV$_{SS1}$	√	–	SEG5	√	√	SEG28	√	–
AVRT	√	√	SEG6	√	√	SEG29	√	–
AVCM	√	√	SEG7	√	√	SEG30	√	–
AV$_{DD}$	√	√	SEG8	√	√	SEG31	√	–
AREGC	√	√	SEG9	√	√	SEG32	√	√
AV$_{SS0}$	√	√	SEG10	√	√	SEG33	√	√
AV$_{SS1}$	√	√	SEG11	√	√	SEG34	√	√
TOOLRxD	√	√	SEG12	√	√	SEG35	√	√
TOOLTxD	√	√	SEG13	√	√	SEG36	√	√
TOOL0	√	√	SEG14	√	√	SEG37	√	√
COM0	√	√	SEG15	√	√	SEG38	√	–
COM1	√	√	SEG16	√	√	SEG39	√	–
COM2	√	√	SEG17	√	√	SEG40	√	–
COM3	√	√	SEG18	√	√	SEG41	√	–
COM4	√	√	SEG19	√	√			

2.2.2 Description of Functions

(1/2)

Function Name	I/O	Function
ANI0 to ANI5	Input	12-bit A/D converter analog input (see **Figure 17-33 Sample Protection Circuit for Analog Inputs**)
ANIN0 to ANIN3	Input	24-bit ΔΣ A/D converter analog input. These are the negative input pins.
ANIP0 to ANIP3	Input	24-bit ΔΣ A/D converter analog input. These are the positive input pins.
INTP0 to INTP9, INTP12 to INTP14	Input	External interrupt request input. Specified the valid edge: Rising edge, falling edge, or both rising and falling edges.
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	–	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
RTCOUT	Output	Independent power supply RTC (1 Hz/64 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. An on-chip pull-up resistor is always valid. When the external reset pin is not used, leave open.
RxD0 to RxD4	Input	Serial data input pins of serial interfaces UART0 to UART4
RxDMG0, RxDMG1	Input	Serial data input pins of serial interfaces UARTMG0 and UARTMG1
TxD0 to TxD4	Output	Serial data output pins of serial interfaces UART0 to UART4
TxDMG0, TxDMG1	Output	Serial data output pins of serial interfaces UARTMG0 and UARTMG1
SCK00, SCK10, SCK30	I/O	Serial clock I/O pins of serial interfaces CSI00, CSI10, and CSI30
SI00, SI10, SI30	Input	Serial data input pins of serial interfaces CSI00, CSI10, and CSI30
SO00, SO10, SO30	Output	Serial data output pins of serial interfaces CSI00, CSI10, and CSI30
IrRxD	Input	Receive data for IrDA
IrTxD	Output	Transmit data for IrDA
SCL00, SCL10, SCL30	Output	Serial clock output pins of serial interfaces IIC00, IIC10, and IIC30
SDA00, SDA10, SDA30	I/O	Serial data I/O pins of serial interfaces IIC00, IIC10, and IIC30
SCLA0	I/O	Serial clock I/O pin of serial interface IICA0
SDAA0	I/O	Serial data I/O pin of serial interface IICA0
SMP0 to SMP5	Input	Sampling inputs
SMO00 to SMO02, SMO10 to SMO12	Output	Sampling clock outputs
TRJIO0, TRJIO1	I/O	Timer RJ I/O
TRJO0, TRJO1	Output	Timer RJ outputs
TI00 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	Output	Timer output pins of 16-bit timers 00 to 07
V_{L1} to V_{L4}	–	LCD drive voltage
CAPH, CAPL	–	Connecting a capacitor for LCD controller/driver

(2/2)

Function Name	I/O	Function
X1, X2	–	If a 24-bit ΔΣ A/D converter is used for external clock input, a 12 MHz oscillator must be connected.
EXCLK	Input	External clock input for main system clock
XT1, XT2	–	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock
V_{DD}	–	<80-pin, 100-pin> Positive power supply for port pins (P121, P122, P137, and P150 to P152) and pins other than RTCIC0 to RTCIC2 which in use for functions other than port functions
EV_{DD0}, EV_{DD1}	–	Positive power supply for port pins (other than P20 to P25, P121 to P124, P137, and P150 to P152)
LVDVBAT	Input	Battery backup power supply voltage detection pin
VRTC	–	Power supply for RTC. Positive power supply for port pins (P123 and P124) Positive power supply for RTCIC0 to RTCIC2 pins
VREFOUT	Output	Voltage reference output
AV_{REFP}	Input	Positive reference voltage input of the 12-bit A/D converter
AV_{REFM}	Input	Negative reference voltage input of the 12-bit A/D converter
V_{SS}	–	Ground voltage for port pins (P121 to P124, P137, and P150 to P152) and pins in use for functions other than port functions
EV_{SS0}, EV_{SS1}	–	Ground voltage for port pins (other than P20 to P25, P121 to P124, P137, and P150 to P152)
AVRT	–	Reference voltage for 24-bit ΔΣ A/D converter
AVCM	–	Common mode voltage for 24-bit ΔΣ A/D converter
AREGC	–	Regulator capacitance for 24-bit ΔΣ A/D converter
AV_{DD}	–	Power supply for 12-bit A/D converter and 24-bit ΔΣ A/D converter <80-pin> Positive power supply for port pins (P20 to P23) <100-pin> Positive power supply for port pins (P20 to P25)
AV_{SS0}	–	Ground voltage for 24-bit ΔΣ A/D converter
AV_{SS1}	–	Ground voltage for 12-bit A/D converter <80-pin> Ground voltage for port pins (P20 to P23) <100-pin> Ground voltage for port pins (P20 to P25)
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
EXLVD	Input	Low voltage detector for external pin
RTCIC0 to RTCIC2	Input	RTC time capture event input
KR0 to KR7	Input	Key interrupt input
SEG0 to SEG41	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.

Table 2-2. Relationships between the Voltage on P40/TOOL0 and Operating Mode After Release from the Reset State

P40/TOOL0	Operating Mode
EV_{DD0}	Normal operating mode
0 V	Flash memory programming mode

For details, see 38.4 Programming Method.

Remarks 1. Use bypass capacitors (about 0.1 μF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS}, EV_{DD0} to EV_{SS0}, and EV_{DD1} to EV_{SS1} lines.

2. For products that do not have an EV_{DD1} or EV_{SS1} pin, read EV_{DD1} as V_{DD} or EV_{DD0}, and EV_{SS1} as V_{SS} or EV_{SS0}.

2.3 Connection of Unused Pins

Table 2-3 shows the connections of unused pins.

Remark The pins mounted depend on the product. See **1.3 Pin Configuration (Top View)** and **2.1 Port Function.**

Table 2-3. Connection of Unused Pins (1/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P02 to P07	I/O	Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Leave open.
P10 to P17		<When setting to port I/O> Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P20 to P25		Input: Independently connect to AV_{DD} or AV_{SS1} via a resistor. Output: Leave open.
P30 to P37		<When setting to port I/O> Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P40/TOOL0		Input: Independently connect to EV_{DD0} or EV_{DD1} via a resistor or leave open. Output: Leave open.
P42, P43		Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Leave open.
P50 to P57		<When setting to port I/O> Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P60 to P62		Input: Independently connect to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor. Output: Set the port's output latch to 0 and leave the pin open, or set the port's output latch to 1 and independently connect the pin to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via a resistor.

Remark For products that do not have an EV_{DD1} or EV_{SS1} pin, read EV_{DD1} as V_{DD} or EV_{DD0}, and EV_{SS1} as V_{SS} or EV_{SS0}.

Table 2-3. Connection of Unused Pins (2/2)

Pin Name	I/O	Recommended Connection of Unused Pins
P70 to P77	I/O	<When setting to port I/O> Input: Independently connect to EV$_{DD0}$, EV$_{DD1}$ or EV$_{SS0}$, EV$_{SS1}$ via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P80 to P85		
P90 to P97		
P121, P122	Input	Independently connect to V$_{DD}$ or V$_{SS}$ via a resistor.
P123, P124	Input	Independently connect to V$_{SS}$ via a resistor.
P125 to P127	I/O	Input: Independently connect to EV$_{DD0}$, EV$_{DD1}$ or EV$_{SS0}$, EV$_{SS1}$ via a resistor. Output: Leave open.
P130	Output	Leave open.
P137	Input	Independently connect to V$_{DD}$ or V$_{SS}$ via a resistor.
<R> P150 to P152	I/O	Set these pins to the input mode, and individually connect each of them via a resistor to V$_{SS}$.
$\overline{\text{RESET}}$	Input	Leave open.
REGC	–	Connect to V$_{SS}$ via capacitor (0.47 to 1 μF).
COM0 to COM7	Output	Leave open.
ANIP0 to ANIP3	Input	Leave open.
ANIN0 to ANIN3		
V$_{L1}$, V$_{L2}$, V$_{L4}$	–	Leave open.
LVDVBAT	–	Connect directly to V$_{SS}$.
VRTC	–	Directly connect to V$_{SS}$.
AVRT, AVCM	–	Connect to AV$_{SS0}$ via capacitor (0.47 μF).
AV$_{SS0}$	–	Make the voltage on AV$_{SS0}$ the same as that on V$_{SS}$.
AV$_{SS1}$	–	Make the voltage on AV$_{SS1}$ the same as that on V$_{SS}$.
AREGC	–	Connect to AV$_{SS0}$ via capacitor (0.47 μF).

Remark For products that do not have an EV$_{DD1}$ or EV$_{SS1}$ pin, read EV$_{DD1}$ as V$_{DD}$ or EV$_{DD0}$, and EV$_{SS1}$ as V$_{SS}$ or EV$_{SS0}$.

2.4 Block Diagrams of Pins

Figures 2-1 to **2-17** show the block diagrams of the pins described in **2.1.1 80-pin product** and **2.1.2 100-pin product**. For the 80-pin product, read EV_{DD1} and EV_{SS1} as EV_{DD0} and EV_{SS0}, respectively.

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

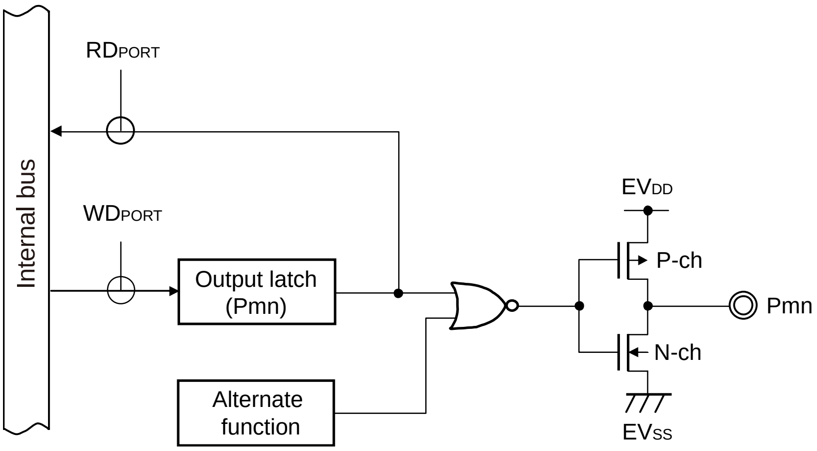


Figure 2-2. Pin Block Diagram for Pin Type 2-1-2

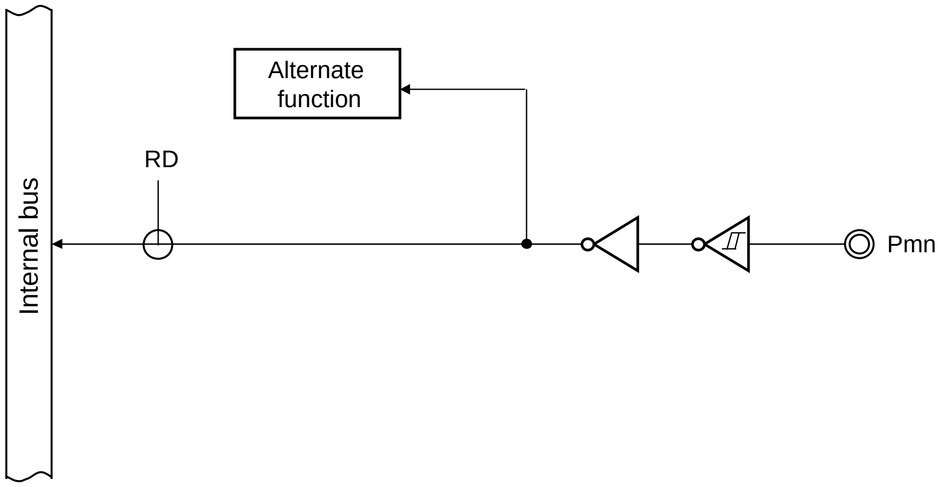
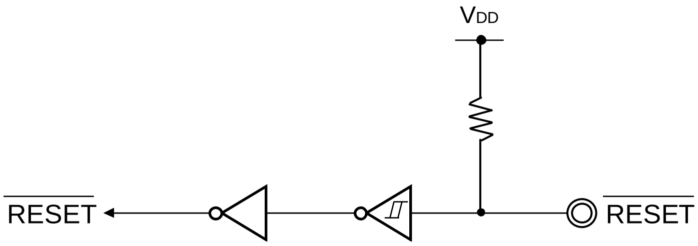
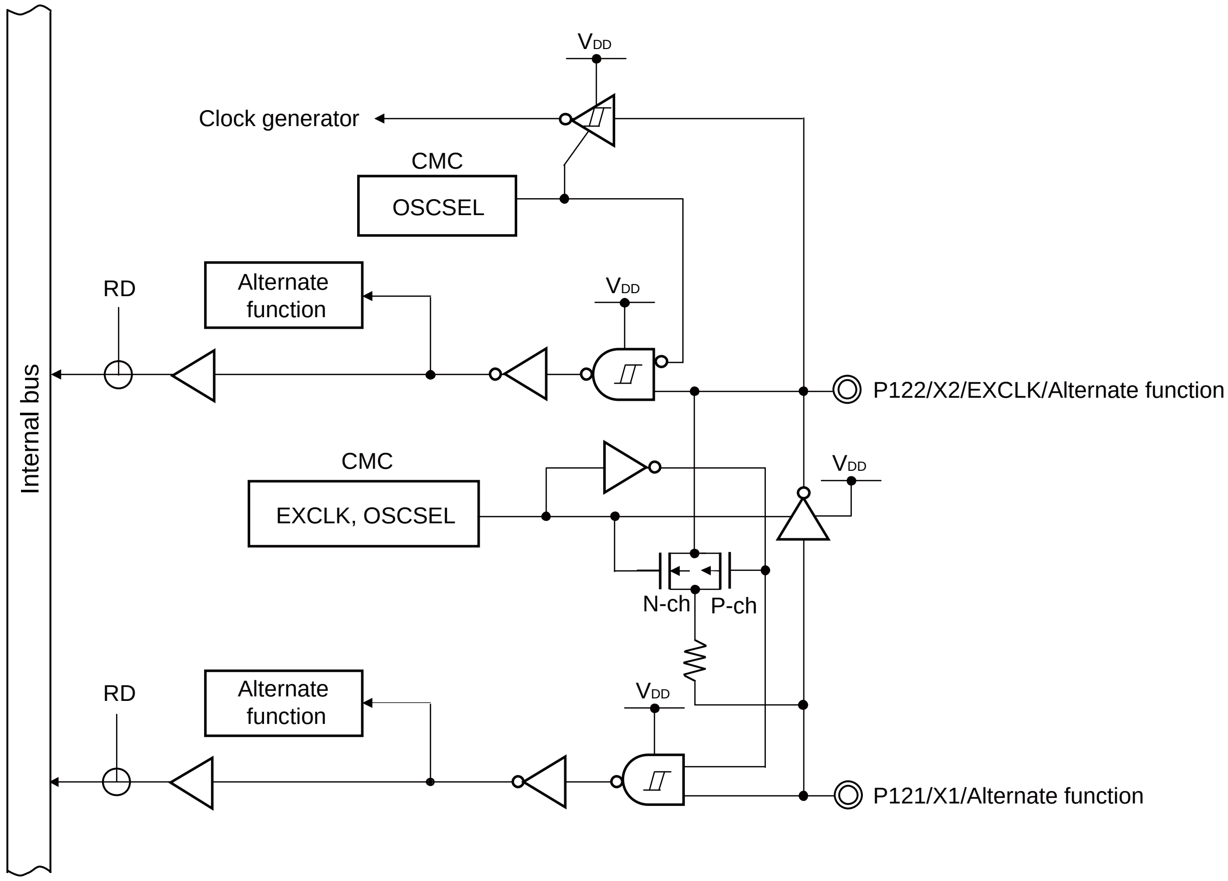


Figure 2-3. Pin Block Diagram for Pin Type 3-1-1



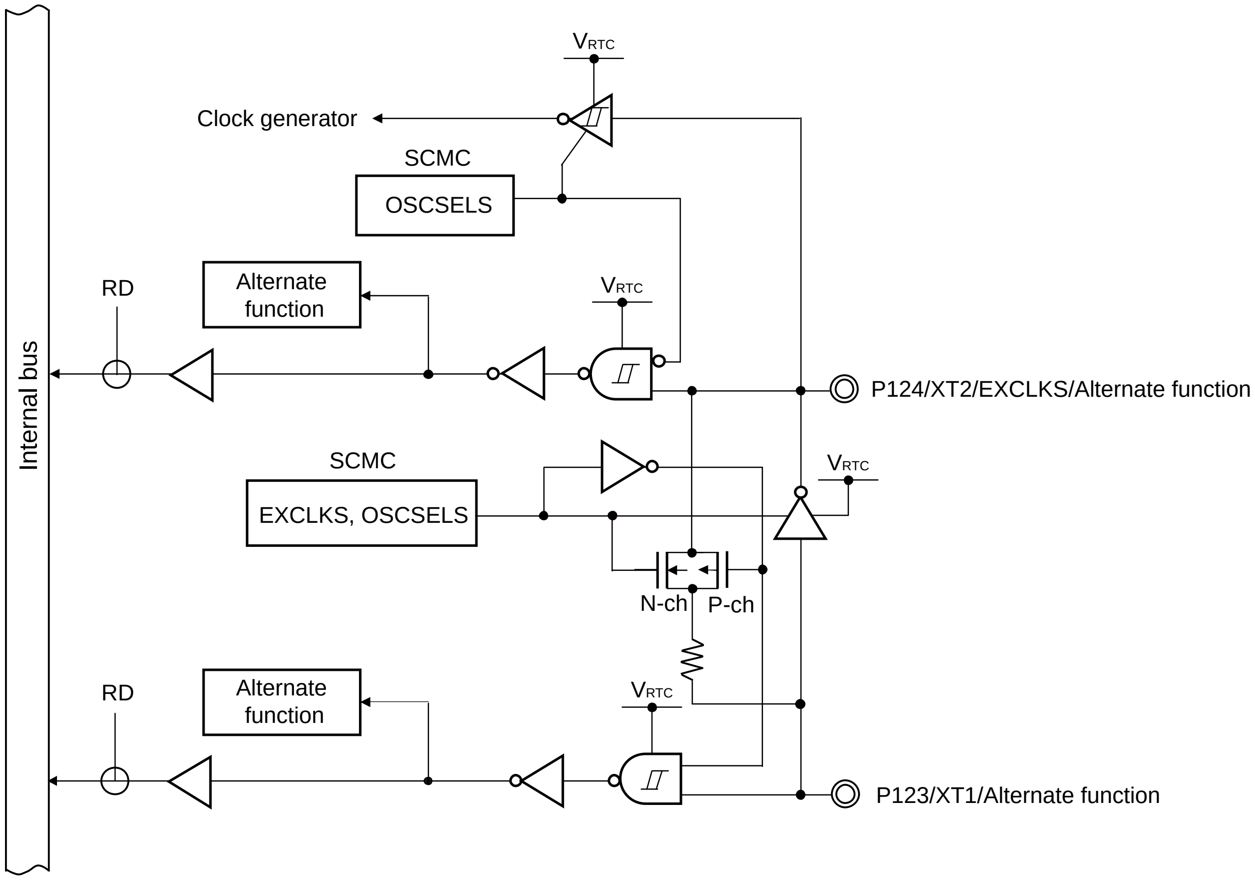
Remark For alternate functions, see **2.1 Port Function**.

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



Remark For alternate functions, see **2.1 Port Function**.

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



Remark For alternate functions, see **2.1 Port Function**.

Figure 2-6. Pin Block Diagram for Pin Type 4-3-5

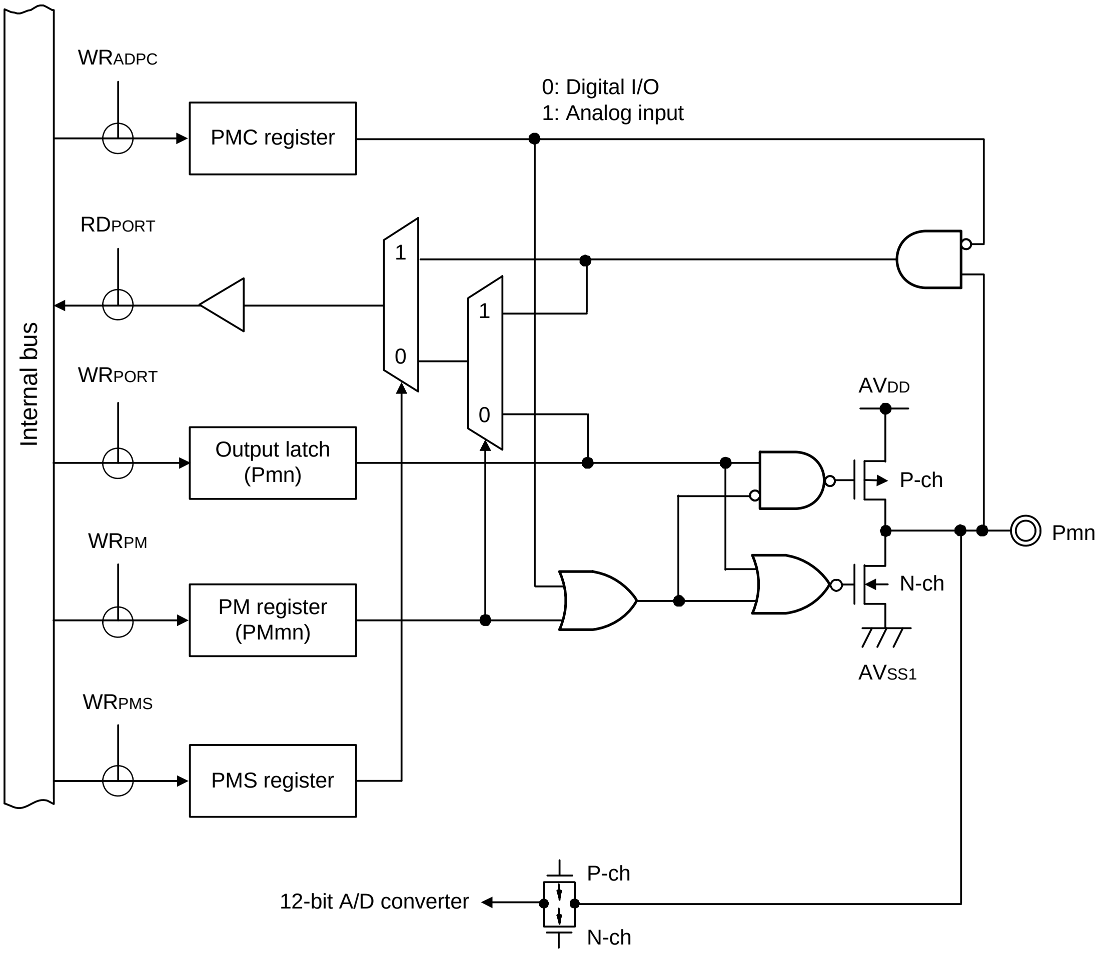


Figure 2-7. Pin Block Diagram for Pin Type 4-15-3

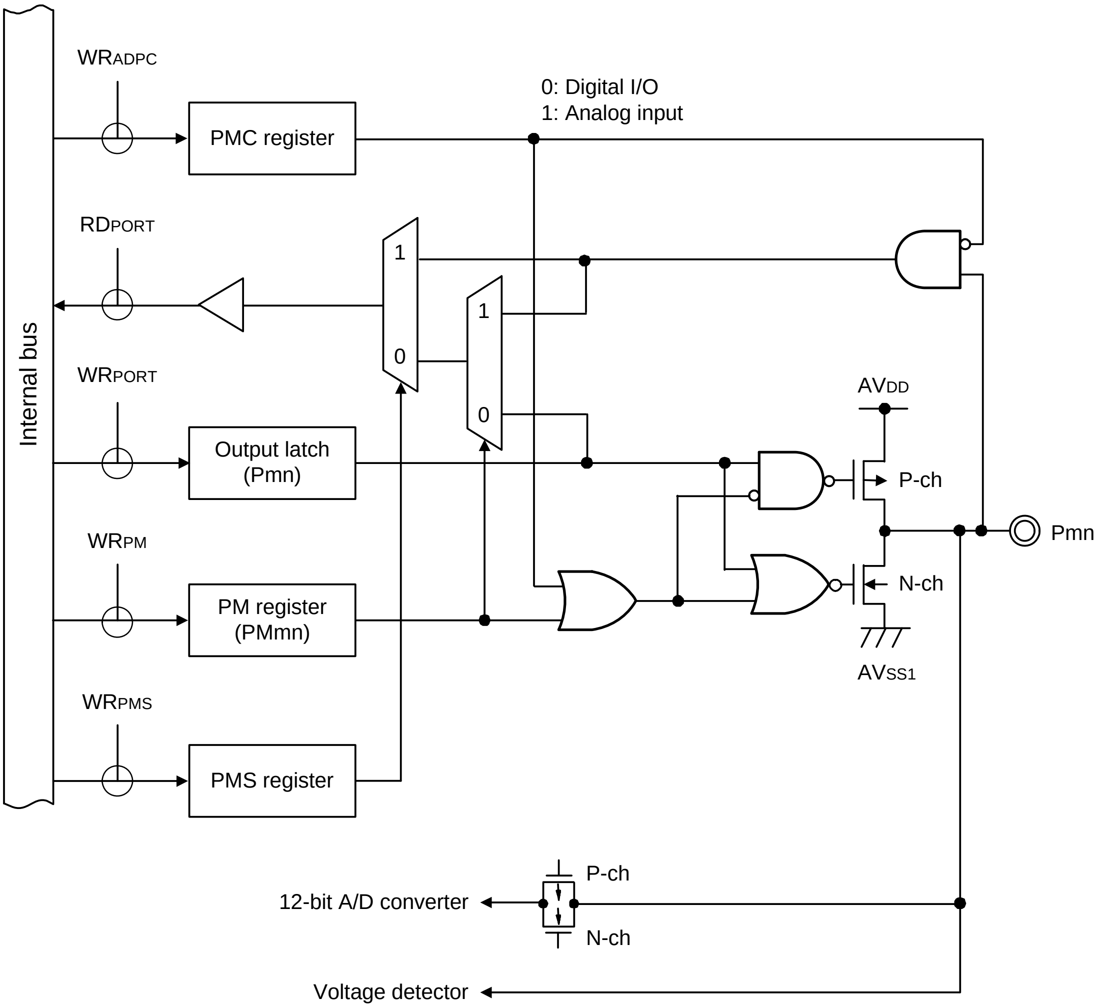
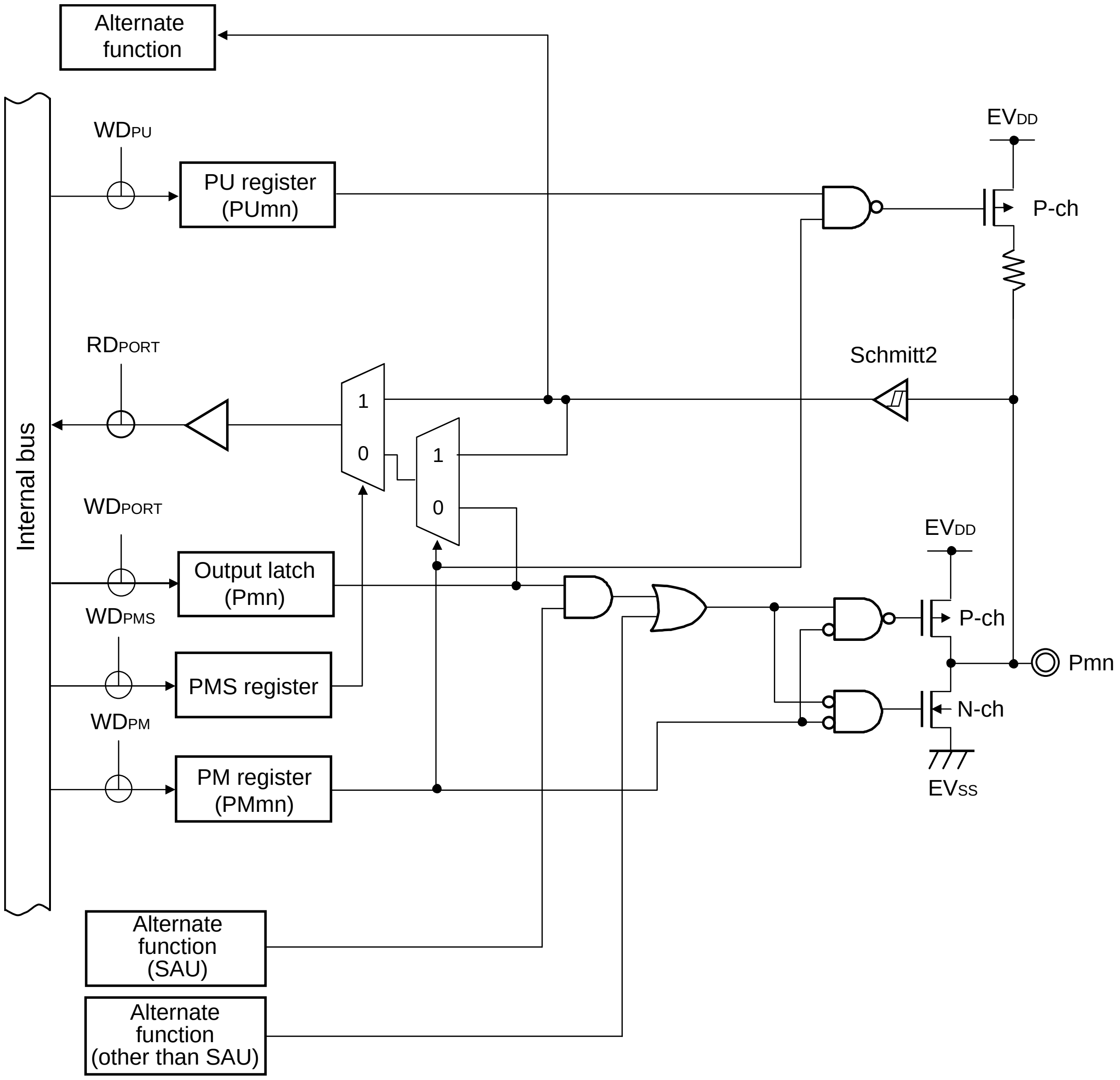


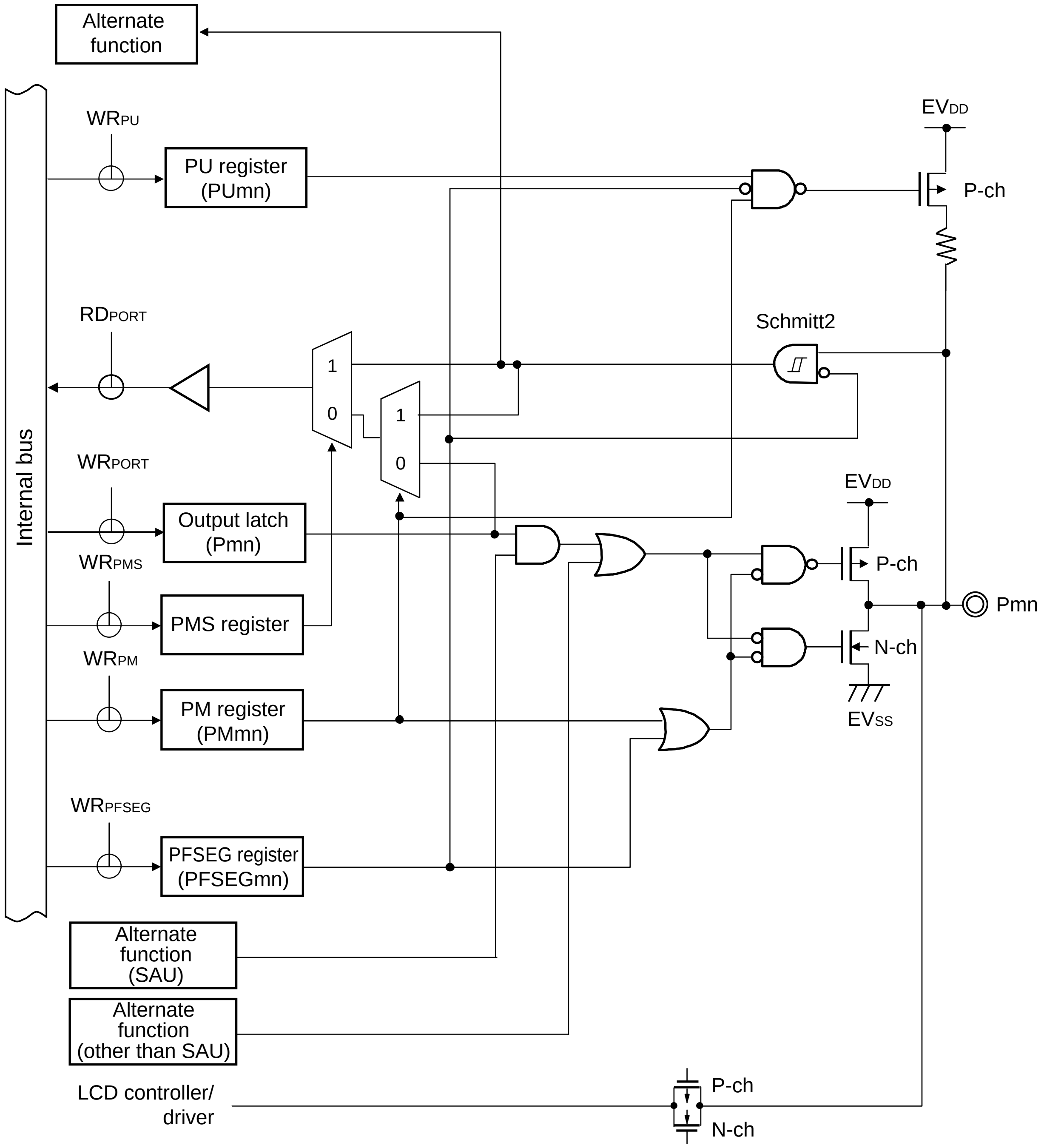
Figure 2-8. Pin Block Diagram for Pin Type 7-1-3



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

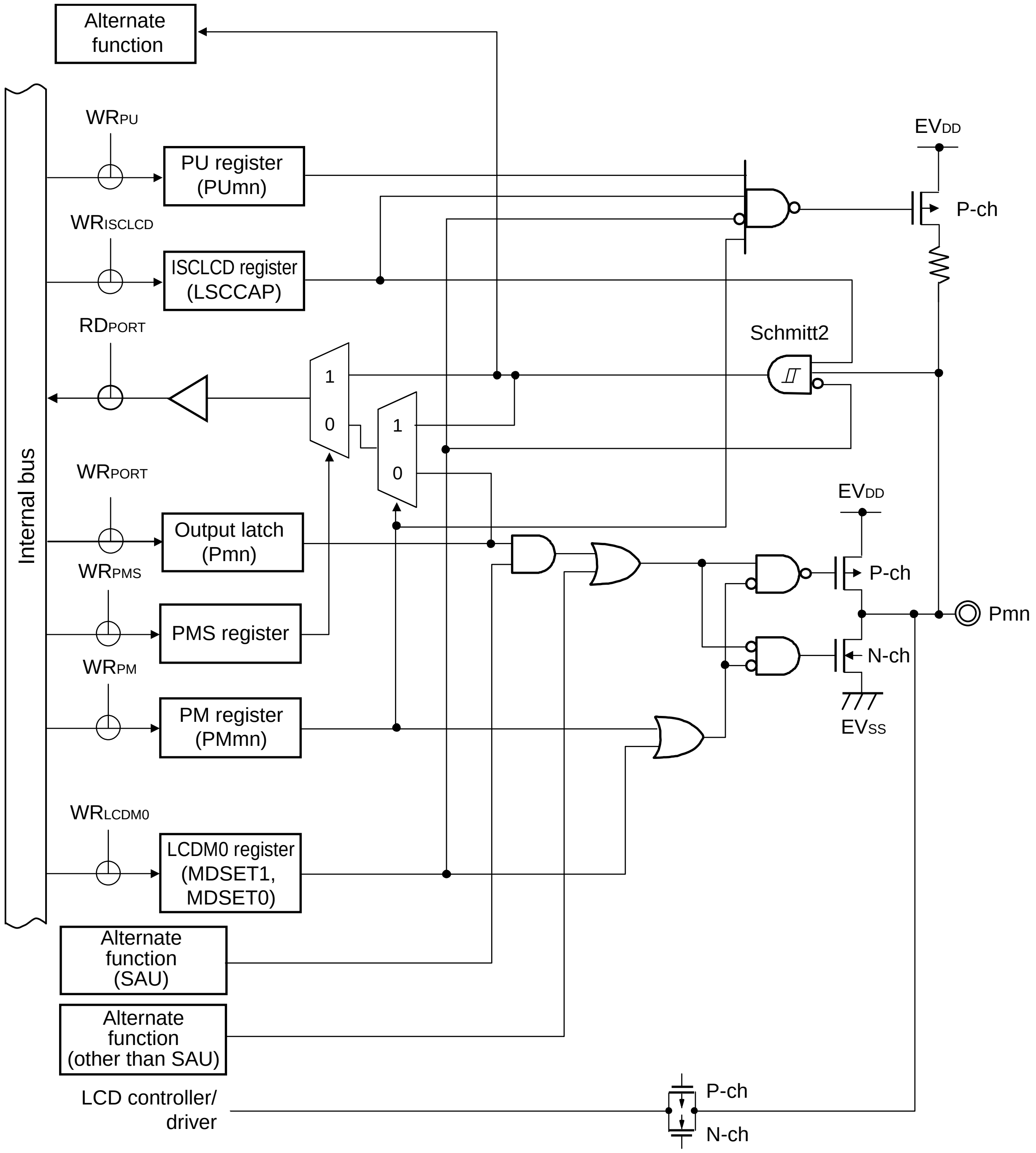
Figure 2-9. Pin Block Diagram for Pin Type 7-5-4



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

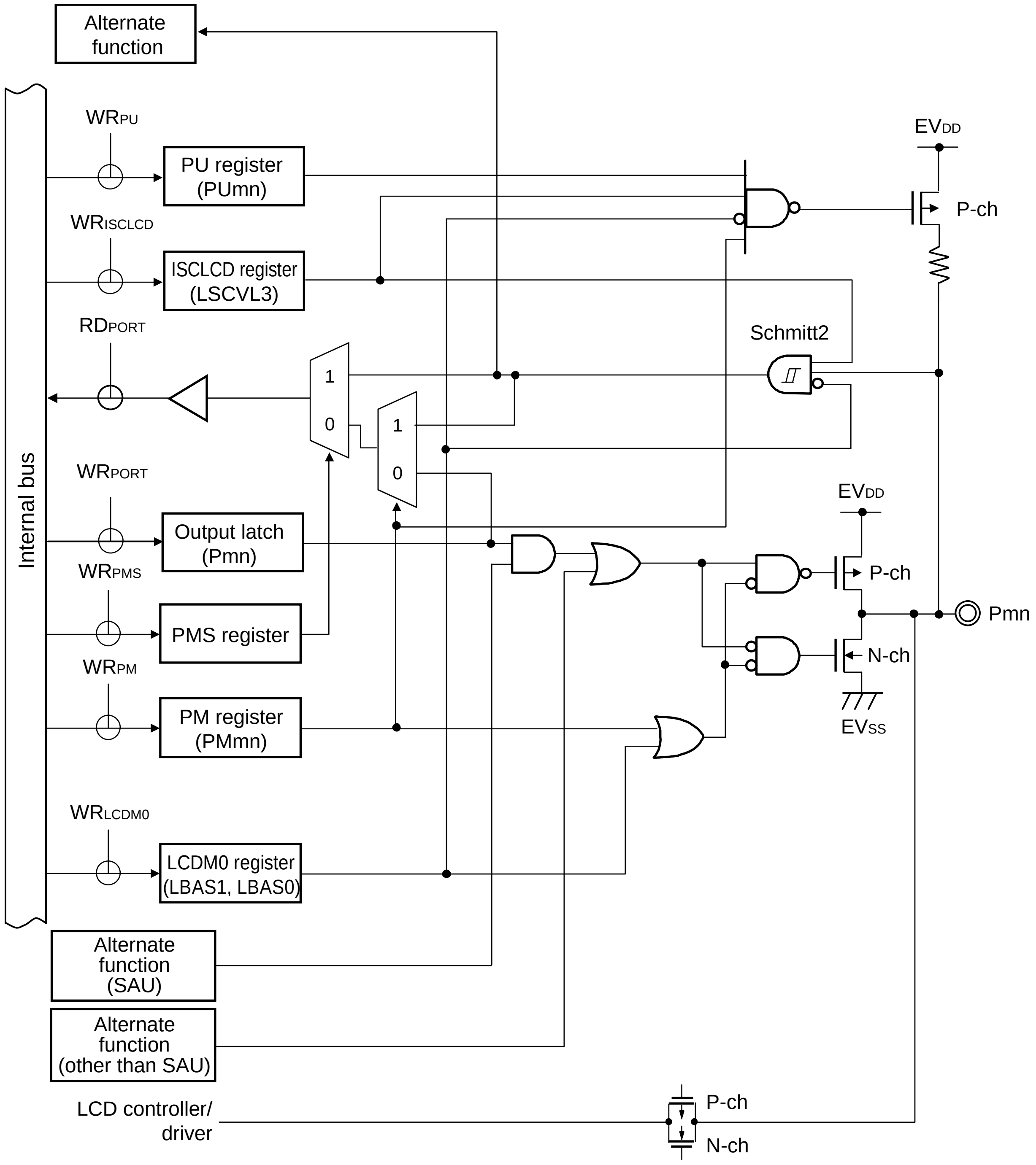
Figure 2-10. Pin Block Diagram for Pin Type 7-5-5



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

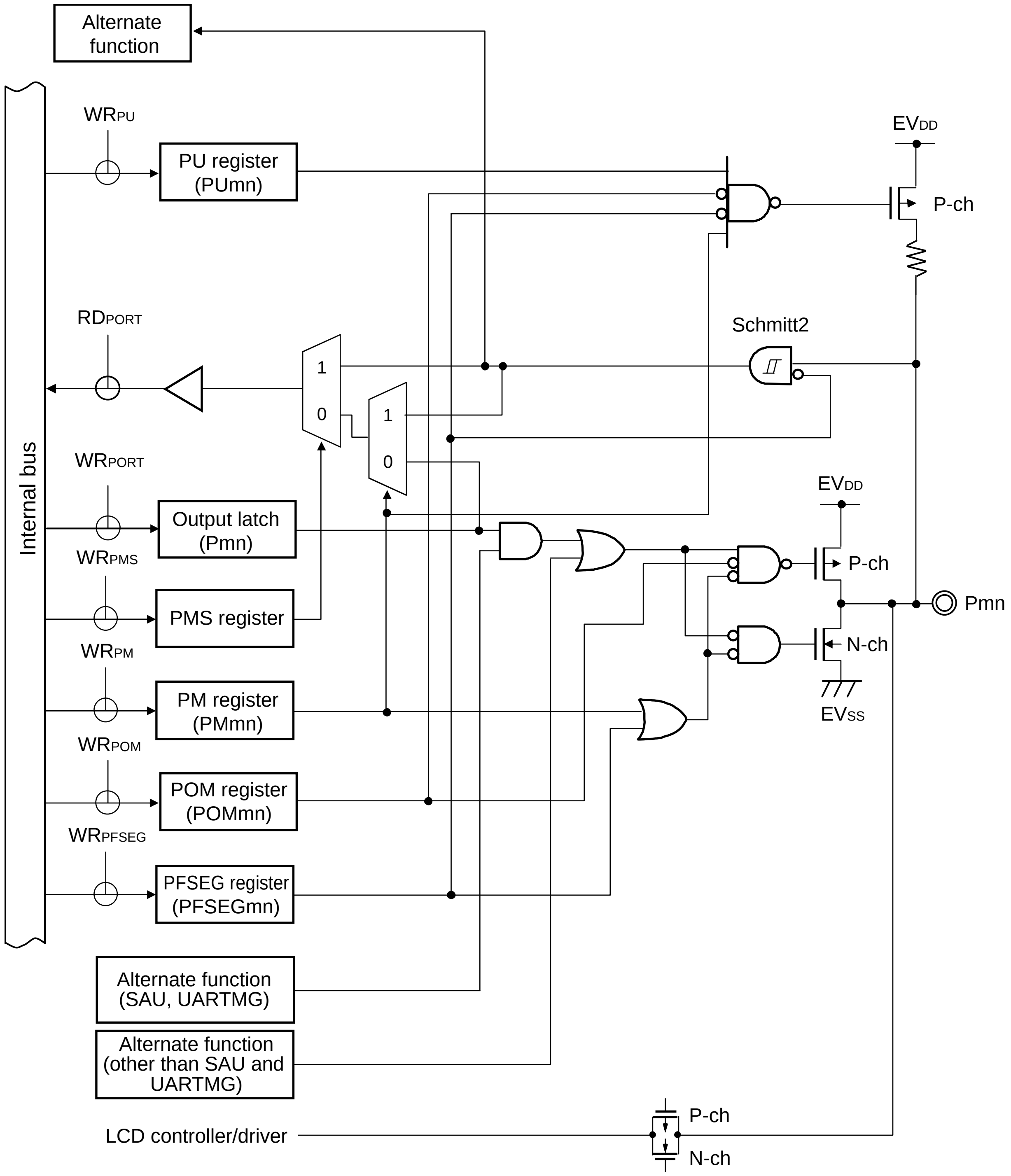
Figure 2-11. Pin Block Diagram for Pin Type 7-5-6



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-12. Pin Block Diagram for Pin Type 7-5-10



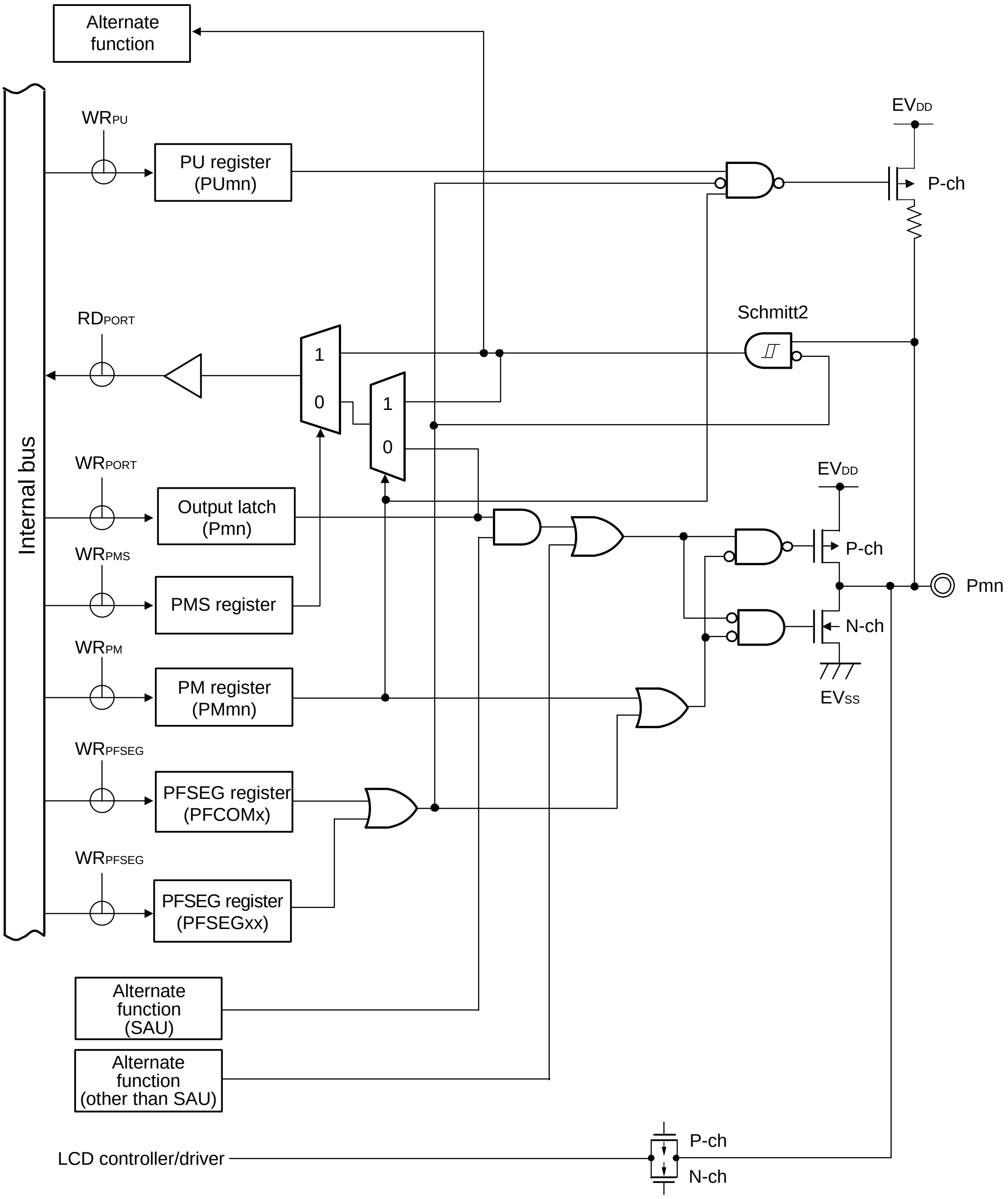
Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

3. UARTMG: Serial interface UARTMG

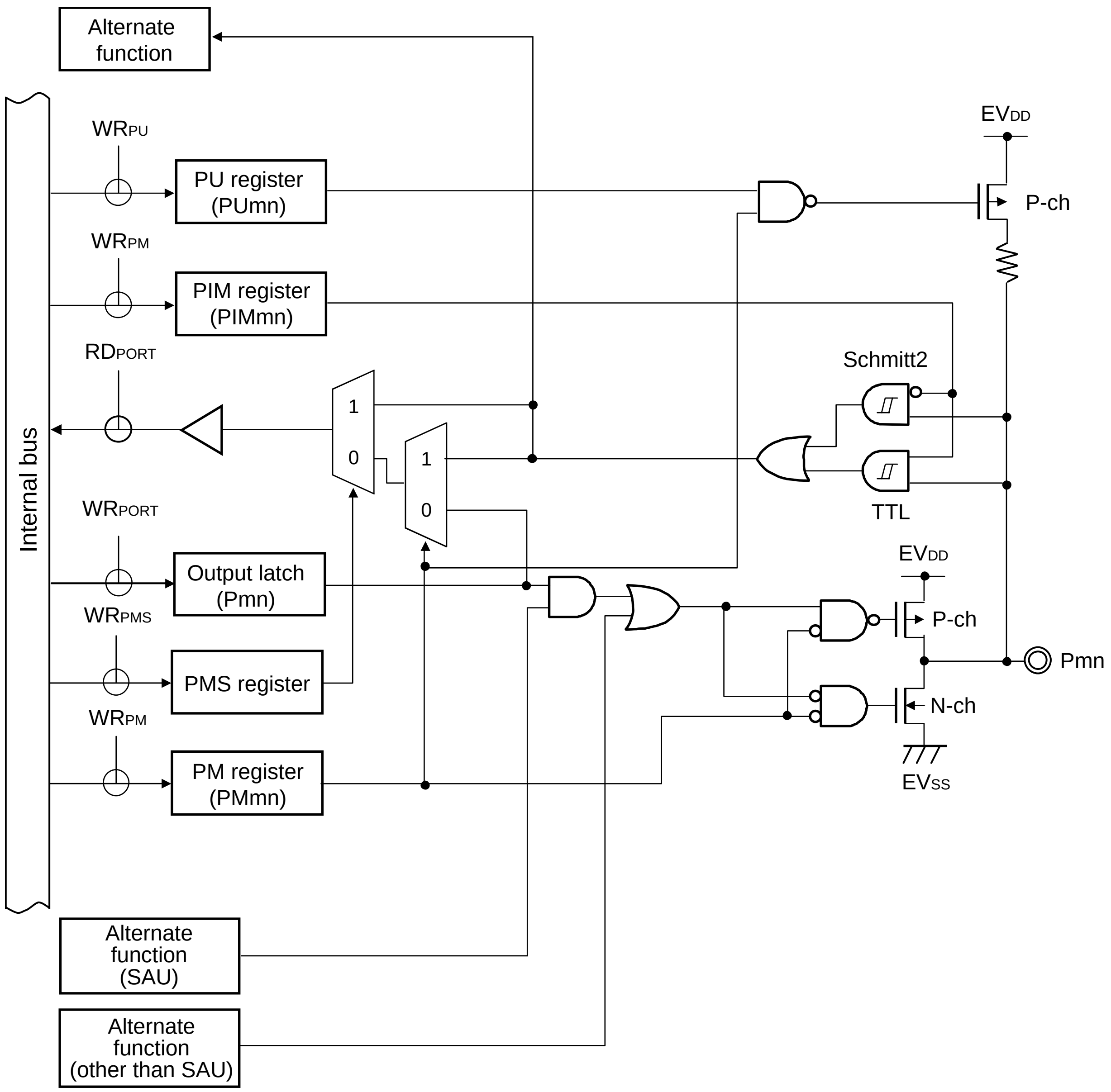
Figure 2-13. Pin Block Diagram for Pin Type 7-5-29



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

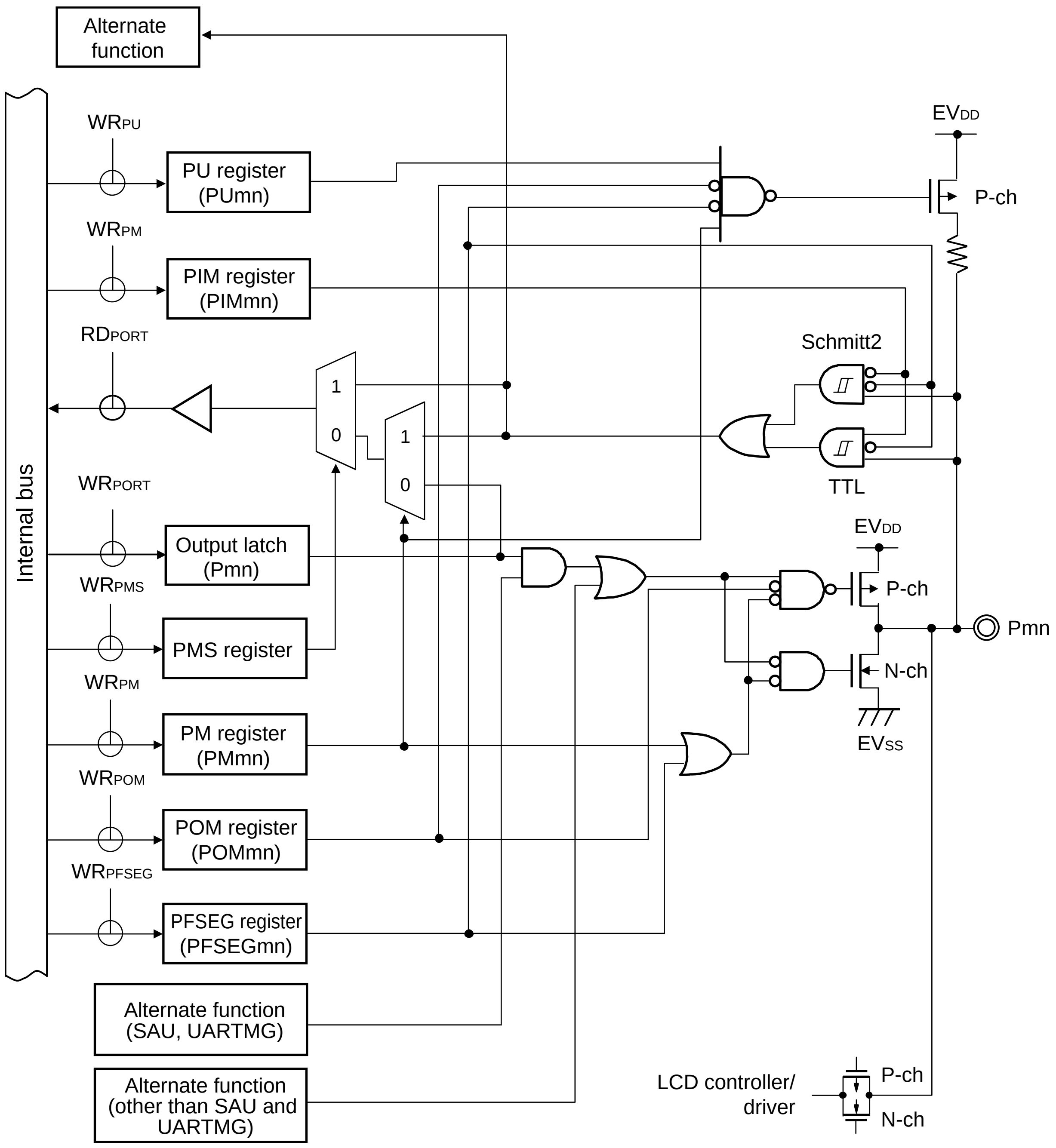
Figure 2-14. Pin Block Diagram for Pin Type 8-1-3



Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-15. Pin Block Diagram for Pin Type 8-5-10



Cautions 1 A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

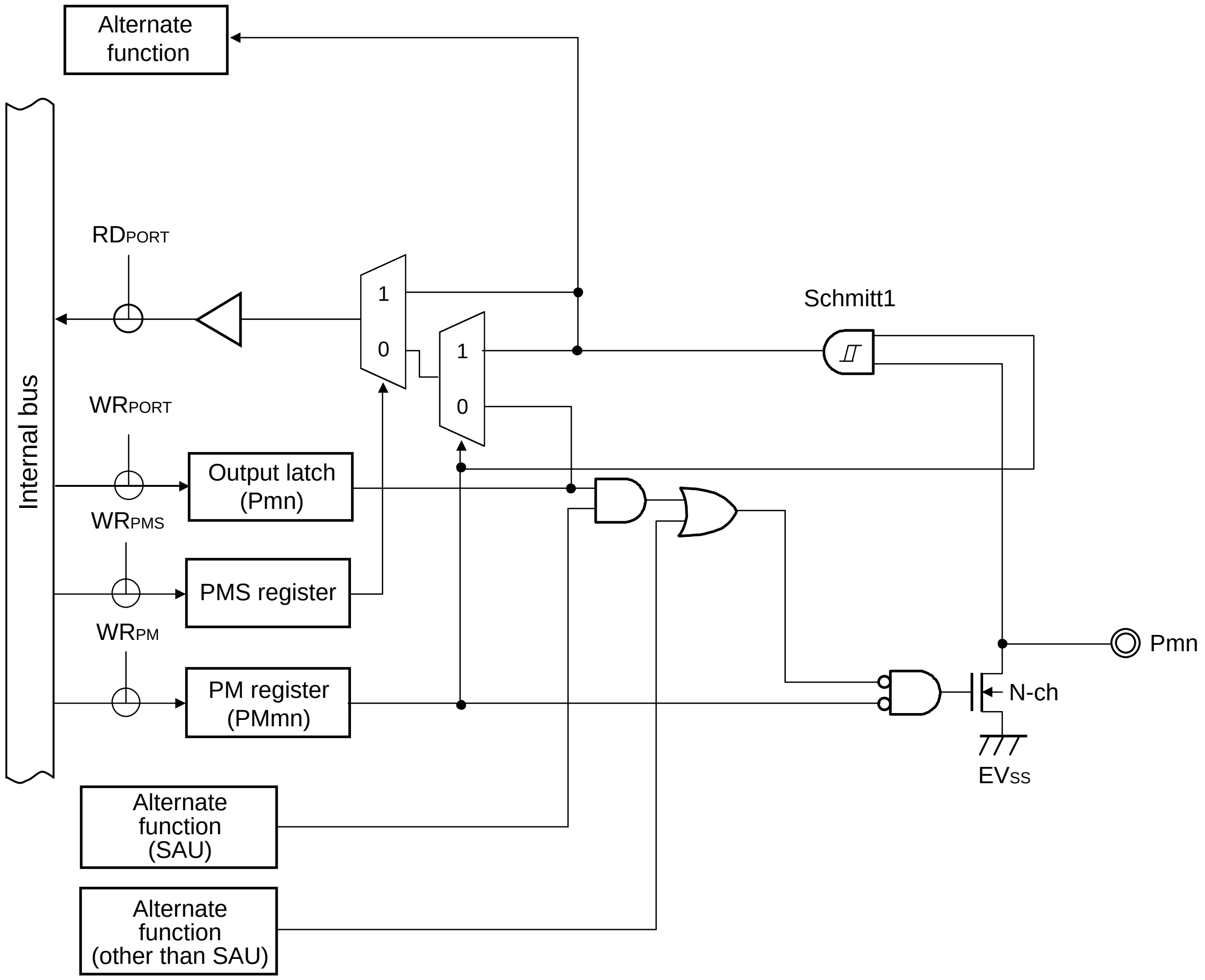
2 Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

3. UARTMG: Serial interface UARTMG

Figure 2-16. Pin Block Diagram for Pin Type 12-1-2

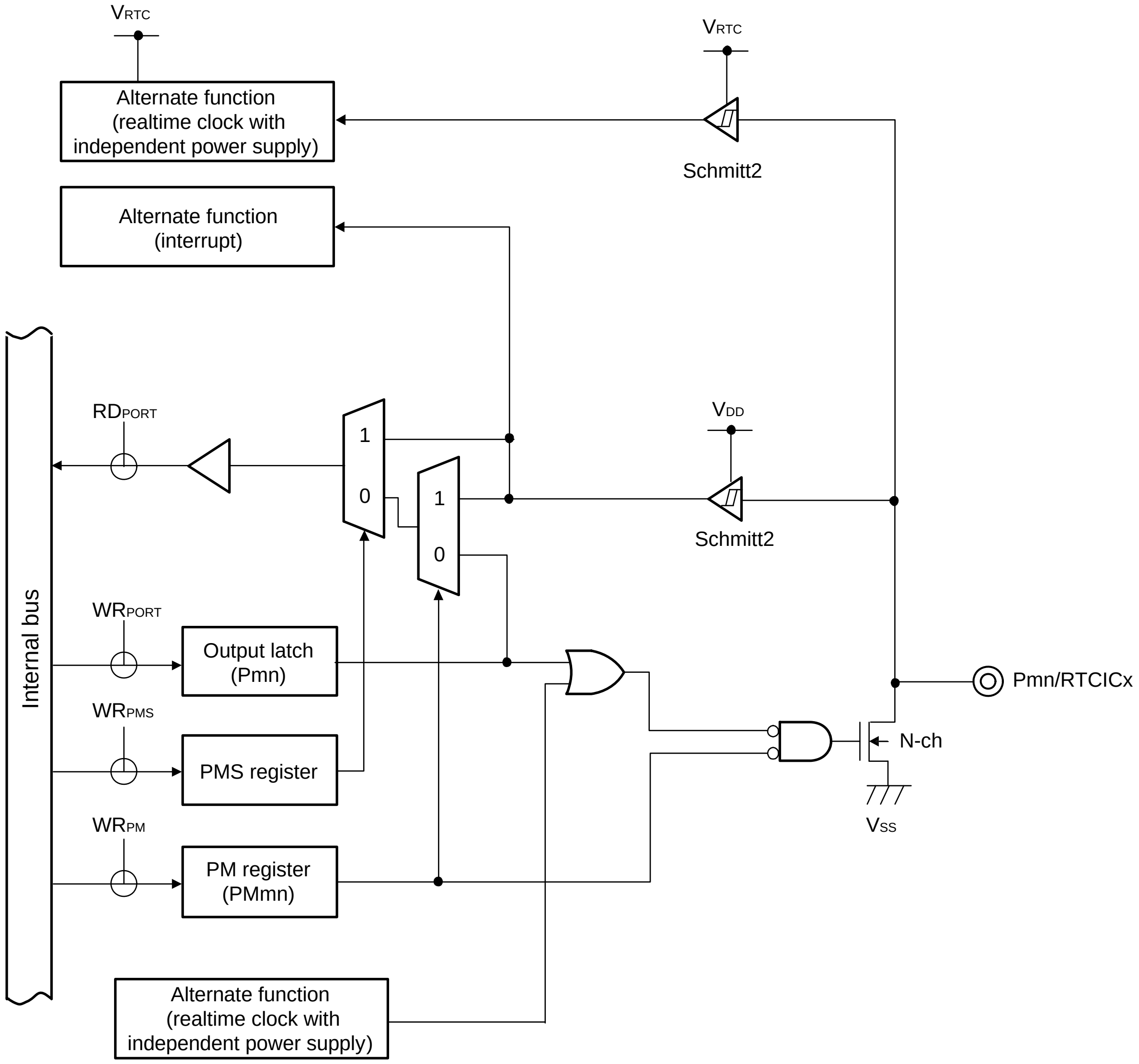


Caution A through current may flow through if the pin is in the intermediate potential, because the input buffer is turned on when the pin is in output mode.

Remarks 1. For alternate functions, see **2.1 Port Function**.

2. SAU: Serial array unit

Figure 2-17. Pin Block Diagram for Pin Type 12-1-6



<R> **Caution 1 When a high-level signal is to be input to any pin among pins P150 to P152 (including if the pin is in use for a multiplexed pin function rather than for GPIO), the pin should always be individually connected via a resistor to V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V.**

2 A through current may flow through if the pin is in the intermediate potential, because the Input buffer is turned on when the pin is in output mode.

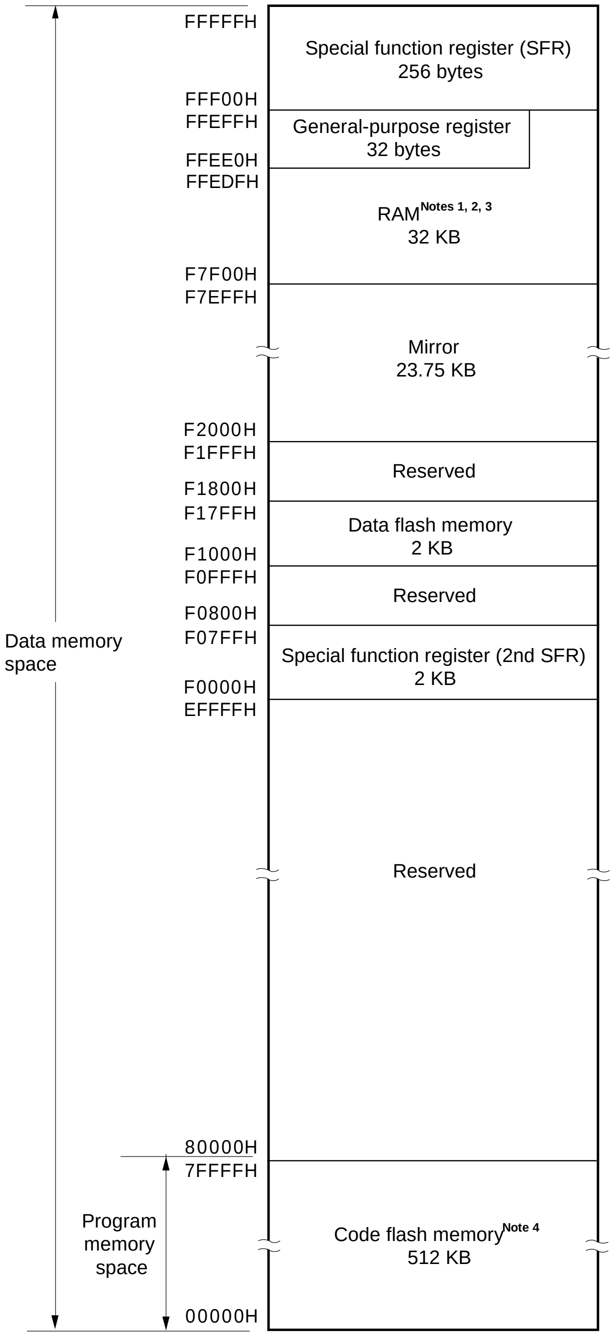
<R> **Remark** For alternate functions, see **2.1 Port Function**.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/I1C (512 KB) can access a 1 MB address space. **Figures 3-1** and **3-2** show the memory maps.

Figure 3-1. Memory Map (R5F10NPL, R5F10NML)



Notes 1. Do not allocate RAM addresses which are used as a stack area, a data buffer used for flash library, an argument of library function, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

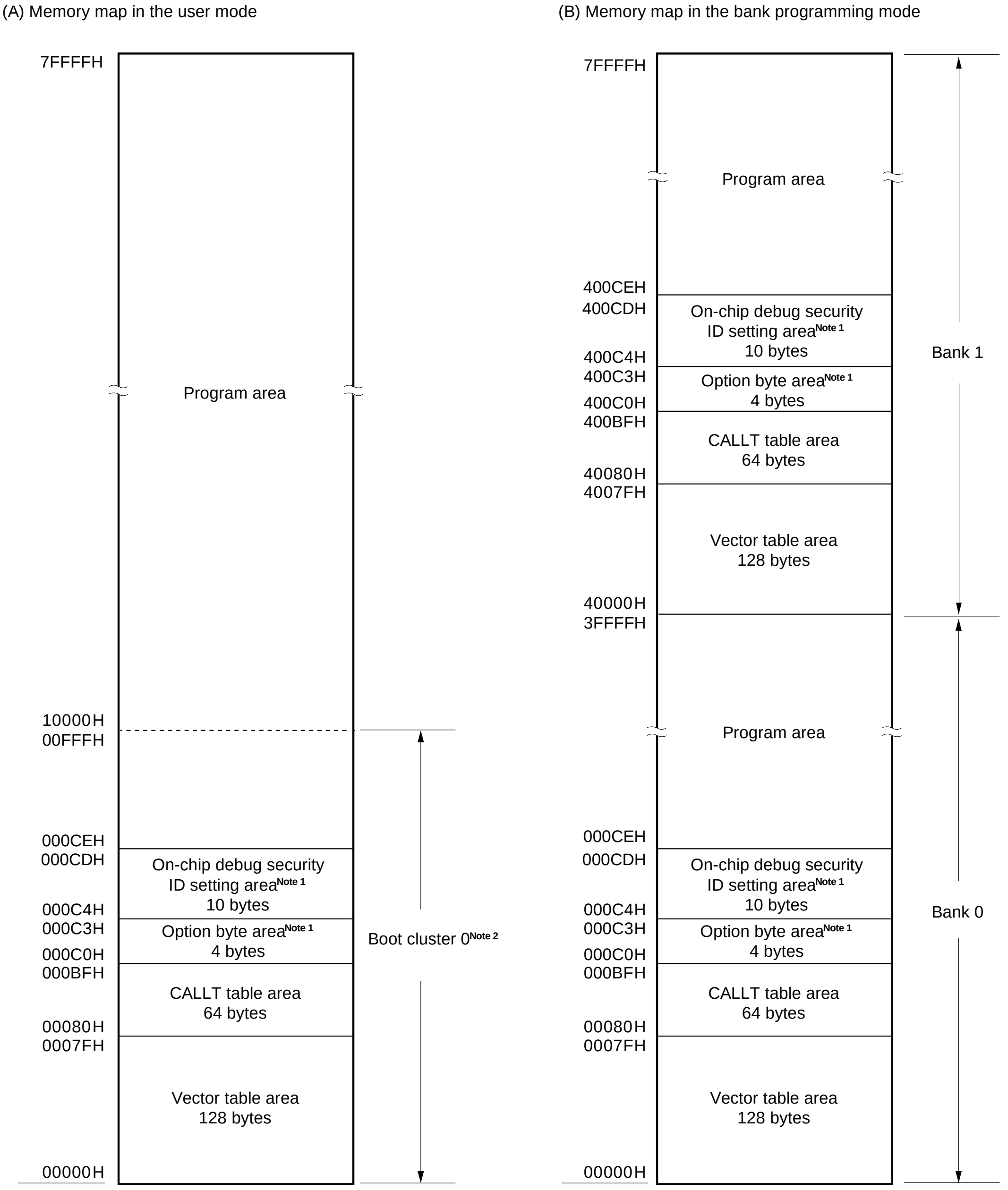
The RAM area used by the flash library starts at F7F00H. For RAM area that flash library uses, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

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

3. When using the trace function of on-chip debugging, area F8300H to F86FFH is disabled.

4. The on-chip ROM (code flash memory) is handled as single area in the user mode and as two areas in the bank programming mode. For the detailed memory maps, see **Figure 3-2 Memory Map of the Code Flash Memory**. For details on the user mode and bank programming mode, see **38.6.2.1 Switching bank modes**.

Figure 3-2. Memory Map of Code Flash Memory


(A) Memory map in the user mode
7FFFFH
Program area
10000H
00FFFH
000CEH
000CDH
On-chip debug security ID setting areaNote 1 10 bytes
000C4H
000C3H
Option byte areaNote 1 4 bytes
000C0H
000BFH
CALLT table area 64 bytes
00080H
0007FH
Vector table area 128 bytes
00000H
Boot cluster 0Note 2
(B) Memory map in the bank programming mode
7FFFFH
Program area
400CEH
400CDH
On-chip debug security ID setting areaNote 1 10 bytes
400C4H
400C3H
Option byte areaNote 1 4 bytes
400C0H
400BFH
CALLT table area 64 bytes
40080H
4007FH
Vector table area 128 bytes
40000H
3FFFFH
Program area
000CEH
000CDH
On-chip debug security ID setting areaNote 1 10 bytes
000C4H
000C3H
Option byte areaNote 1 4 bytes
000C0H
000BFH
CALLT table area 64 bytes
00080H
0007FH
Vector table area 128 bytes
00000H
Bank 1
Bank 0

Notes 1. When bank swapping is not to be used:

The option bytes are set in 000C0H to 000C3H, and the on-chip debugging security ID is set in 000C4H to 000CDH.

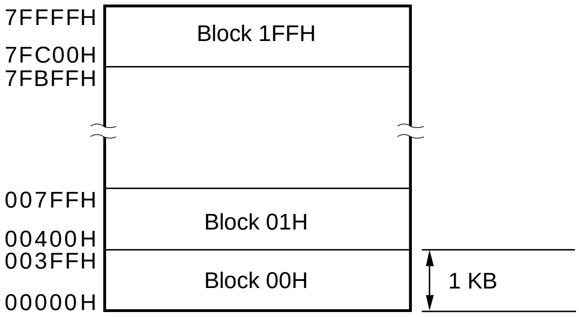
When bank swapping is to be used:

The option bytes are set in 000C0H to 000C3H and 400C0H to 400C3H, and the on-chip debugging security ID is set in 000C4H to 000CDH and 400C4H to 400CDH.

2. Writing to boot cluster 0 can be prohibited by the security settings (see **38.7 Security Settings**).

Caution When executing instructions from the RAM area 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. Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 34.3.3 RAM parity error detection function.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(R5F10NPL, R5F10NML)

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

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (1/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H	10000H to 103FFH	40H	18000H to 183FFH	60H
00400H to 007FFH	01H	08400H to 087FFH	21H	10400H to 107FFH	41H	18400H to 187FFH	61H
00800H to 00BFFH	02H	08800H to 08BFFH	22H	10800H to 10BFFH	42H	18800H to 18BFFH	62H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H	10C00H to 10FFFH	43H	18C00H to 18FFFH	63H
01000H to 013FFH	04H	09000H to 093FFH	24H	11000H to 113FFH	44H	19000H to 193FFH	64H
01400H to 017FFH	05H	09400H to 097FFH	25H	11400H to 117FFH	45H	19400H to 197FFH	65H
01800H to 01BFFH	06H	09800H to 09BFFH	26H	11800H to 11BFFH	46H	19800H to 19BFFH	66H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H	11C00H to 11FFFH	47H	19C00H to 19FFFH	67H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H	12000H to 123FFH	48H	1A000H to 1A3FFH	68H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H	12400H to 127FFH	49H	1A400H to 1A7FFH	69H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH	12800H to 12BFFH	4AH	1A800H to 1ABFFH	6AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH	12C00H to 12FFFH	4BH	1AC00H to 1AFFFH	6BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH	13000H to 133FFH	4CH	1B000H to 1B3FFH	6CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH	13400H to 137FFH	4DH	1B400H to 1B7FFH	6DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH	13800H to 13BFFH	4EH	1B800H to 1BBFFH	6EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH	13C00H to 13FFFH	4FH	1BC00H to 1BFFFH	6FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H	14000H to 143FFH	50H	1C000H to 1C3FFH	70H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H	14400H to 147FFH	51H	1C400H to 1C7FFH	71H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H	14800H to 14BFFH	52H	1C800H to 1CBFFH	72H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H	14C00H to 14FFFH	53H	1CC00H to 1CFFFH	73H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H	15000H to 153FFH	54H	1D000H to 1D3FFH	74H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H	15400H to 157FFH	55H	1D400H to 1D7FFH	75H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H	15800H to 15BFFH	56H	1D800H to 1DBFFH	76H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H	15C00H to 15FFFH	57H	1DC00H to 1DFFFH	77H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H	16000H to 163FFH	58H	1E000H to 1E3FFH	78H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H	16400H to 167FFH	59H	1E400H to 1E7FFH	79H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH	16800H to 16BFFH	5AH	1E800H to 1EBFFH	7AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH	16C00H to 16FFFH	5BH	1EC00H to 1EFFFH	7BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH	17000H to 173FFH	5CH	1F000H to 1F3FFH	7CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH	17400H to 177FFH	5DH	1F400H to 1F7FFH	7DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH	17800H to 17BFFH	5EH	1F800H to 1FBFFH	7EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH	17C00H to 17FFFH	5FH	1FC00H to 1FFFFH	7FH

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (2/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
20000H to 203FFH	80H	28000H to 283FFH	A0H	30000H to 303FFH	C0H	38000H to 383FFH	E0H
20400H to 207FFH	81H	28400H to 287FFH	A1H	30400H to 307FFH	C1H	38400H to 387FFH	E1H
20800H to 20BFFH	82H	28800H to 28BFFH	A2H	30800H to 30BFFH	C2H	38800H to 38BFFH	E2H
20C00H to 20FFFH	83H	28C00H to 28FFFH	A3H	30C00H to 30FFFH	C3H	38C00H to 38FFFH	E3H
21000H to 213FFH	84H	29000H to 293FFH	A4H	31000H to 313FFH	C4H	39000H to 393FFH	E4H
21400H to 217FFH	85H	29400H to 297FFH	A5H	31400H to 317FFH	C5H	39400H to 397FFH	E5H
21800H to 21BFFH	86H	29800H to 29BFFH	A6H	31800H to 31BFFH	C6H	39800H to 39BFFH	E6H
21C00H to 21FFFH	87H	29C00H to 29FFFH	A7H	31C00H to 31FFFH	C7H	39C00H to 39FFFH	E7H
22000H to 223FFH	88H	2A000H to 2A3FFH	A8H	32000H to 323FFH	C8H	3A000H to 3A3FFH	E8H
22400H to 227FFH	89H	2A400H to 2A7FFH	A9H	32400H to 327FFH	C9H	3A400H to 3A7FFH	E9H
22800H to 22BFFH	8AH	2A800H to 2ABFFH	AAH	32800H to 32BFFH	CAH	3A800H to 3ABFFH	EAH
22C00H to 22FFFH	8BH	2AC00H to 2AFFFH	ABH	32C00H to 32FFFH	CBH	3AC00H to 3AFFFH	EBH
23000H to 233FFH	8CH	2B000H to 2B3FFH	ACH	33000H to 333FFH	CCH	3B000H to 3B3FFH	ECH
23400H to 237FFH	8DH	2B400H to 2B7FFH	ADH	33400H to 337FFH	CDH	3B400H to 3B7FFH	EDH
23800H to 23BFFH	8EH	2B800H to 2BBFFH	AEH	33800H to 33BFFH	CEH	3B800H to 3BBFFH	EEH
23C00H to 23FFFH	8FH	2BC00H to 2BFFFH	AFH	33C00H to 33FFFH	CFH	3BC00H to 3BFFFH	EFH
24000H to 243FFH	90H	2C000H to 2C3FFH	B0H	34000H to 343FFH	D0H	3C000H to 3C3FFH	F0H
24400H to 247FFH	91H	2C400H to 2C7FFH	B1H	34400H to 347FFH	D1H	3C400H to 3C7FFH	F1H
24800H to 24BFFH	92H	2C800H to 2CBFFH	B2H	34800H to 34BFFH	D2H	3C800H to 3CBFFH	F2H
24C00H to 24FFFH	93H	2CC00H to 2CFFFH	B3H	34C00H to 34FFFH	D3H	3CC00H to 3CFFFH	F3H
25000H to 253FFH	94H	2D000H to 2D3FFH	B4H	35000H to 353FFH	D4H	3D000H to 3D3FFH	F4H
25400H to 257FFH	95H	2D400H to 2D7FFH	B5H	35400H to 357FFH	D5H	3D400H to 3D7FFH	F5H
25800H to 25BFFH	96H	2D800H to 2DBFFH	B6H	35800H to 35BFFH	D6H	3D800H to 3DBFFH	F6H
25C00H to 25FFFH	97H	2DC00H to 2DFFFH	B7H	35C00H to 35FFFH	D7H	3DC00H to 3DFFFH	F7H
26000H to 263FFH	98H	2E000H to 2E3FFH	B8H	36000H to 363FFH	D8H	3E000H to 3E3FFH	F8H
26400H to 267FFH	99H	2E400H to 2E7FFH	B9H	36400H to 367FFH	D9H	3E400H to 3E7FFH	F9H
26800H to 26BFFH	9AH	2E800H to 2EBFFH	BAH	36800H to 36BFFH	DAH	3E800H to 3EBFFH	FAH
26C00H to 26FFFH	9BH	2EC00H to 2EFFFH	BBH	36C00H to 36FFFH	DBH	3EC00H to 3EFFFH	FBH
27000H to 273FFH	9CH	2F000H to 2F3FFH	BCH	37000H to 373FFH	DCH	3F000H to 3F3FFH	FCH
27400H to 277FFH	9DH	2F400H to 2F7FFH	BDH	37400H to 377FFH	DDH	3F400H to 3F7FFH	FDH
27800H to 27BFFH	9EH	2F800H to 2FBFFH	BEH	37800H to 37BFFH	DEH	3F800H to 3FBFFH	FEH
27C00H to 27FFFH	9FH	2FC00H to 2FFFFH	BFH	37C00H to 37FFFH	DFH	3FC00H to 3FFFFH	FFH

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (3/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
40000H to 403FFH	100H	48000H to 483FFH	120H	50000H to 503FFH	140H	58000H to 583FFH	160H
40400H to 407FFH	101H	48400H to 487FFH	121H	50400H to 507FFH	141H	58400H to 587FFH	161H
40800H to 40BFFH	102H	48800H to 48BFFH	122H	50800H to 50BFFH	142H	58800H to 58BFFH	162H
40C00H to 40FFFH	103H	48C00H to 48FFFH	123H	50C00H to 50FFFH	143H	58C00H to 58FFFH	163H
41000H to 413FFH	104H	49000H to 493FFH	124H	51000H to 513FFH	144H	59000H to 593FFH	164H
41400H to 417FFH	105H	49400H to 497FFH	125H	51400H to 517FFH	145H	59400H to 597FFH	165H
41800H to 41BFFH	106H	49800H to 49BFFH	126H	51800H to 51BFFH	146H	59800H to 59BFFH	166H
41C00H to 41FFFH	107H	49C00H to 49FFFH	127H	51C00H to 51FFFH	147H	59C00H to 59FFFH	167H
42000H to 423FFH	108H	4A000H to 4A3FFH	128H	52000H to 523FFH	148H	5A000H to 5A3FFH	168H
42400H to 427FFH	109H	4A400H to 4A7FFH	129H	52400H to 527FFH	149H	5A400H to 5A7FFH	169H
42800H to 42BFFH	10AH	4A800H to 4ABFFH	12AH	52800H to 52BFFH	14AH	5A800H to 5ABFFH	16AH
42C00H to 42FFFH	10BH	4AC00H to 4AFFFH	12BH	52C00H to 52FFFH	14BH	5AC00H to 5AFFFH	16BH
43000H to 433FFH	10CH	4B000H to 4B3FFH	12CH	53000H to 533FFH	14CH	5B000H to 5B3FFH	16CH
43400H to 437FFH	10DH	4B400H to 4B7FFH	12DH	53400H to 537FFH	14DH	5B400H to 5B7FFH	16DH
43800H to 43BFFH	10EH	4B800H to 4BBFFH	12EH	53800H to 53BFFH	14EH	5B800H to 5BBFFH	16EH
43C00H to 43FFFH	10FH	4BC00H to 4BFFFH	12FH	53C00H to 53FFFH	14FH	5BC00H to 5BFFFH	16FH
44000H to 443FFH	110H	4C000H to 4C3FFH	130H	54000H to 543FFH	150H	5C000H to 5C3FFH	170H
44400H to 447FFH	111H	4C400H to 4C7FFH	131H	54400H to 547FFH	151H	5C400H to 5C7FFH	171H
44800H to 44BFFH	112H	4C800H to 4CBFFH	132H	54800H to 54BFFH	152H	5C800H to 5CBFFH	172H
44C00H to 44FFFH	113H	4CC00H to 4CFFFH	133H	54C00H to 54FFFH	153H	5CC00H to 5CFFFH	173H
45000H to 453FFH	114H	4D000H to 4D3FFH	134H	55000H to 553FFH	154H	5D000H to 5D3FFH	174H
45400H to 457FFH	115H	4D400H to 4D7FFH	135H	55400H to 557FFH	155H	5D400H to 5D7FFH	175H
45800H to 45BFFH	116H	4D800H to 4DBFFH	136H	55800H to 55BFFH	156H	5D800H to 5DBFFH	176H
45C00H to 45FFFH	117H	4DC00H to 4DFFFH	137H	55C00H to 55FFFH	157H	5DC00H to 5DFFFH	177H
46000H to 463FFH	118H	4E000H to 4E3FFH	138H	56000H to 563FFH	158H	5E000H to 5E3FFH	178H
46400H to 467FFH	119H	4E400H to 4E7FFH	139H	56400H to 567FFH	159H	5E400H to 5E7FFH	179H
46800H to 46BFFH	11AH	4E800H to 4EBFFH	13AH	56800H to 56BFFH	15AH	5E800H to 5EBFFH	17AH
46C00H to 46FFFH	11BH	4EC00H to 4EFFFH	13BH	56C00H to 56FFFH	15BH	5EC00H to 5EFFFH	17BH
47000H to 473FFH	11CH	4F000H to 4F3FFH	13CH	57000H to 573FFH	15CH	5F000H to 5F3FFH	17CH
47400H to 477FFH	11DH	4F400H to 4F7FFH	13DH	57400H to 577FFH	15DH	5F400H to 5F7FFH	17DH
47800H to 47BFFH	11EH	4F800H to 4FBFFH	13EH	57800H to 57BFFH	15EH	5F800H to 5FBFFH	17EH
47C00H to 47FFFH	11FH	4FC00H to 4FFFFH	13FH	57C00H to 57FFFH	15FH	5FC00H to 5FFFFH	17FH

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (4/4)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
60000H to 603FFH	180H	68000H to 683FFH	1A0H	70000H to 703FFH	1C0H	78000H to 783FFH	1E0H
60400H to 607FFH	181H	68400H to 687FFH	1A1H	70400H to 707FFH	1C1H	78400H to 787FFH	1E1H
60800H to 60BFFH	182H	68800H to 68BFFH	1A2H	70800H to 70BFFH	1C2H	78800H to 78BFFH	1E2H
60C00H to 60FFFH	183H	68C00H to 68FFFH	1A3H	70C00H to 70FFFH	1C3H	78C00H to 78FFFH	1E3H
61000H to 613FFH	184H	69000H to 693FFH	1A4H	71000H to 713FFH	1C4H	79000H to 793FFH	1E4H
61400H to 617FFH	185H	69400H to 697FFH	1A5H	71400H to 717FFH	1C5H	79400H to 797FFH	1E5H
61800H to 61BFFH	186H	69800H to 69BFFH	1A6H	71800H to 71BFFH	1C6H	79800H to 79BFFH	1E6H
61C00H to 61FFFH	187H	69C00H to 69FFFH	1A7H	71C00H to 71FFFH	1C7H	79C00H to 79FFFH	1E7H
62000H to 623FFH	188H	6A000H to 6A3FFH	1A8H	72000H to 723FFH	1C8H	7A000H to 7A3FFH	1E8H
62400H to 627FFH	189H	6A400H to 6A7FFH	1A9H	72400H to 727FFH	1C9H	7A400H to 7A7FFH	1E9H
62800H to 62BFFH	18AH	6A800H to 6ABFFH	1AAH	72800H to 72BFFH	1CAH	7A800H to 7ABFFH	1EAH
62C00H to 62FFFH	18BH	6AC00H to 6AFFFH	1ABH	72C00H to 72FFFH	1CBH	7AC00H to 7AFFFH	1EBH
63000H to 633FFH	18CH	6B000H to 6B3FFH	1ACH	73000H to 733FFH	1CCH	7B000H to 7B3FFH	1ECH
63400H to 637FFH	18DH	6B400H to 6B7FFH	1ADH	73400H to 737FFH	1CDH	7B400H to 7B7FFH	1EDH
63800H to 63BFFH	18EH	6B800H to 6BBFFH	1AEH	73800H to 73BFFH	1CEH	7B800H to 7BBFFH	1EEH
63C00H to 63FFFH	18FH	6BC00H to 6BFFFH	1AFH	73C00H to 73FFFH	1CFH	7BC00H to 7BFFFH	1EFH
64000H to 643FFH	190H	6C000H to 6C3FFH	1B0H	74000H to 743FFH	1D0H	7C000H to 7C3FFH	1F0H
64400H to 647FFH	191H	6C400H to 6C7FFH	1B1H	74400H to 747FFH	1D1H	7C400H to 7C7FFH	1F1H
64800H to 64BFFH	192H	6C800H to 6CBFFH	1B2H	74800H to 74BFFH	1D2H	7C800H to 7CBFFH	1F2H
64C00H to 64FFFH	193H	6CC00H to 6CFFFH	1B3H	74C00H to 74FFFH	1D3H	7CC00H to 7CFFFH	1F3H
65000H to 653FFH	194H	6D000H to 6D3FFH	1B4H	75000H to 753FFH	1D4H	7D000H to 7D3FFH	1F4H
65400H to 657FFH	195H	6D400H to 6D7FFH	1B5H	75400H to 757FFH	1D5H	7D400H to 7D7FFH	1F5H
65800H to 65BFFH	196H	6D800H to 6DBFFH	1B6H	75800H to 75BFFH	1D6H	7D800H to 7DBFFH	1F6H
65C00H to 65FFFH	197H	6DC00H to 6DFFFH	1B7H	75C00H to 75FFFH	1D7H	7DC00H to 7DFFFH	1F7H
66000H to 663FFH	198H	6E000H to 6E3FFH	1B8H	76000H to 763FFH	1D8H	7E000H to 7E3FFH	1F8H
66400H to 667FFH	199H	6E400H to 6E7FFH	1B9H	76400H to 767FFH	1D9H	7E400H to 7E7FFH	1F9H
66800H to 66BFFH	19AH	6E800H to 6EBFFH	1BAH	76800H to 76BFFH	1DAH	7E800H to 7EBFFH	1FAH
66C00H to 66FFFH	19BH	6EC00H to 6EFFFH	1BBH	76C00H to 76FFFH	1DBH	7EC00H to 7EFFFH	1FBH
67000H to 673FFH	19CH	6F000H to 6F3FFH	1BCH	77000H to 773FFH	1DCH	7F000H to 7F3FFH	1FCH
67400H to 677FFH	19DH	6F400H to 6F7FFH	1BDH	77400H to 777FFH	1DDH	7F400H to 7F7FFH	1FDH
67800H to 67BFFH	19EH	6F800H to 6FBFFH	1BEH	77800H to 77BFFH	1DEH	7F800H to 7FBFFH	1FEH
67C00H to 67FFFH	19FH	6FC00H to 6FFFFH	1BFH	77C00H to 77FFFH	1DFH	7FC00H to 7FFFFH	1FFH

Remark R5F10NPL, R5F10NML: Block numbers 00H to 1FFH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The RL78/I1C (512 KB) products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10NPL, R5F10NML	Flash memory	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.

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

When bank swapping is to be used, set a vector table also at 40000H to 4007FH.

Table 3-3. Vector Table (1/3)

Vector Table Address	Interrupt Source
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE
0004H	INTWDTI
0006H	INTLVI
0008H	INTP0
000AH	INTP1
000CH	INTP2
000EH	INTP3
0010H	INTP4
0012H	INTP5
0014H	INTST2
	INTSTMG0
0016H	INTSR2
	INTSRMG0
0018H	INTSRE2
	INTSREMG0
001AH	INTCR
	INTSMP00
	INTSMP10
001CH	INTAES
	INTAESF
001EH	INTST0/INTCSI00/INTIIC00
0020H	INTIICA0
0022H	INTSR0
0024H	INTSRE0
	INTTM01H
0026H	INTST1/INTCSI10/INTIIC10
0028H	INTSR1
002AH	INTSRE1
	INTTM03H
002CH	INTTM00
002EH	INTRTCALM0
	INTSMP01
	INTSMP11
0030H	INTFM
	INTSMP02
	INTSMP12
0032H	INTTM01
0034H	INTTM02
0036H	INTTM03

Table 3-3. Vector Table (2/3)

Vector Table Address	Interrupt Source
0038H	INTAD
003AH	INTRTCRPD
	INTSMP03
	INTSMP13
003CH	INTIT
003EH	INTKR
	INTRTCALM1
	INTSMP04
	INTSMP14
0040H	INTST3/INTCSI30/INTIIC30
0042H	INTSR3
0044H	INTDSAD
	INTSMP05
	INTSMP15
0046H	INTTM04
0048H	INTTM05
	INTSMOTA0
004AH	INTP6
004CH	INTP7
004EH	INTRTCIC2
	INTP12
0050H	INTRTCIC1
	INTP13
0052H	INTRTCIC0
	INTP14
0054H	INTTM06
	INTSMOTB0
0056H	INTTM07
	INTSMOTA1
0058H	INTIT00
005AH	INTIT01
005CH	INTSRE3
005EH	INTMACLOF
	INTSMOTB1
0060H	INTOSDC
0062H	INTFL
0064H	INTP8
0066H	INTP9
0068H	INTIT10
006AH	INTIT11

Table 3-3. Vector Table (3/3)

Vector Table Address	Interrupt Source
006CH	INTLVDVDD
	INTIT20
006EH	INTLVDVBAT
	INTIT21
0070H	INTLVDVRTC
	INTIT30
0072H	INTLVDEXLVD
	INTIT31
0074H	INTST4
	INTSTMG1
0076H	INTSR4
	INTSRMG1
0078H	INTSRE4
	INTSREMG1
007AH	INTTRJ0
	INTDSADDEC
007CH	INTTRJ1

(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 of 2 bytes). When bank swapping is to be used, set a CALLT instruction table also at 40080H to 400BFH.

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 400C0H to 400C3H when bank swapping is to be used. For details, see **CHAPTER 37 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 400C4H to 400CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when bank swapping is not to be used and at 000C4H to 000CDH and 400C4H to 400CDH when bank swapping is to be used. For details, see **CHAPTER 39 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/I1C (512 KB) mirrors the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

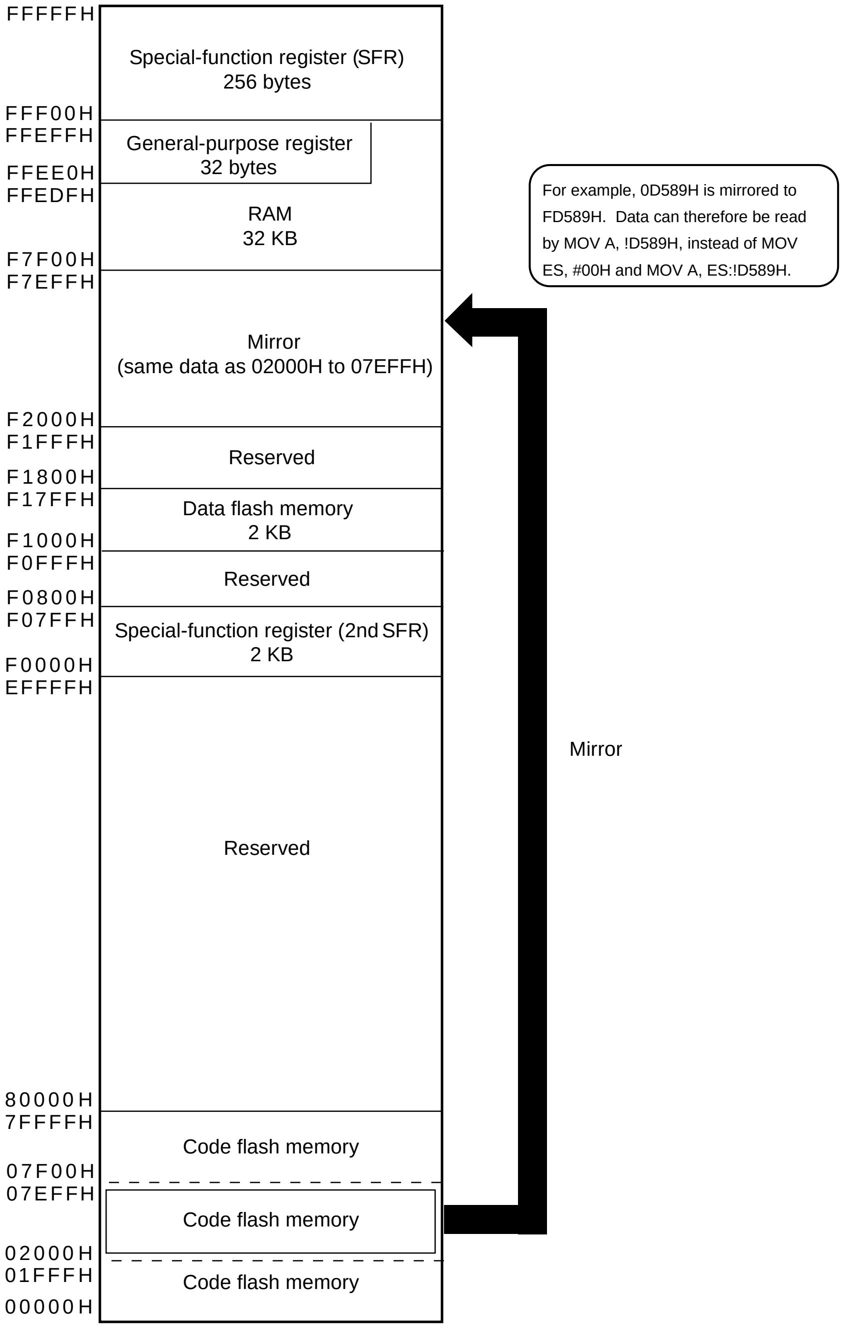
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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 R5F10NPL, R5F10NML (Flash memory: 512 KB, RAM: 32 KB)



For example, 0D589H is mirrored to FD589H. Data can therefore be read by MOV A, !D589H, instead of MOV ES, #00H and MOV A, ES:!D589H.

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.
  Reset signal generation sets this register to 00H.

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

Address: FFFFEH After reset: 00H 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 After setting the PMC register, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

The RL78/I1C (512 KB) products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F10NPL, R5F10NML	32768 × 8 bit (F7F00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are executed. (Instructions cannot be executed in the area to which general-purpose registers are allocated.) 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.

Cautions 1. The space (FFEE0H to FFEFFH) that the general-purpose registers are allocated cannot be used for fetching instructions or as a stack area.

2. Do not allocate RAM addresses which are used as a stack area, a data buffer used for flash library, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

3. Use of the following RAM area is prohibited when performing self-programming or rewriting of the data flash memory, because these areas are used for each library.

F7F00H to F8309H

4. The following RAM area cannot be used as a stack memory when using the trace function of on-chip debugging.

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** in **3.2.4 Special function registers (SFRs)**).

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

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

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

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

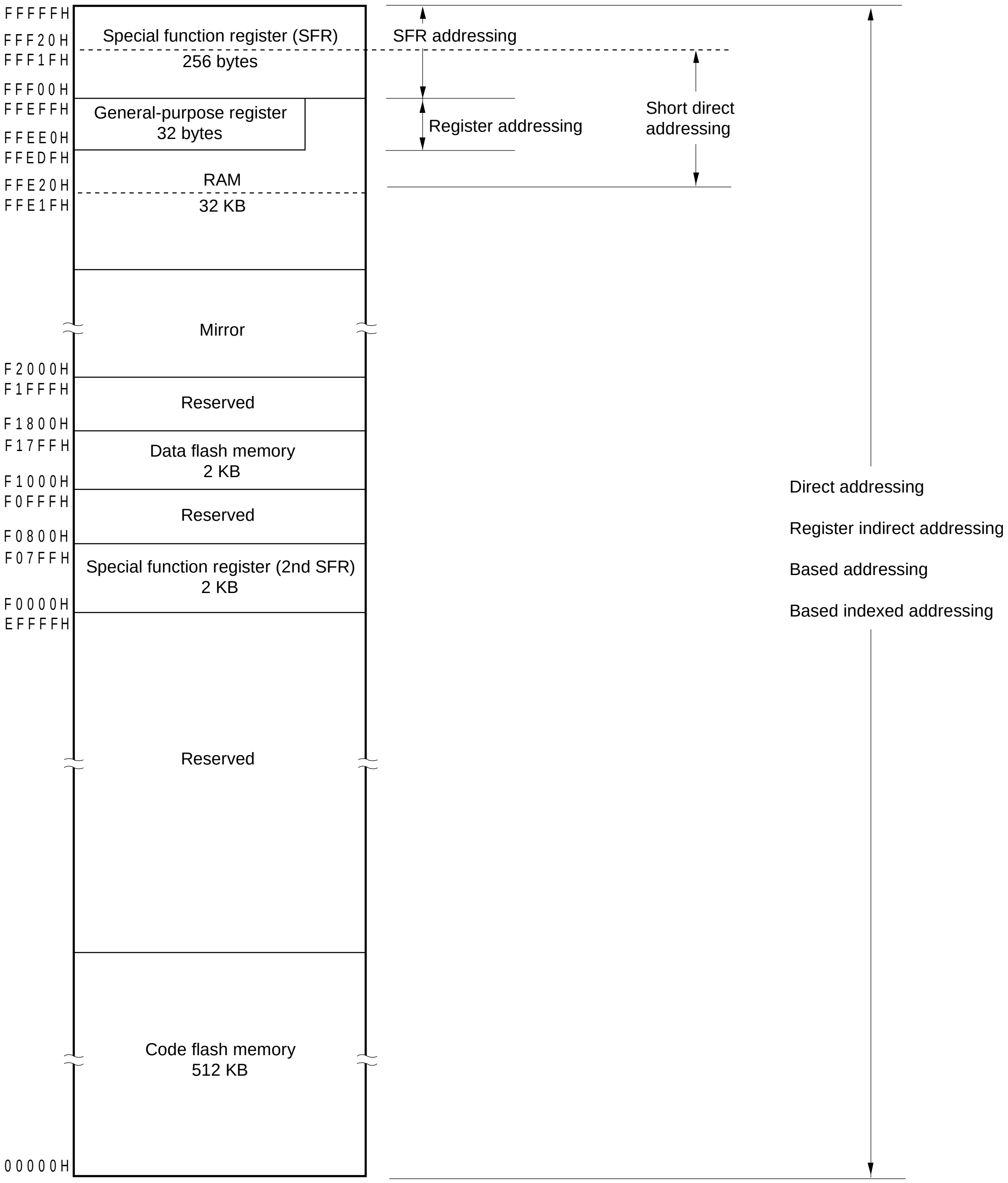
Caution Do not access addresses to which extended SFRs are not assigned.

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/I1C (512 KB), 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 (SFR) and general-purpose registers are available for use. **Figure 3-4** shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-4. Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/I1C (512 KB) products incorporate the following processor registers.

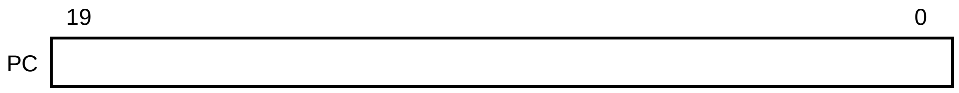
3.2.1 Control registers

The control registers control the program sequence, statuses 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.
Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-5. 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. Reset signal generation sets the PSW register to 06H.

Figure 3-6. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(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 maskable interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.
The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

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

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

These are 2-bit flags to select one of the four register banks.
In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

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

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

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) (see **27.3.3**) can not be acknowledged. Actual vectored interrupt 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 3-7. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	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 (restored) from the stack memory.

Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or a stack area.

3. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

4. Use of the following RAM area is prohibited when performing self-programming, or rewriting of the data flash memory, because these areas are used for each library.

F7F00H to F8309H

5. The following RAM area cannot be used as a stack memory when using the trace function of on-chip debugging.

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 consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

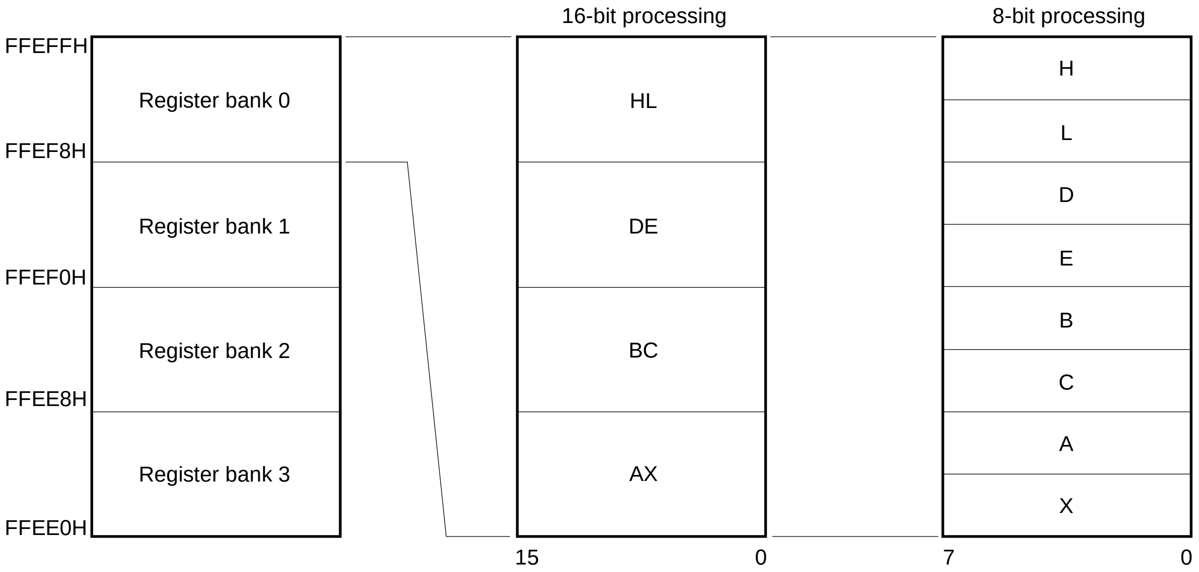
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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 interrupts for each bank.

Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-8. 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 direct addressing), respectively.

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

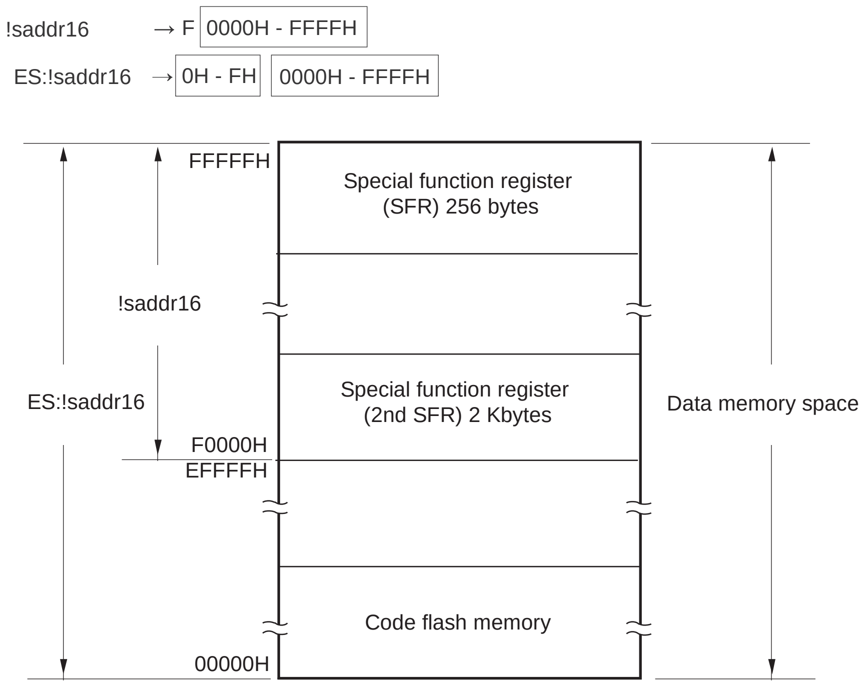
Figure 3-9. Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

The data area that can be accessed by using 16-bit addresses is the 64 KB from F0000H to FFFFFH. By using the ES register, this area can be extended to the 1 MB from 00000H to FFFFFH.

Figure 3-10. Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

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

SFRs are allocated to the FFF00H to FFFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating 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
  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
  Indicates each register status upon reset signal generation.

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

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

Table 3-5. SFR List (1/4)

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
FFF0FH	Port register 15	P15		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF14H	Serial data register 12	TXD3/ SIO30	SDR12	R/W[Note]	–	√	√	0000H
FFF15H		–			–	–		
FFF16H	Serial data register 13	RXD3	SDR13	R/W[Note]	–	√	√	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
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
FFF2FH	Port mode register 15	PM15		R/W	√	√	–	FFH
FFF34H	Key return control register	KRCTL		R/W	√	√	–	00H
FFF35H	Key return flag register	KRF		R/W	–	√	–	00H

Note These registers are read-only in the 80-pin product.

Table 3-5. SFR List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF37H	Key return mode register 0	KRM0		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF3AH	External interrupt rising edge enable register 1[Note]	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1[Note]	EGN1		R/W	√	√	–	00H
FFF3CH	Multiplication data register B(L)	MULBL		R/W	–	–	√	0000H
FFF3DH								
FFF3EH	Multiplication data register B(H)	MULBH		R/W	–	–	√	0000H
FFF3FH								
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	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	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
FFF58H	Serial data register 20[Note]	TXD4	SDR20	R/W	–	√	√	0000H
FFF59H		–			–	–		
FFF5AH	Serial data register 21[Note]	RXD4	SDR21	R/W	–	√	√	0000H
FFF5BH		–			–	–		
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								

Note These registers are incorporated in the 100-pin product, but are not incorporated in the 80-pin product.

Table 3-5. SFR List (3/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF6CH FFF6DH	Timer data register 06	TDR06	R/W	–	–	√	0000H
FFF6EH FFF6FH	Timer data register 07	TDR07	R/W	–	–	√	0000H
FFF90H FFF91H	12-bit interval timer control register	ITMC	R/W	–	–	√	0FFFH
FFFA0H	Clock operation mode control register	CMC	R/W	–	√	–	00H[Note 1]
FFFA1H	Clock operation status control register	CSC	R/W	√	√	–	C0H[Note 1]
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 resister	CKSEL	R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF	R	–	√	–	Undefined[Note 2]
FFFA9H	Voltage detection register	LVIM	R/W	√	√	–	00H[Note 2]
FFFAAH	Voltage detection level register	LVIS	R/W	√	√	–	00H/01H/ 81H[Note 2]
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/9AH[Note 3]
FFFACH	CRC input register	CRCIN	R/W	–	√	–	00H

Notes **1.** These registers are reset only by a power-on reset.

2. The reset values of the registers vary depending on the reset source as shown below.

<table>
<tr><th colspan="2">Reset Source
Register</th><th>$\overline{\text{RESET}}$ Input</th><th>Reset by POR</th><th>Reset by Execution of Illegal Instruction</th><th>Reset by WDT</th><th>Reset by RAM Parity Error</th><th>Reset by Illegal-Memory Access</th><th>Reset by LVD</th></tr>
<tr><td rowspan="5">RESF</td><td>TRAP</td><td colspan="2" rowspan="5">Cleared (0)</td><td>Set (1)</td><td colspan="3">Held</td><td rowspan="4">Held</td></tr>
<tr><td>WDTRF</td><td>Held</td><td>Set (1)</td><td colspan="2">Held</td></tr>
<tr><td>RPERF</td><td colspan="2">Held</td><td>Set (1)</td><td>Held</td></tr>
<tr><td>IAWRF</td><td colspan="3">Held</td><td>Set (1)</td></tr>
<tr><td>LVIRF</td><td colspan="4">Held</td><td>Set (1)</td></tr>
<tr><td rowspan="3">LVIM</td><td>LVISEN</td><td colspan="6">Cleared (0)</td><td rowspan="3">Held</td></tr>
<tr><td>LVIOMSK</td><td colspan="6" rowspan="2">Held</td></tr>
<tr><td>LVIF</td></tr>
<tr><td colspan="2">LVIS</td><td colspan="6">Cleared (00H/01H/81H)</td><td>Clear (00H/81H)[Note 4]</td></tr>
</table>

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

4. When option byte LVIMDS1, LVIMDS0 = 0, 1: LVD reset is not generated.

Table 3-5. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD2H	Interrupt request flag register 3L	IF3L	IF3	R/W	√	√	√	00H
FFFD3H	Interrupt request flag register 3H	IF3H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD6H	Interrupt mask flag register 3L	MK3L	MK3	R/W	√	√	√	FFH
FFFD7H	Interrupt mask flag register 3H	MK3H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDAH	Priority specification flag register 03L	PR03L	PR03	R/W	√	√	√	FFH
FFFDBH	Priority specification flag register 03H	PR03H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFDEH	Priority specification flag register 13L	PR13L	PR13	R/W	√	√	√	FFH
FFFDFH	Priority specification flag register 13H	PR13H		R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiply and accumulation register (L)	MACRL		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	–	–	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

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

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

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (!addr16.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>.<Bit number> or <Address>.<Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

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

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

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

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3-6. Extended SFR (2nd SFR) List (1/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
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
F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0044H	Port input mode register 4	PIM4	R/W	√	√	–	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	–	00H
F0048H	Port input mode register 8	PIM8	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	–	00H
F0058H	Port output mode register 8	POM8	R/W	√	√	–	00H
F0062H	Port mode control register 2	PMC2	R/W	√	√	–	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0073H	Input switch control register	ISC	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0075H	Voltage reference control register	VREFAMPCNT	R/W	–	√	–	00H
F0076H	UARTMG0 clock doubler control register[Note 1]	CLKDCTL	R/W	√	√	–	00H
F0077H	Peripheral I/O redirection register 0	PIOR0	R/W	–	√	–	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0079H	A/D conversion clock control register	ADCKS	R/W	–	√	–	00H
F007AH	Frequency measurement circuit clock select register	FMCKS	R/W	√	√	–	00H
F007BH	Port mode select resister	PMS	R/W	√	√	–	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H
F0098H	Peripheral clock control register	PCKC	R/W	√	√	–	00H
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	–	√	–	Undefined Note 2
F00AAH	Flash operating mode select register	FLMODE	R/W	√	√	–	**Note 3**
F00ABH	Flash operating mode protect register	FLMWRP	R/W	√	√	–	00H

Notes 1. This register is incorporated in the 100-pin product, but is not incorporated in the 80-pin product.

2. The value set by FRQSEL2 to FRQSEL0 of the option byte 000C2H.

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

Table 3-6. Extended SFR (2nd SFR) List (2/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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 option control register	OSMC		R/W	√	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00F8H	Regulator mode control register	PMMC		R/W	√	√	–	00H
F00F9H	Power-on-reset status register	PORSR		R/W	–	√	–	00H^{Note}
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 adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–	–		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								

Note This register is reset only by a power-on reset.

Table 3-6. Extended SFR (2nd SFR) List (3/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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[Note]	–	√	√	0000H
F014DH		–			–	–		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W[Note]	–	√	√	0000H
F014FH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								

Note These registers are read-only in the R5F10NML.

Table 3-6. Extended SFR (2nd SFR) List (4/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0154H	Serial mode register 12	SMR12		R/W[Note 1]	–	–	√	0020H/ 0000H[Note 2]
F0155H								
F0156H	Serial mode register 13	SMR13		R/W[Note 1]	–	–	√	0020H/ 0000H[Note 2]
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[Note 1]	–	–	√	0087H/ 0000H[Note 3]
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W[Note 1]	–	–	√	0087H/ 0000H[Note 3]
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/ 0303H[Note 4]
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–	–		
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								

Notes 1. These registers are read-only in the R5F10NML.

2. R5F10NPL: 0020H
R5F10NML: 0000H

3. R5F10NPL: 0087H
R5F10NML: 0000H

4. R5F10NPL: 0F0FH
R5F10NML: 0303H

Table 3-6. Extended SFR (2nd SFR) List (5/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	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		–			–	–		

Table 3-6. Extended SFR (2nd SFR) List (6/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
F01C0H	Serial status register 20[Note]	SSR20L	SSR20	R	–	√	√	0000H
F01C1H		–			–	–		
F01C2H	Serial status register 21[Note]	SSR21L	SSR21	R	–	√	√	0000H
F01C3H		–			–	–		
F01C8H	Serial flag clear trigger register 20[Note]	SIR20L	SIR20	R/W	–	√	√	0000H
F01C9H		–			–	–		
F01CAH	Serial flag clear trigger register 21[Note]	SIR21L	SIR21	R/W	–	√	√	0000H
F01CBH		–			–	–		
F01D0H	Serial mode register 20[Note]	SMR20		R/W	–	–	√	0020H
F01D1H								
F01D2H	Serial mode register 21[Note]	SMR21		R/W	–	–	√	0020H
F01D3H								
F01D8H	Serial communication operation setting register 20[Note]	SCR20		R/W	–	–	√	0087H
F01D9H								
F01DAH	Serial communication operation setting register 21[Note]	SCR21		R/W	–	–	√	0087H
F01DBH								
F01E0H	Serial channel enable status register 2[Note]	SE2L	SE2	R	√	√	√	0000H
F01E1H		–			–	–		
F01E2H	Serial channel start register 2[Note]	SS2L	SS2	R/W	√	√	√	0000H
F01E3H		–			–	–		
F01E4H	Serial channel stop register 2[Note]	ST2L	ST2	R/W	√	√	√	0000H
F01E5H		–			–	–		
F01E6H	Serial clock select register 2[Note]	SPS2L	SPS2	R/W	–	√	√	0000H
F01E7H		–			–	–		
F01E8H	Serial output register 2[Note]	SO2		R/W	–	–	√	0303H
F01E9H								
F01EAH	Serial output enable register 2[Note]	SOE2L	SOE2	R/W	√	√	√	0000H
F01EBH		–			–	–		

Note These registers are incorporated in the 100-pin product, but are not incorporated in the 80-pin product.

Table 3-6. Extended SFR (2nd SFR) List (7/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01F4H	Serial output level register 2[Note]	SOL2L	SOL2	R/W	–	√	√	0000H
F01F5H		–			–	–		
F0210H	Transmit buffer register 0[Note]	TXBMG0		R/W	–	√	–	FFH
F0211H	Receive buffer register 0[Note]	RXBMG0		R	–	√	–	FFH
F0212H	Operating mode setting register 00[Note]	ASIMMG00		R/W	√	√	–	01H
F0213H	Operating mode setting register 01[Note]	ASIMMG01		R/W	√	√	–	1AH
F0214H	Baud rate generator control register 0[Note]	BRGCMG0		R/W	–	√	–	FFH
F0215H	Status register 0[Note]	ASISMG0		R	–	√	–	00H
F0216H	Status clear trigger register 0[Note]	ASCTMG0		R/W	√	√	–	00H
F0218H	Transmit buffer register 1[Note]	TXBMG1		R/W	–	√	–	FFH
F0219H	Receive buffer register 1[Note]	RXBMG1		R	–	√	–	FFH
F021AH	Operating mode setting register 10[Note]	ASIMMG10		R/W	√	√	–	01H
F021BH	Operating mode setting register 11[Note]	ASIMMG11		R/W	√	√	–	1AH
F021CH	Baud rate generator control register 1[Note]	BRGCMG1		R/W	–	√	–	FFH
F021DH	Status register 1[Note]	ASISMG1		R	–	√	–	00H
F021EH	Status clear trigger register 1[Note]	ASCTMG1		R/W	√	√	–	00H
F0230H	IICA control register 00	IICCTL00		R/W	√	√	–	00H
F0231H	IICA control register 01	IICCTL01		R/W	√	√	–	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	–	√	–	FFH
F0234H	Slave address register 0	SVA0		R/W	–	√	–	00H
F0240H	Event output destination select register 00	ELSELR00		R/W	–	√	–	00H
F0241H	Event output destination select register 01	ELSELR01		R/W	–	√	–	00H
F0242H	Event output destination select register 02	ELSELR02		R/W	–	√	–	00H
F0243H	Event output destination select register 03	ELSELR03		R/W	–	√	–	00H
F0244H	Event output destination select register 04	ELSELR04		R/W	–	√	–	00H
F0245H	Event output destination select register 05	ELSELR05		R/W	–	√	–	00H
F0246H	Event output destination select register 06	ELSELR06		R/W	–	√	–	00H
F0247H	Event output destination select register 07	ELSELR07		R/W	–	√	–	00H
F0248H	Event output destination select register 08	ELSELR08		R/W	–	√	–	00H
F0249H	Event output destination select register 09	ELSELR09		R/W	–	√	–	00H
F024AH	Event output destination select register 10	ELSELR10		R/W	–	√	–	00H
F024BH	Event output destination select register 11	ELSELR11		R/W	–	√	–	00H
F024CH	Event output destination select register 12	ELSELR12		R/W	–	√	–	00H
F024DH	Event output destination select register 13	ELSELR13		R/W	–	√	–	00H
F024EH	Event output destination select register 14	ELSELR14		R/W	–	√	–	00H
F024FH	Event output destination select register 15	ELSELR15		R/W	–	√	–	00H

Note These registers are incorporated in the 100-pin product, but are not incorporated in the 80-pin product.

Table 3-6. Extended SFR (2nd SFR) List (8/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0250H	Event output destination select register 16	ELSELR16	R/W	–	√	–	00H
F0251H	Event output destination select register 17	ELSELR17	R/W	–	√	–	00H
F0252H	Event output destination select register 18	ELSELR18	R/W	–	√	–	00H
F0253H	Event output destination select register 19	ELSELR19	R/W	–	√	–	00H
F0254H	Event output destination select register 20	ELSELR20	R/W	–	√	–	00H
F0255H	Event output destination select register 21	ELSELR21	R/W	–	√	–	00H
F0256H	Event output destination select register 22	ELSELR22	R/W	–	√	–	00H
F0257H	Event output destination select register 23	ELSELR23	R/W	–	√	–	00H
F0258H	Event output destination select register 24	ELSELR24	R/W	–	√	–	00H
F0259H	Event output destination select register 25	ELSELR25	R/W	–	√	–	00H
F0260H	Event output destination select register 26	ELSELR26	R/W	–	√	–	00H
F0261H	Event output destination select register 27	ELSELR27	R/W	–	√	–	00H
F0262H	Event output destination select register 28	ELSELR28	R/W	–	√	–	00H
F0263H	Event output destination select register 29	ELSELR29	R/W	–	√	–	00H
F0280H	Multiplication data register A (L) (unsigned)	MUL32UL	R/W	–	–	√	0000H
F0281H							
F0282H	Multiplication data register A (H) (unsigned)	MUL32UH	R/W	–	–	√	0000H
F0283H							
F0284H	Multiplication data register A (L) (signed)	MUL32SL	R/W	–	–	√	0000H
F0285H							
F0286H	Multiplication data register A (H) (signed)	MUL32SH	R/W	–	–	√	0000H
F0287H							
F0288H	Multiply-accumulation data register A (L) (unsigned)	MAC32UL	R/W	–	–	√	0000H
F0289H							
F028AH	Multiply-accumulation data register A (H) (unsigned)	MAC32UH	R/W	–	–	√	0000H
F028BH							
F028CH	Multiply-accumulation data register A (L) (signed)	MAC32SL	R/W	–	–	√	0000H
F028DH							
F028EH	Multiply-accumulation data register A (H) (signed)	MAC32SH	R/W	–	–	√	0000H
F028FH							
F0290H	Multiplication result register 0	MULR0	R/W	–	–	√	0000H
F0291H							
F0292H	Multiplication result register 1	MULR1	R/W	–	–	√	0000H
F0293H							
F0294H	Multiplication result register 2	MULR2	R/W	–	–	√	0000H
F0295H							
F0296H	Multiplication result register 3	MULR3	R/W	–	–	√	0000H
F0297H							
F029AH	Multiplication control register	MULC	R/W	√	√	–	00H
F029BH	Multiplication result select register	MULRSEL	R/W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (9/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F02A0H	Timer RJ control register 0	TRJCR0	R/W	–	√	–	00H
F02A1H	Timer RJ I/O control register 0	TRJIOC0	R/W	√	√	–	00H
F02A2H	Timer RJ mode register 0	TRJMR0	R/W	√	√	–	00H
F02A3H	Timer RJ event pin select register 0	TRJISR0	R/W	√	√	–	00H
F02A4H	Timer RJ control register 1	TRJCR1	R/W	–	√	–	00H
F02A5H	Timer RJ I/O control register 1	TRJIOC1	R/W	√	√	–	00H
F02A6H	Timer RJ mode register 1	TRJMR1	R/W	√	√	–	00H
F02A7H	Timer RJ event pin select register 1	TRJISR1	R/W	√	√	–	00H
F02B0H	SMOTD compare register A0	SMOTDCRSA0	R/W	–	√	–	00H
F02B1H	SMOTD compare register B0	SMOTDCRSB0	R/W	–	√	–	00H
F02B2H	SMOTD clock select register 0	SMOTDTCS0	R/W	–	√	–	00H
F02B3H	SMOTD control register 0	SMOTDCR0	R/W	√	√	–	00H
F02B4H	SMOTD sampling level setting register 0	SMOTDSMS0	R/W	√	√	–	00H
F02B5H	SMOTD sampling pin status register 0	SMOTDSMD0	R	√	√	–	00H
F02B6H	SMOTD output control register 0	SMOTDOE0	R/W	√	√	–	00H
F02B8H	SMOTD compare register A1	SMOTDCRSA1	R/W	–	√	–	00H
F02B9H	SMOTD compare register B1	SMOTDCRSB1	R/W	–	√	–	00H
F02BAH	SMOTD clock select register 1	SMOTDTCS1	R/W	–	√	–	00H
F02BBH	SMOTD control register 1	SMOTDCR1	R/W	√	√	–	00H
F02BCH	SMOTD sampling level setting register 1	SMOTDSMS1	R/W	√	√	–	00H
F02BDH	SMOTD sampling pin status register 1	SMOTDSMD1	R	√	√	–	00H
F02BEH	SMOTD output control register 1	SMOTDOE1	R/W	√	√	–	00H
F02D0H	Oscillation stop detection control register	OSDC	R/W	–	–	√	0FFFH
F02D1H							
F02D8H	High-speed on-chip oscillator clock frequency correction control register	HOCOFC	R/W	–	√	–	00H
F02E0H	DTC base address register	DTCBAR	R/W	–	√	–	00H
F02E5H	PLL control register	DSCCTL	R/W	√	√	–	00H
F02E6H	Main clock control register	MCKC	R/W	√	√	–	00H
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
F02F3H							
F02FAH	CRC data register	CRCD	R/W	–	–	√	0000H
F02FBH							
F0300H	LCD port function register 0	PFSEG0	R/W	√	√	–	FFH
F0301H	LCD port function register 1	PFSEG1	R/W	√	√	–	FFH

Table 3-6. Extended SFR (2nd SFR) List (10/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0302H	LCD port function register 2	PFSEG2		R/W	√	√	–	FFH
F0303H	LCD port function register 3	PFSEG3		R/W	√	√	–	FFH
F0304H	LCD port function register 4	PFSEG4		R/W	√	√	–	FFH/0FH Note 1
F0305H	LCD port function register 5	PFSEG5		R/W	√	√	–	FFH/3FH Note 2
F0306H	LCD port function register 6	PFSEG6		R/W	√	√	–	03H
F0308H	LCD input switch control register	ISCLCD		R/W	√	√	–	00H
F0312H	Frequency measurement count register L	FMCRL		R	–	–	√	0000H
F0313H								
F0314H	Frequency measurement count register H	FMCRH		R	–	–	√	0000H
F0315H								
F0316H	Frequency measurement control register	FMCTL		R/W	√	√	–	00H
F0332H	V_{DD} pin voltage detection control register	LVDVDD		R/W	√	√	–	00H
F0333H	VBAT pin voltage detection control register	LVDVBAT		R/W	√	√	–	00H
F0334H	VRTC pin voltage detection control register	LVDVRTC		R/W	√	√	–	00H
F0335H	EXLVD pin voltage detection control register	LVDEXLVD		R/W	√	√	–	00H
F0350H	8-bit interval timer compare register 00	TRTCMP00	TRTCMP0	R/W	–	√	√	FFH
F0351H	8-bit interval timer compare register 01	TRTCMP01		R/W	–	√		FFH
F0352H	8-bit interval timer control register 0	TRTCR0		R/W	√	√	–	00H
F0353H	8-bit interval timer frequency division register 0	TRTMD0		R/W	–	√	–	00H
F0358H	8-bit interval timer compare register 10	TRTCMP10	TRTCMP1	R/W	–	√	√	FFH
F0359H	8-bit interval timer compare register 11	TRTCMP11		R/W	–	√		FFH
F035AH	8-bit interval timer control register 1	TRTCR1		R/W	√	√	–	00H
F035BH	8-bit interval timer frequency division register 1	TRTMD1		R/W	–	√	–	00H
F0360H	8-bit interval timer compare register 20	TRTCMP20	TRTCMP2	R/W	–	√	√	FFH
F0361H	8-bit interval timer compare register 21	TRTCMP22		R/W	–	√		FFH
F0362H	8-bit interval timer control register 2	TRTCR2		R/W	√	√	–	00H
F0363H	8-bit interval timer frequency division register 2	TRTMD2		R/W	–	√	–	00H
F0368H	8-bit interval timer compare register 30	TRTCMP30	TRTCMP3	R/W	–	√	√	FFH
F0369H	8-bit interval timer compare register 31	TRTCMP33		R/W	–	√		FFH
F036AH	8-bit interval timer control register 3	TRTCR3		R/W	√	√	–	00H
F036BH	8-bit interval timer frequency division register 3	TRTMD3		R/W	–	√	–	00H
F0380H	RTC power-on-reset status register	RTCPORSR		R/W	–	√	–	00H
F0382H	Noise filter enable register for RTCICn pin (n = 0 to 2)	RTCICNFEN		R/W	–	√	–	00H
F0384H	Sub clock operation mode control register	SCMC		R/W	–	√	–	00H Note 3
F0386H	Sub clock operation status control register	SCSC		R/W	√	√	–	40H
F03A0H	IrDA control register	IRCR		R/W	√	√	–	00H
F03B0H	Temperature sensor control test register	TMPCTL		R/W	√	√	–	00H

Notes 1. The value after a reset is FFH for the 100-pin product or 0FH for the 80-pin product.

2. The value after a reset is FFH for the 100-pin product or 3FH for the 80-pin product.

3. This register is reset only by a power-on reset.

Table 3-6. Extended SFR (2nd SFR) List (11/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0400H	LCD display data memory 0	SEG0	R/W	–	√	–	00H
F0401H	LCD display data memory 1	SEG1	R/W	–	√	–	00H
F0402H	LCD display data memory 2	SEG2	R/W	–	√	–	00H
F0403H	LCD display data memory 3	SEG3	R/W	–	√	–	00H
F0404H	LCD display data memory 4	SEG4	R/W	–	√	–	00H
F0405H	LCD display data memory 5	SEG5	R/W	–	√	–	00H
F0406H	LCD display data memory 6	SEG6	R/W	–	√	–	00H
F0407H	LCD display data memory 7	SEG7	R/W	–	√	–	00H
F0408H	LCD display data memory 8	SEG8	R/W	–	√	–	00H
F0409H	LCD display data memory 9	SEG9	R/W	–	√	–	00H
F040AH	LCD display data memory 10	SEG10	R/W	–	√	–	00H
F040BH	LCD display data memory 11	SEG11	R/W	–	√	–	00H
F040CH	LCD display data memory 12	SEG12	R/W	–	√	–	00H
F040DH	LCD display data memory 13	SEG13	R/W	–	√	–	00H
F040EH	LCD display data memory 14	SEG14	R/W	–	√	–	00H
F040FH	LCD display data memory 15	SEG15	R/W	–	√	–	00H
F0410H	LCD display data memory 16	SEG16	R/W	–	√	–	00H
F0411H	LCD display data memory 17	SEG17	R/W	–	√	–	00H
F0412H	LCD display data memory 18	SEG18	R/W	–	√	–	00H
F0413H	LCD display data memory 19	SEG19	R/W	–	√	–	00H
F0414H	LCD display data memory 20	SEG20	R/W	–	√	–	00H
F0415H	LCD display data memory 21	SEG21	R/W	–	√	–	00H
F0416H	LCD display data memory 22	SEG22	R/W	–	√	–	00H
F0417H	LCD display data memory 23	SEG23	R/W	–	√	–	00H
F0418H	LCD display data memory 24	SEG24	R/W	–	√	–	00H
F0419H	LCD display data memory 25	SEG25	R/W	–	√	–	00H
F041AH	LCD display data memory 26	SEG26	R/W	–	√	–	00H
F041BH	LCD display data memory 27	SEG27	R/W	–	√	–	00H
F041CH	LCD display data memory 28	SEG28	R/W	–	√	–	00H
F041DH	LCD display data memory 29	SEG29	R/W	–	√	–	00H
F041EH	LCD display data memory 30	SEG30	R/W	–	√	–	00H
F041FH	LCD display data memory 31	SEG31	R/W	–	√	–	00H
F0420H	LCD display data memory 32	SEG32	R/W	–	√	–	00H
F0421H	LCD display data memory 33	SEG33	R/W	–	√	–	00H
F0422H	LCD display data memory 34	SEG34	R/W	–	√	–	00H
F0423H	LCD display data memory 35	SEG35	R/W	–	√	–	00H
F0424H	LCD display data memory 36	SEG36	R/W	–	√	–	00H
F0425H	LCD display data memory 37	SEG37	R/W	–	√	–	00H
F0426H	LCD display data memory 38	SEG38	R/W	–	√	–	00H
F0427H	LCD display data memory 39	SEG39	R/W	–	√	–	00H
F0428H	LCD display data memory 40	SEG40	R/W	–	√	–	00H
F0429H	LCD display data memory 41	SEG41	R/W	–	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (12/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0540H	8-bit interval timer counter register 00	TRT00	TRT0	R	–	√	√	00H
F0541H	8-bit interval timer counter register 01	TRT01		R	–	√		00H
F0548H	8-bit interval timer counter register 10	TRT10	TRT1	R	–	√	√	00H
F0549H	8-bit interval timer counter register 11	TRT11		R	–	√		00H
F0550H	8-bit interval timer counter register 20	TRT20	TRT2	R	–	√	√	00H
F0551H	8-bit interval timer counter register 21	TRT21		R	–	√		00H
F0558H	8-bit interval timer counter register 30	TRT30	TRT3	R	–	√	√	00H
F0559H	8-bit interval timer counter register 31	TRT31		R	–	√		00H
F0560H	Timer RJ counter register 0	TRJ0		R/W	–	–	√	FFFFH
F0561H								
F0562H	Timer RJ counter register 1	TRJ1		R/W	–	–	√	FFFFH
F0563H								
F0581H	64-Hz counter	R64CNT		R	–	√	–	Undefined
F0583H	Second counter	RSECCNT		R/W	–	√	–	Undefined
	Binary counter 0	BCNT0		R/W	–	√	–	Undefined
F0585H	Minute counter	RMINCNT		R/W	–	√	–	Undefined
	Binary counter 1	BCNT1		R/W	–	√	–	Undefined
F0587H	Hour counter	RHRCNT		R/W	–	√	–	Undefined
	Binary counter 2	BCNT2		R/W	–	√	–	Undefined
F0589H	Week counter	RWKCNT		R/W	–	√	–	Undefined
	Binary counter 3	BCNT3		R/W	–	√	–	Undefined
F058BH	Day counter	RDAYCNT		R/W	–	√	–	Undefined
F058DH	Month counter	RMONCNT		R/W	–	√	–	Undefined
F058EH	Year counter	RYRCNT		R/W	–	–	√	Undefined
F058FH								
F0591H	Second alarm register 0	RSECAR0		R/W	–	√	–	Undefined
	Binary counter 0 alarm register 0	BCNT0AR0		R/W	–	√	–	Undefined
F0593H	Minute alarm register 0	RMINAR0		R/W	–	√	–	Undefined
	Binary counter 1 alarm register 0	BCNT1AR0		R/W	–	√	–	Undefined
F0595H	Hour alarm register 0	RHRAR0		R/W	–	√	–	Undefined
	Binary counter 2 alarm register 0	BCNT2AR0		R/W	–	√	–	Undefined
F0597H	Week alarm register 0	RWKAR0		R/W	–	√	–	Undefined
	Binary counter 3 alarm register 0	BCNT3AR0		R/W	–	√	–	Undefined
F0599H	Day alarm register 0	RDAYAR0		R/W	–	√	–	Undefined
	Binary counter 0 alarm enable register 0	BCNT0AER0		R/W	–	√	–	Undefined
F059BH	Month alarm register 0	RMONAR0		R/W	–	√	–	Undefined
	Binary counter 1 alarm enable register 0	BCNT1AER0		R/W	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (13/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F059CH	Year alarm register 0	RYRAR0	R/W	–	–	√	Undefined
	Binary counter 2 alarm enable register 0	BCNT2AER0	R/W	–	–	√	Undefined
F059FH	Year alarm enable register 0	RYRAREN0	R/W	–	√	–	Undefined
	Binary counter 3 alarm enable register 0	BCNT3AER0	R/W	–	√	–	Undefined
F05A1H	RTC status register	RSR	R/W	–	√	–	Undefined
F05A3H	RTC control register 1	RCR1	R/W	–	√	–	Undefined
F05A5H	RTC control register 2	RCR2	R/W	–	√	–	Undefined
F05A7H	RTC control register 3	RCR3	R/W	–	√	–	Undefined
F05A9H	RTC control register 4	RCR4	R/W	–	√	–	Undefined
F05AFH	Time error correction register	RADJ	R/W	–	√	–	Undefined
F05B1H	Second alarm register 1	RSECAR1	R/W	–	√	–	Undefined
	Binary counter 0 alarm register 1	BCNT0AR1	R/W	–	√	–	Undefined
F05B3H	Minute alarm register 1	RMINAR1	R/W	–	√	–	Undefined
	Binary counter 1 alarm register 1	BCNT1AR1	R/W	–	√	–	Undefined
F05B5H	Hour alarm register 1	RHRAR1	R/W	–	√	–	Undefined
	Binary counter 2 alarm register 1	BCNT2AR1	R/W	–	√	–	Undefined
F05B7H	Week alarm register 1	RWKAR1	R/W	–	√	–	Undefined
	Binary counter 3 alarm register 1	BCNT3AR1	R/W	–	√	–	Undefined
F05B9H	Day alarm register 1	RDAYAR1	R/W	–	√	–	Undefined
	Binary counter 0 alarm enable register 1	BCNT0AER1	R/W	–	√	–	Undefined
F05BBH	Month alarm register 1	RMONAR1	R/W	–	√	–	Undefined
	Binary counter 1 alarm enable register 1	BCNT1AER1	R/W	–	√	–	Undefined
F05BCH	Year alarm register 1	RYRAR1	R/W	–	–	√	Undefined
	Binary counter 2 alarm enable register 1	BCNT2AER1	R/W	–	–	√	Undefined
F05BFH	Year alarm enable register 1	RYRAREN1	R/W	–	√	–	Undefined
	Binary counter 3 alarm enable register 1	BCNT3AER1	R/W	–	√	–	Undefined
F05C1H	Time capture control register 0	RTCCR0	R/W	–	√	–	Undefined
F05C3H	Time capture control register 1	RTCCR1	R/W	–	√	–	Undefined
F05C5H	Time capture control register 2	RTCCR2	R/W	–	√	–	Undefined
F05D3H	Second capture register 0	RSECCP0	R	–	√	–	Undefined
	BCNT0 capture register 0	BCNT0CP0	R	–	√	–	Undefined
F05D5H	Minute capture register 0	RMINCP0	R	–	√	–	Undefined
	BCNT1 capture register 0	BCNT1CP0	R	–	√	–	Undefined
F05D7H	Hour capture register 0	RHRCP0	R	–	√	–	Undefined
	BCNT2 capture register 0	BCNT2CP0	R	–	√	–	Undefined
F05DBH	Day capture register 0	RDAYCP0	R	–	√	–	Undefined
	BCNT3 capture register 0	BCNT3CP0	R	–	√	–	Undefined

Table 3-6. Extended SFR (2nd SFR) List (14/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F05DDH	Month capture register 0	RMONCP0	R	–	√	–	Undefined
F05E3H	Second capture register 1	RSECCP1	R	–	√	–	Undefined
	BCNT0 capture register 1	BCNT0CP1	R	–	√	–	Undefined
F05E5H	Minute capture register 1	RMINCP1	R	–	√	–	Undefined
	BCNT1 capture register 1	BCNT1CP1	R	–	√	–	Undefined
F05E7H	Hour capture register 1	RHRCP1	R	–	√	–	Undefined
	BCNT2 capture register 1	BCNT2CP1	R	–	√	–	Undefined
F05EBH	Day capture register 1	RDAYCP1	R	–	√	–	Undefined
	BCNT3 capture register 1	BCNT3CP1	R	–	√	–	Undefined
F05EDH	Month capture register 1	RMONCP1	R	–	√	–	Undefined
F05F3H	Second capture register 2	RSECCP2	R	–	√	–	Undefined
	BCNT0 capture register 2	BCNT0CP2	R	–	√	–	Undefined
F05F5H	Minute capture register 2	RMINCP2	R	–	√	–	Undefined
	BCNT1 capture register 2	BCNT1CP2	R	–	√	–	Undefined
F05F7H	Hour capture register 2	RHRCP2	R	–	√	–	Undefined
	BCNT2 capture register 2	BCNT2CP2	R	–	√	–	Undefined
F05FBH	Day capture register 2	RDAYCP2	R	–	√	–	Undefined
	BCNT3 capture register 2	BCNT3CP2	R	–	√	–	Undefined
F05FDH	Month capture register 2	RMONCP2	R	–	√	–	Undefined
F0600H	A/D control register	ADCSR	R/W	–	–	√	0000H
F0601H							
F0604H	A/D channel select register A0	ADANSA0	R/W	–	–	√	0000H
F0605H							
F0608H	A/D-converted value addition/average function channel select register 0	ADADS0	R/W	–	–	√	0000H
F0609H							
F060CH	A/D-converted value addition/average count select register	ADADC	R/W	√	√	–	00H
F060EH	A/D control extended register	ADCER	R/W	–	–	√	0000H
F061EH							
F0610H	A/D conversion start trigger select register	ADSTRGR	R/W	–	–	√	0000H
F0611H							
F0612H	A/D conversion extended input control register	ADEXICR	R/W	–	–	√	0000H
F0613H							
F061AH	A/D temperature sensor data register	ADTSDR	R	–	–	√	0000H
F061BH							
F061CH	A/D internal reference voltage data register	ADOCDR	R	–	–	√	0000H
F061DH							
F061EH	A/D self-diagnosis data register	ADRD	R	–	–	√	0000H
F061FH							

Table 3-6. Extended SFR (2nd SFR) List (15/16)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0620H	A/D data register 0	ADDR0	R	–	–	√	0000H
F0621H							
F0622H	A/D data register 1	ADDR1	R	–	–	√	0000H
F0623H							
F0624H	A/D data register 2	ADDR2	R	–	–	√	0000H
F0625H							
F0626H	A/D data register 3	ADDR3	R	–	–	√	0000H
F0627H							
F0628H	A/D data register 4	ADDR4	R	–	–	√	0000H
F0629H							
F062AH	A/D data register 5	ADDR5	R	–	–	√	0000H
F062BH							
F0666H	A/D sample-and-hold circuit control register	ADSHCR	R/W	–	–	√	001AH
F0667H							
F068AH	A/D high-potential/low-potential reference voltage control register	ADHVREFCNT	R/W	√	√	–	00H
F06DEH	A/D sampling state register T	ADSSTRT	R/W	–	√	–	00DH
F06DFH	A/D sampling state register O	ADSSTRO	R/W	–	√	–	00DH
F06E0H	A/D sampling state register 0	ADSSTR0	R/W	–	√	–	00DH
F06E1H	A/D sampling state register 1	ADSSTR1	R/W	–	√	–	00DH
F06E2H	A/D sampling state register 2	ADSSTR2	R/W	–	√	–	00DH
F06E3H	A/D sampling state register 3	ADSSTR3	R/W	–	√	–	00DH
F06E4H	A/D sampling state register 4	ADSSTR4	R/W	–	√	–	00DH
F06E5H	A/D sampling state register 5	ADSSTR5	R/W	–	√	–	00DH
F0740H	ΔΣ A/D converter mode register	DSADMR	R/W	–	–	√	0000H
F0741H							
F0742H	ΔΣ A/D converter gain control register 0	DSADGCR0	R/W	–	√	–	00H
F0743H	ΔΣ A/D converter gain control register 1	DSADGCR1	R/W	–	√	–	00H
F0745H	ΔΣ A/D converter HPF control register	DSADHPFCR	R/W	–	√	–	00H
F0746H	ΔΣ A/D converter decimation filter control register	DSADDECCR	R/W	√	√	–	00H
F0750H	ΔΣ A/D converter phase control register 0	DSADPHCR0	R/W	–	–	√	0000H
F0751H							
F0752H	ΔΣ A/D converter phase control register 1	DSADPHCR1	R/W	–	–	√	0000H
F0753H							
F0754H	ΔΣ A/D converter phase control register 2	DSADPHCR2	R/W	–	–	√	0000H
F0755H							
F0756H	ΔΣ A/D converter phase control register 3[Note]	DSADPHCR3	R/W	–	–	√	0000H
F0757H							

Note This register is incorporated in the 100-pin product, but is not incorporated in the 80-pin product.

Table 3-6. Extended SFR (2nd SFR) List (16/16)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0760H	ΔΣ A/D converter conversion result register 0L	DSADCR0L	DSADCR0	R	–	√	√	00H
F0761H	ΔΣ A/D converter conversion result register 0M	DSADCR0M		R	–	√		00H
F0762H	ΔΣ A/D converter conversion result register 0H	DSADCR0H		R	–	√	–	00H
F0764H	ΔΣ A/D converter conversion result register 1L	DSADCR1L	DSADCR1	R	–	√	√	00H
F0765H	ΔΣ A/D converter conversion result register 1M	DSADCR1M		R	–	√		00H
F0766H	ΔΣ A/D converter conversion result register 1H	DSADCR1H		R	–	√	–	00H
F0768H	ΔΣ A/D converter conversion result register 2L	DSADCR2L	DSADCR2	R	–	√	√	00H
F0769H	ΔΣ A/D converter conversion result register 2M	DSADCR2M		R	–	√		00H
F076AH	ΔΣ A/D converter conversion result register 2H	DSADCR2H		R	–	√	–	00H
F076CH	ΔΣ A/D converter conversion result register 3L[Note]	DSADCR3L	DSADCR3	R	–	√	√	00H
F076DH	ΔΣ A/D converter conversion result register 3M[Note]	DSADCR3M		R	–	√		00H
F076EH	ΔΣ A/D converter conversion result register 3H[Note]	DSADCR3H		R	–	√	–	00H
F0770H	ΔΣ A/D converter conversion result register (LPF) 0L	DSADCRLPF0L	DSADCRLPF0	R	–	√	√	00H
F0771H	ΔΣ A/D converter conversion result register (LPF) 0M	DSADCRLPF0M		R	–	√		00H
F0772H	ΔΣ A/D converter conversion result register (LPF) 0H	DSADCRLPF0H		R	–	√	–	00H
F0774H	ΔΣ A/D converter conversion result register (LPF) 1L	DSADCRLPF1L	DSADCRLPF1	R	–	√	√	00H
F0775H	ΔΣ A/D converter conversion result register (LPF) 1M	DSADCRLPF1M		R	–	√		00H
F0776H	ΔΣ A/D converter conversion result register (LPF) 1H	DSADCRLPF1H		R	–	√	–	00H
F0778H	ΔΣ A/D converter conversion result register (LPF) 2L	DSADCRLPF2L	DSADCRLPF2	R	–	√	√	00H
F0779H	ΔΣ A/D converter conversion result register (LPF) 2M	DSADCRLPF2M		R	–	√		00H
F077AH	ΔΣ A/D converter conversion result register (LPF) 2H	DSADCRLPF2H		R	–	√	–	00H
F077CH	ΔΣ A/D converter conversion result register (LPF) 3L[Note]	DSADCRLPF3L	DSADCRLPF3	R	–	√	√	00H
F077DH	ΔΣ A/D converter conversion result register (LPF) 3M[Note]	DSADCRLPF3M		R	–	√		00H
F077EH	ΔΣ A/D converter conversion result register (LPF) 3H[Note]	DSADCRLPF3H		R	–	√	–	00H

Note These registers are incorporated in the 100-pin product, but are not incorporated in the 80-pin product.

Remark For SFRs in the SFR area, see **Table 3-5 SFR List**.

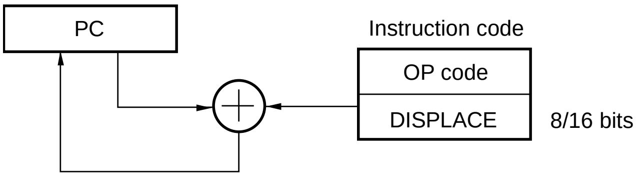
3.3 Instruction Address Addressing

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 3-11. 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 3-12. Example of CALL !!addr20/BR !!addr20

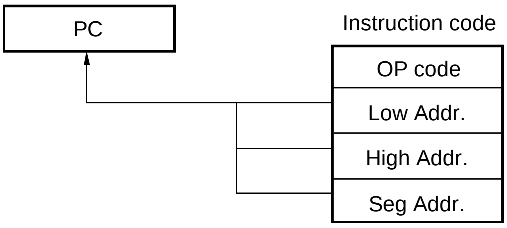
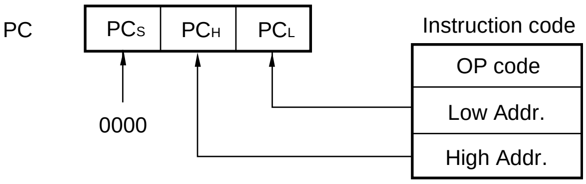


Figure 3-13. Example of CALL !addr16/BR !addr16



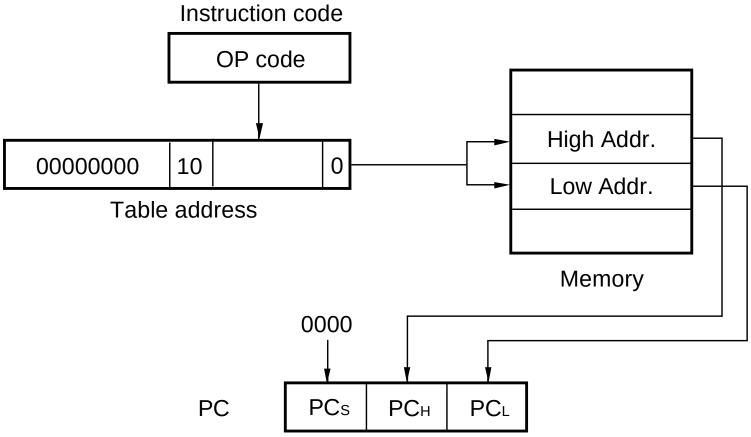
3.3.3 Table indirect addressing

[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 KB space from 00000H to 0FFFFH.

Figure 3-14. Outline of Table Indirect Addressing

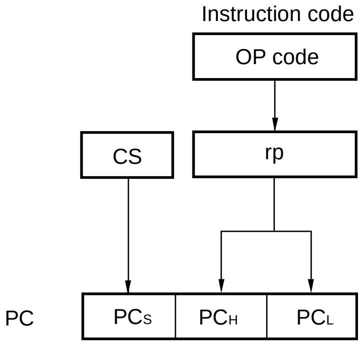


3.3.4 Register direct addressing

[Function]

Register direct 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 direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-15. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

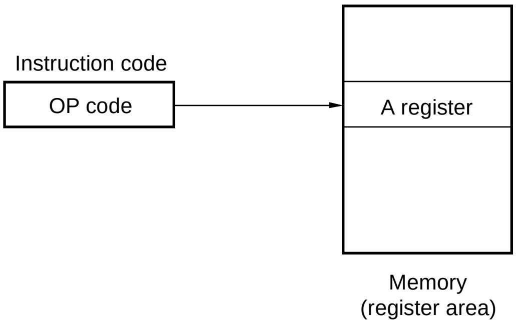
[Function]

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 3-16. Outline of Implied Addressing



3.4.2 Register addressing

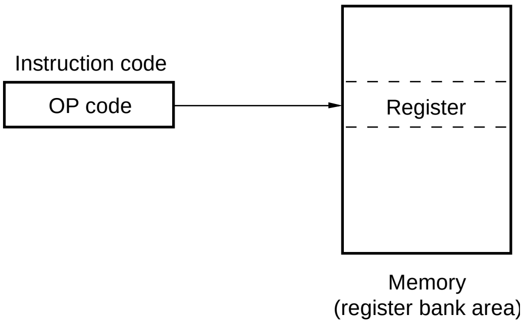
[Function]

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 3-17. Outline of Register Addressing



3.4.3 Direct addressing

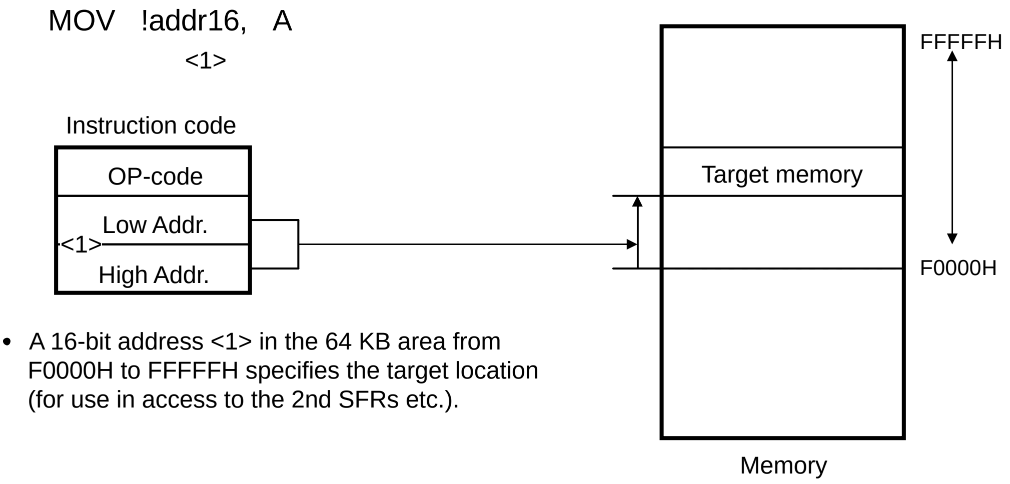
[Function]

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

[Operand format]

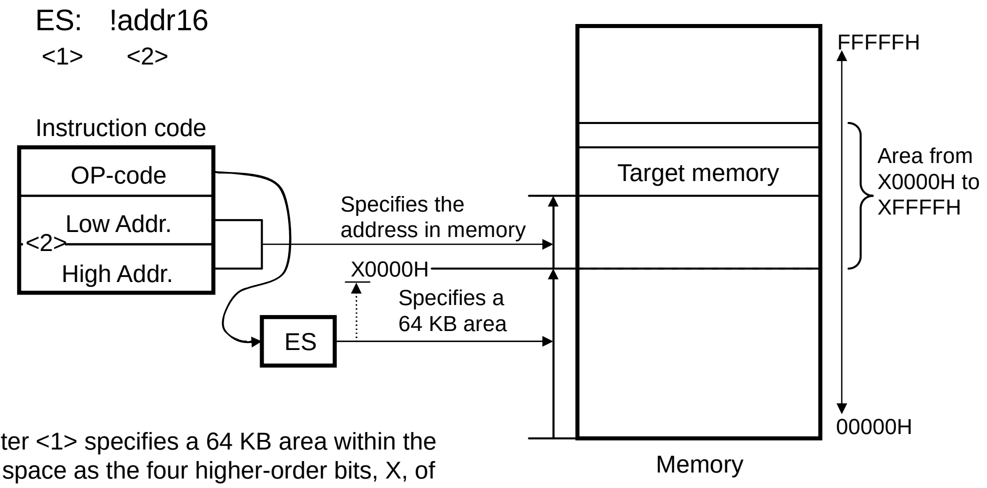
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-18. Example of !addr16



- A 16-bit address <1> in the 64 KB area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

Figure 3-19. Example of ES:!addr16



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

3.4.4 Short direct addressing

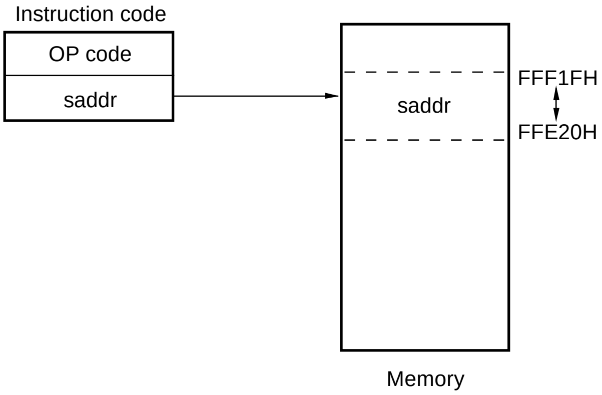
[Function]

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 3-20. 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

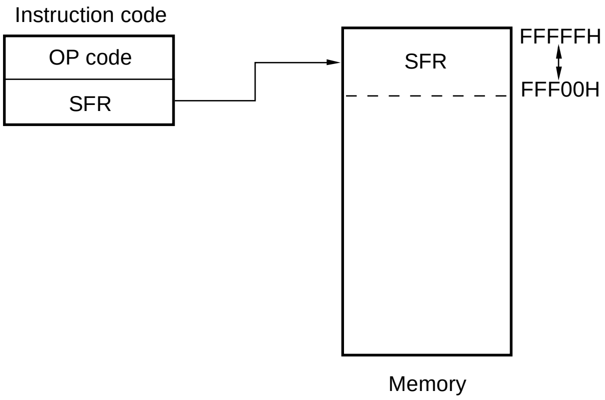
[Function]

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 3-21. Outline of SFR Addressing



3.4.6 Register indirect addressing

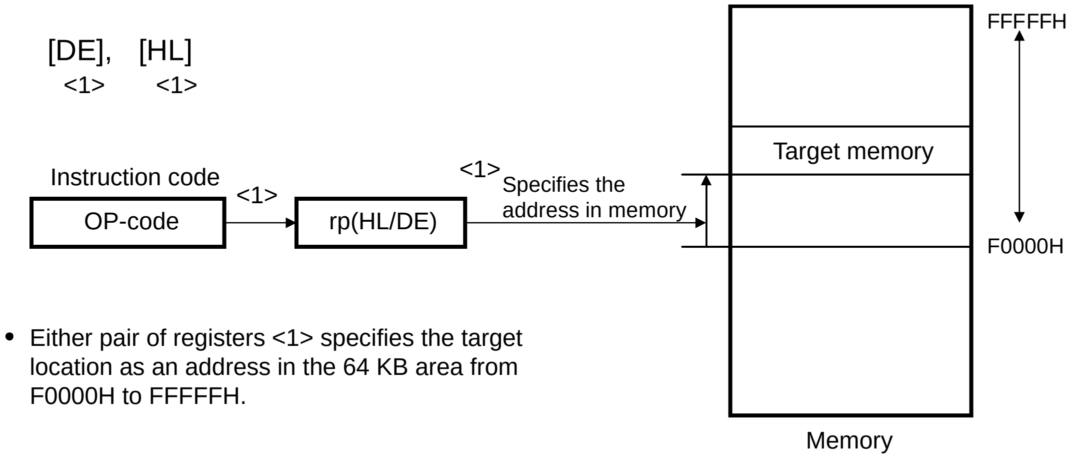
[Function]

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]

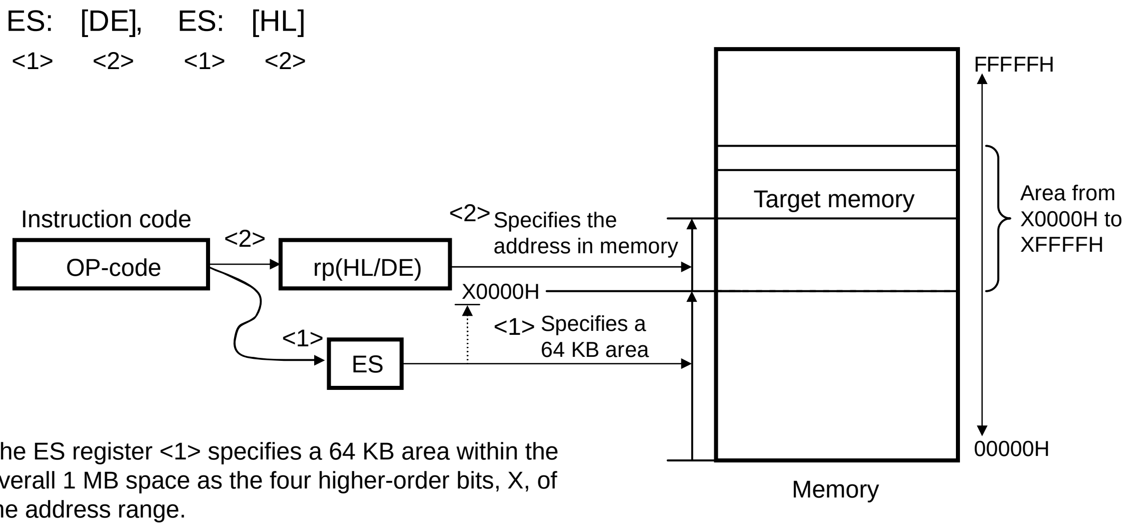
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-22. Example of [DE], [HL]



- Either pair of registers <1> specifies the target location as an address in the 64 KB area from F0000H to FFFFFH.

Figure 3-23. Example of ES:[DE], ES:[HL]



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

3.4.7 Based addressing

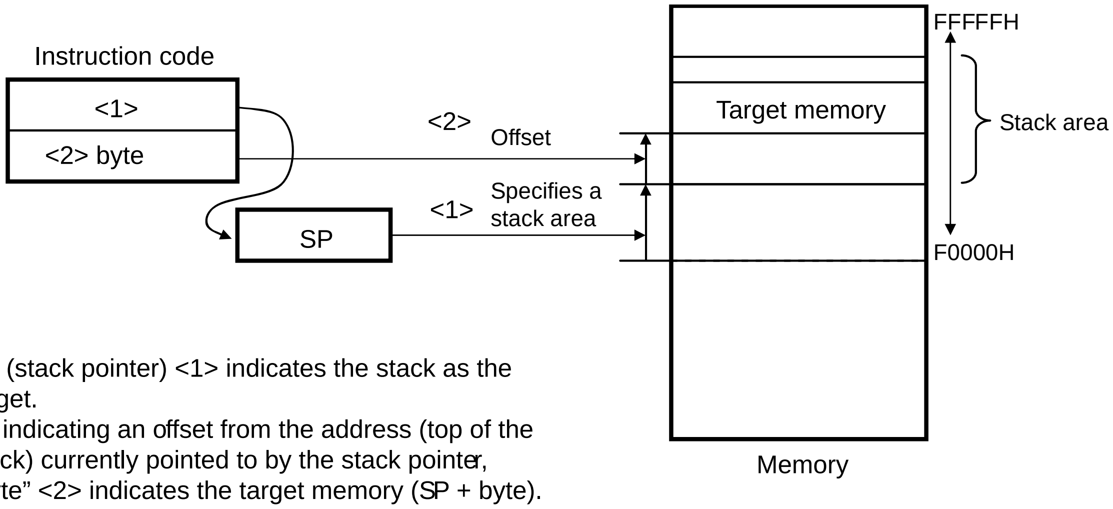
[Function]

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]

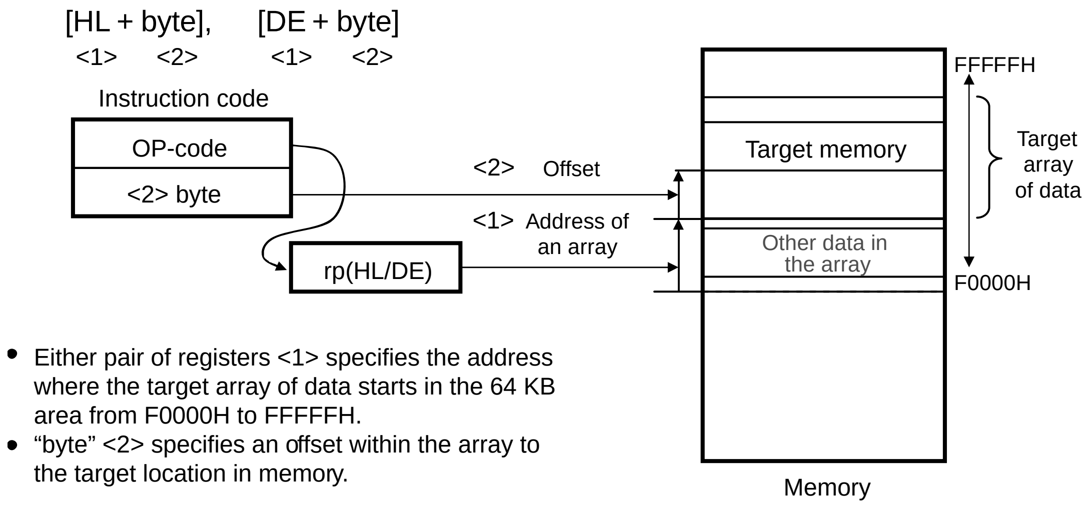
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit 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 3-24. Example of [SP+byte]



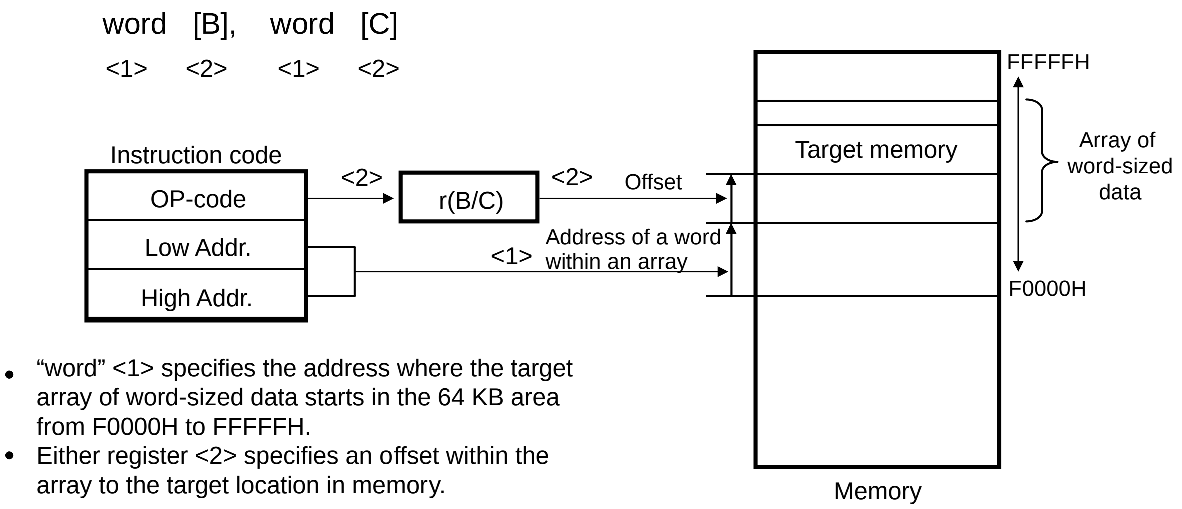
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-25. Example of [HL + byte], [DE + byte]



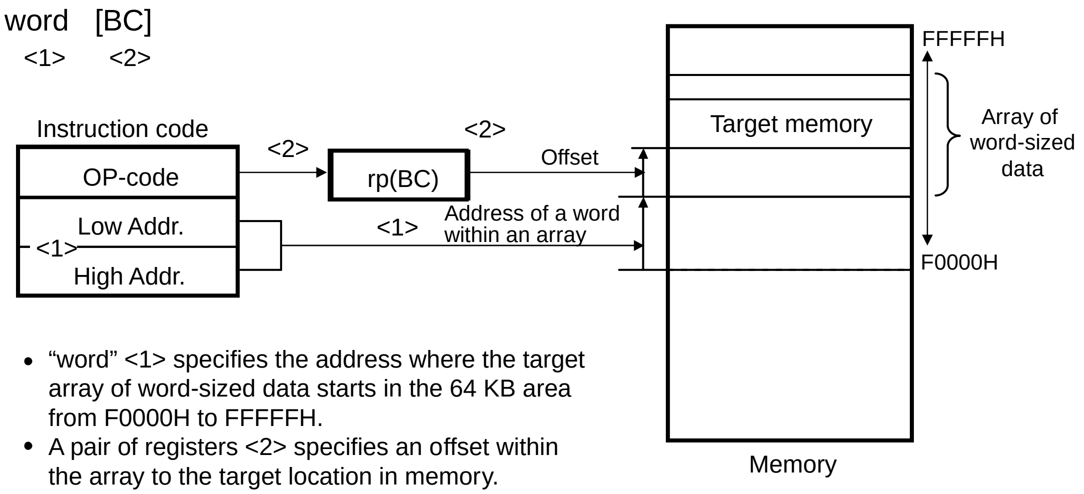
- Either pair of registers <1> specifies the address where the target array of data starts in the 64 KB area from F0000H to FFFFFH.
- “byte” <2> specifies an offset within the array to the target location in memory.

Figure 3-26. Example of word[B], word[C]



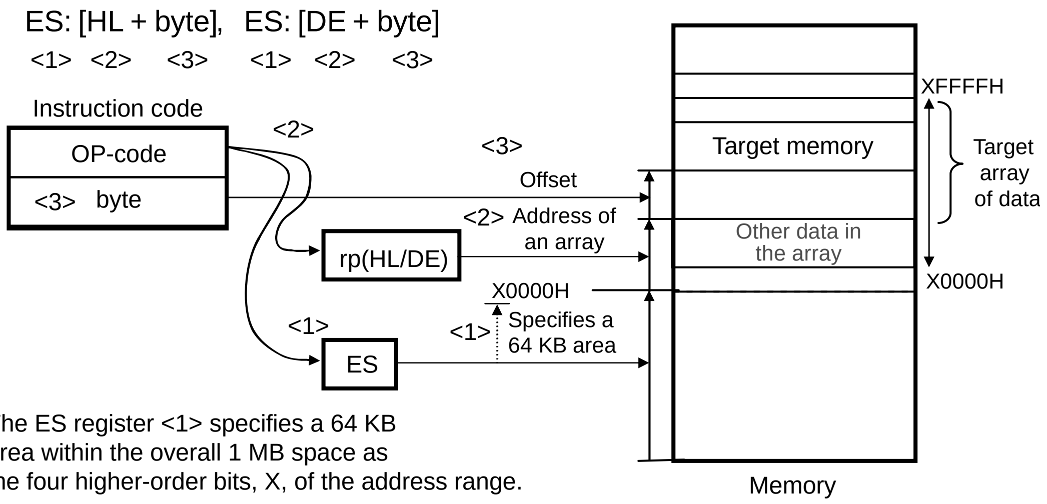
- “word” <1> specifies the address where the target array of word-sized data starts in the 64 KB area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[BC]



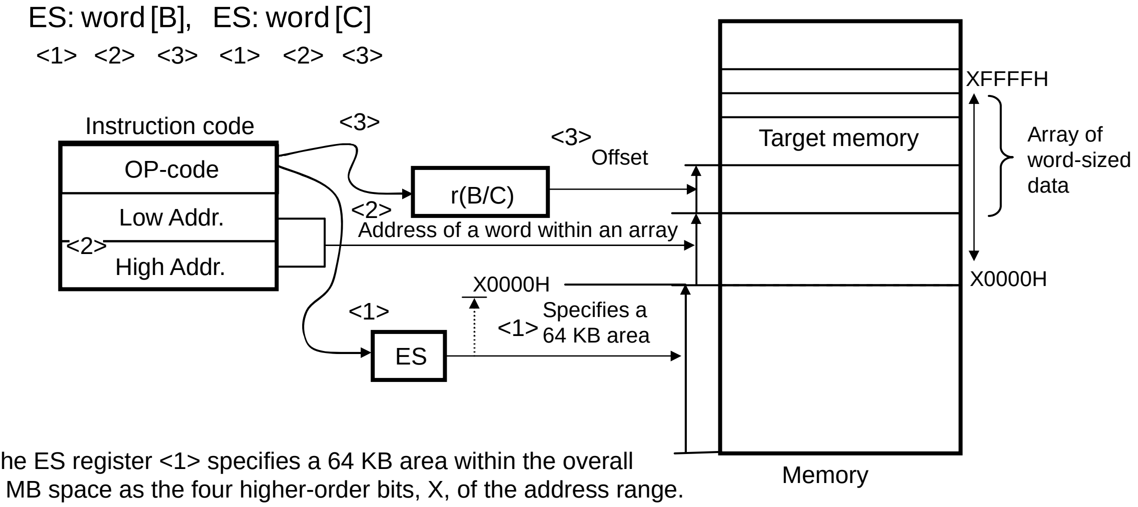
- “word” <1> specifies the address where the target array of word-sized data starts in the 64 KB area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-28. Example of ES:[HL + byte], ES:[DE + byte]



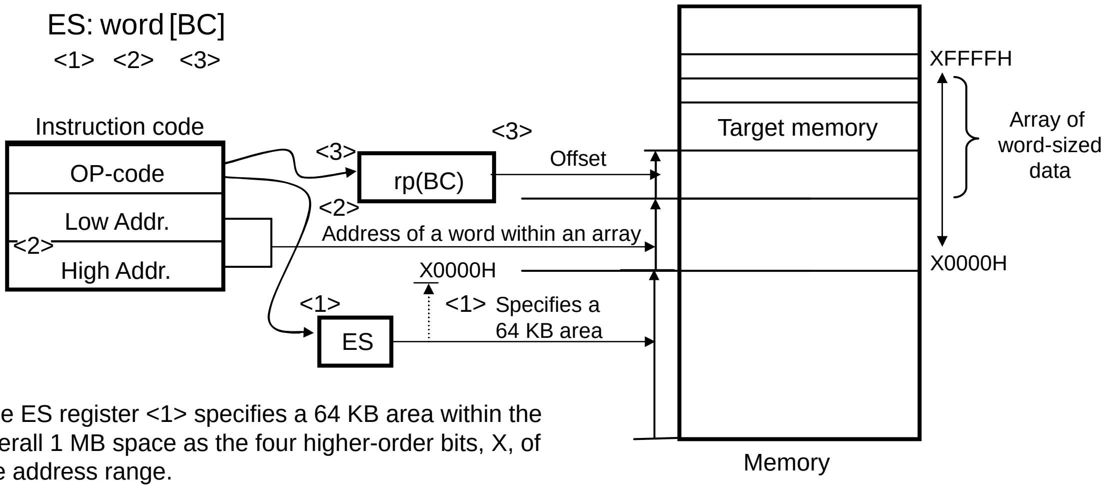
- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64 KB area specified in the ES register <1>.
- “byte” <3> specifies an offset within the array to the target location in memory.

Figure 3-29. Example of ES:word[B], ES:word[C]



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- “word” <2> specifies the address where the target array of word-sizeddata starts in the 64 KB area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3-30. Example of ES:word[BC]



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- “word” <2> specifies the address where the target array of word-sized data starts in the 64 KB area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

3.4.8 Based indexed addressing

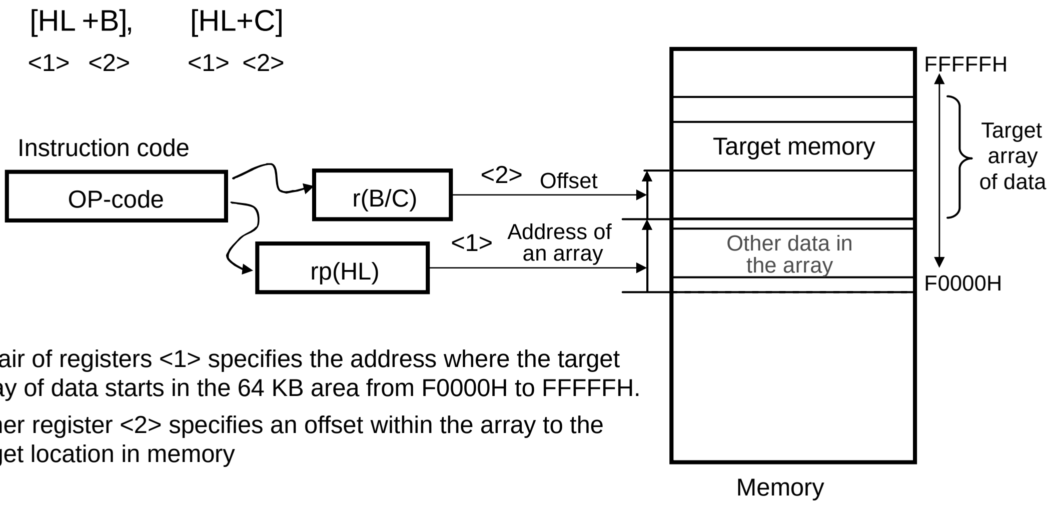
[Function]

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]

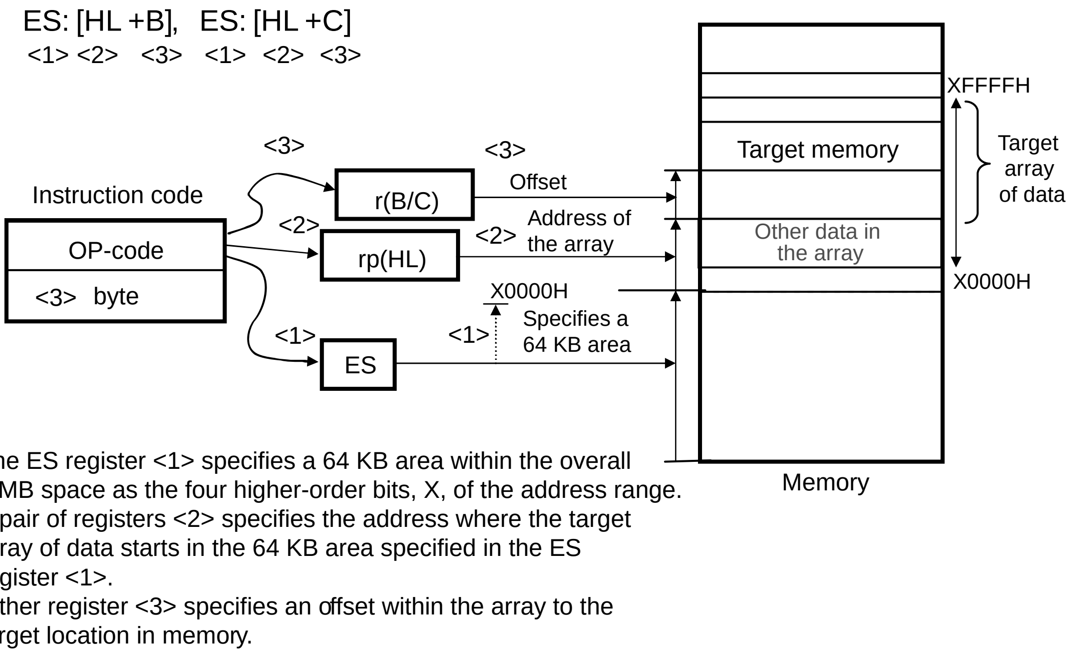
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-31. Example of [HL+B], [HL+C]



- A pair of registers <1> specifies the address where the target array of data starts in the 64 KB area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory

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



- The ES register <1> specifies a 64 KB area within the overall 1 MB space as the four higher-order bits, X, of the address range.
- A pair of registers <2> specifies the address where the target array of data starts in the 64 KB area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

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.

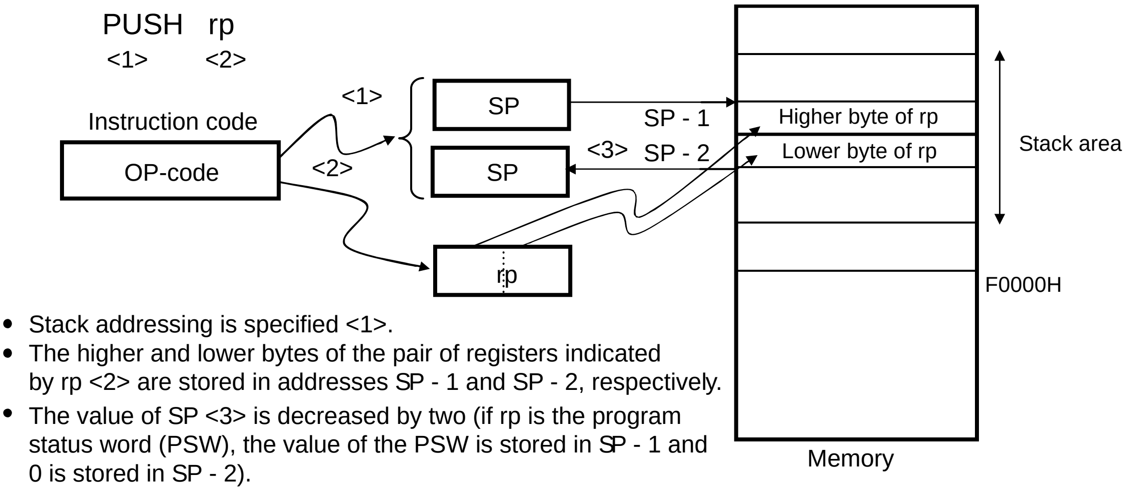
Only the internal RAM area can be set as the stack area.

[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

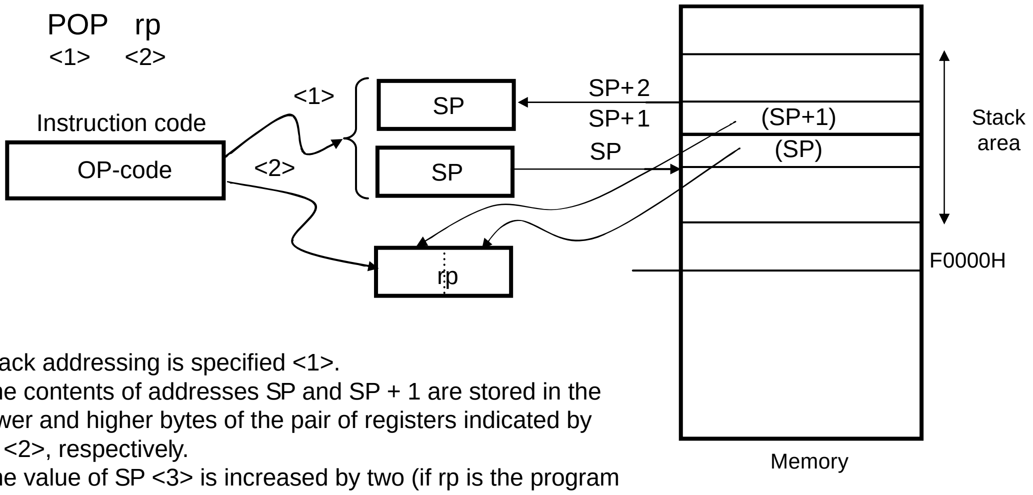
Each stack operation saves or restores data as shown in **Figures 3-33** to **3-38**.

Figure 3-33. Example of PUSH rp



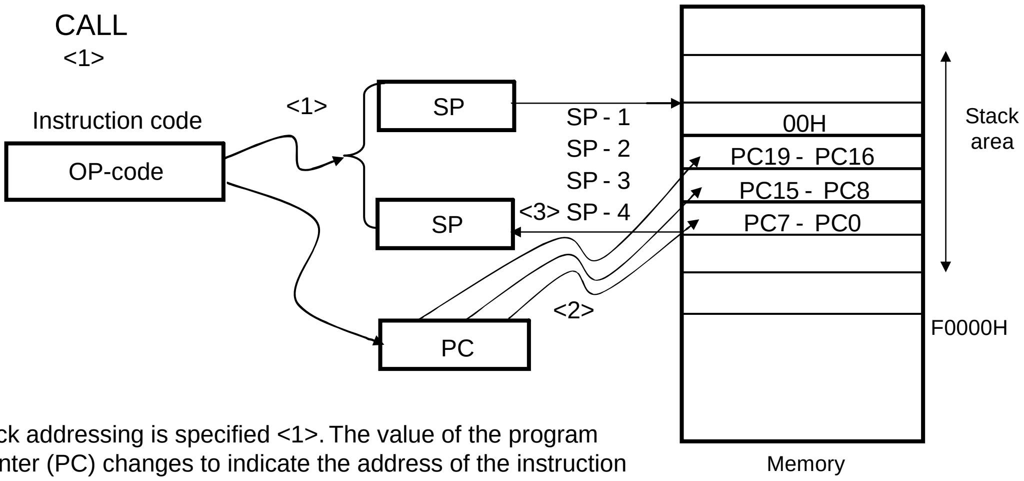
- Stack addressing is specified <1>.
- The higher and lower bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP - 1 and 0 is stored in SP - 2).

Figure 3-34. Example of POP



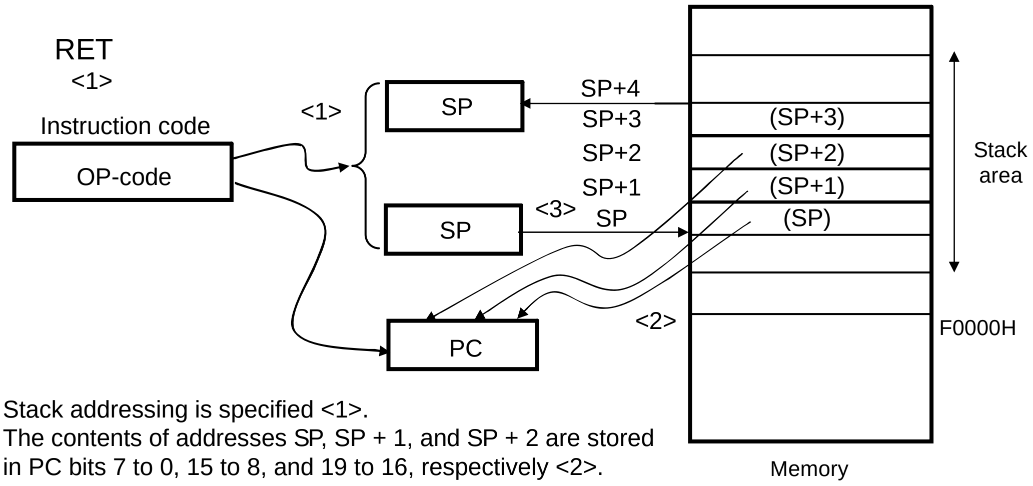
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower and higher bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3-35. Example of CALL, CALLT



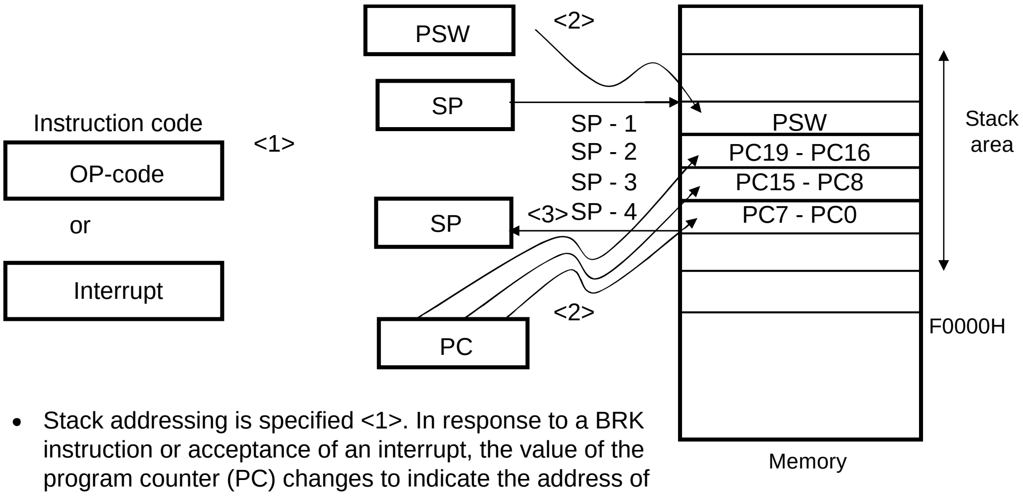
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-36. Example of RET



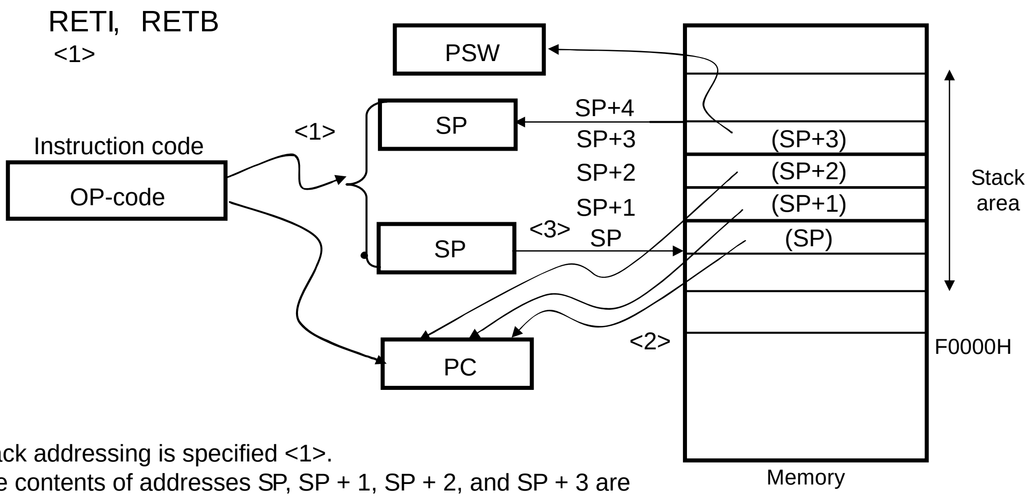
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-37. Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-38. Example of RETI, RETB



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/I1C (512 KB) 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 **CHAPTER 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, PM15) Port registers (P0 to P9, P12, P13, P15) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU8, PU9, PU12) Port input mode registers (PIM0, PIM1, PIM4, PIM5, PIM8) Port output mode registers (POM0, POM1, POM5, POM8) Port mode control register (PMC2) Peripheral I/O redirection register (PIOR0) LCD port function registers (PFSEG0 to PFSEG6) LCD input switch control register (ISCLCD)
Port	● 80-pin product Total: 60 (CMOS I/O: 48 (N-ch open drain I/O [EV_{DD} tolerance]: 13), CMOS input: 5, CMOS output: 1, N-ch open drain I/O [6 V tolerance]: 6) ● 100-pin product Total: 76 (CMOS I/O: 64 (N-ch open drain I/O [EV_{DD} tolerance]: 18), CMOS input: 5, CMOS output: 1, N-ch open drain I/O [6 V tolerance]: 6)
Pull-up resistor	● 80-pin product Total: 45 ● 100-pin product Total: 59

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P02 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P02, P03, P05 and P06 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P02 to P07 pins can be specified as N-ch open-drain output (EV_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

This port can also be used for programming UART transmission/reception, serial interface data I/O, clock I/O, timer I/O, and external interrupt request input. For the 80-pin product, this port can be used for segment output of LCD controller/driver and sampling output.

Reset signal generation sets port 0 to input mode. For the 80-pin product, P02 to P07 pins are set to the digital input invalid mode[Note].

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

4.2.2 Port 1

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

Input to the P15 and P16 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P15 to P17 pins can be specified as N-ch open-drain output (EV_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

This port can also be used for serial interface data I/O, clock I/O, and segment output of LCD controller/driver.

Reset signal generation sets port 1 to the digital input invalid mode[Note].

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

4.2.3 Port 2

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

This port can also be used for analog inputs to the 12-bit A/D converter, positive and negative reference voltage inputs to the 12-bit A/D converter, voltage reference voltage output of the 12-bit A/D converter, and an electric potential input for the detection of low external voltages.

To use P20/ANI3 to P25/ANI2 pins as digital I/O pins, set them in the digital I/O mode by using port mode control register 2 (PMC2).

To use P20/ANI3 to P25/ANI2 pins as analog input pins, set them in the analog input mode by using port mode control register 2 (PMC2) and in the input mode by using the PM2 register.

Reset signal generation sets port 2 to the analog input mode.

4.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for clock/buzzer output, timer I/O, and segment output of LCD controller/driver. For the 80-pin product, this port can be used for sampling input and sampling output.

Reset signal generation sets port 3 to the digital input invalid mode[Note].

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

4.2.5 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40, P42, and P43 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

Input to the P42 and P43 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 4 (PIM4).

This port can also be used for external interrupt request input , clock/buzzer output, timer I/O, data I/O for a flash memory programmer/debugger, and real-time clock correction clock output.

Reset signal generation sets port 4 to input mode.

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Input to the P52, P53, P55, and P57 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).

Output from the P51, P54, P56, and P57 pins can be specified as N-ch open-drain output (EV_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).

This port can also be used for serial interface data I/O, clock I/O, transmission/reception for IrDA, segment output of LCD controller/driver, sampling output, and external interrupt request input.

Reset signal generation sets port 5 to the digital input invalid mode[Note].

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

4.2.7 Port 6

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

The output of the P60, P61, and P62 pins is N-ch open-drain output (6 V tolerance).

This port can also be used for serial interface data I/O, clock I/O, and timer I/O. For the 80-pin product, this port can be used for external interrupt request input.

Reset signal generation sets port 6 to input mode.

4.2.8 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for segment output of LCD controller/driver, key interrupt input, and external interrupt request input. For the 80-pin product, this port can be used for sampling input.

Reset signal generation sets port 7 to the digital input invalid mode[Note].

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

4.2.9 Port 8

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P85 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8 (PU8).

Input to the P80, P81, and P84 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 8 (PIM8).

Output from the P80 to P82, P84, and P85 pins can be specified as N-ch open-drain output (EV_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

This port can also be used for serial interface data I/O, clock I/O, segment output of LCD controller/driver, sampling input, and timer I/O.

Reset signal generation sets port 8 to the digital input invalid mode[Note].

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

4.2.10 Port 9

Port 9 is an I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9). When the P90 to P97 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 9 (PU9).

This port can also be used for segment output of LCD controller/driver, sampling input, and sampling output.

Reset signal generation sets port 9 to the digital input invalid mode[Note].

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

4.2.11 Port 12

P125 to P127 are I/O port pins with output latches. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P125 to P127 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input-only port pins.

This port can also be used for connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, external clock input for subsystem clock, connecting a capacitor for LCD controller/driver, power supply voltage pin for driving the LCD, external interrupt request input, and timer I/O.

Reset signal generation sets P121 to P124 to input mode. P125 to P127 are set in the digital invalid mode[Note].

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

4.2.12 Port 13

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

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

This port can also be used for external interrupt request input. For the 80-pin product, this port can be used for clock/buzzer output and sampling output.

4.2.13 Port 15

P150 to P152 are I/O port pins with output latches. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

<R> Outputs from the P150 to P152 pins are N-ch open-drain (6 V tolerant).

This port can also be used for RTC time capture input, real-time clock correction clock output, and external interrupt request input.

<R> Reset signal generation sets port 15 to input mode.

<R>

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- 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 register (PMCx)
- Peripheral I/O redirection register (PIOR0)
- LCD port function registers (PFSEG0 to PFSEG6)
- LCD input switch control register (ISCLCD)

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

Table 4-2. PMxx, Pxx, PUxx, PIMxx, POMxx Registers and the Bits Mounted on Each Product

Port		Bit Name					80 Pin	100 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register		
Port 0	0	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–
	2	PM02	P02	PU02	PIM02	POM02	√	√
	3	PM03	P03	PU03	PIM03	POM03	√	√
	4	PM04	P04	PU04	–	POM04	√	√
	5	PM05	P05	PU05	PIM05	POM05	√	√
	6	PM06	P06	PU06	PIM06	POM06	√	√
	7	PM07	P07	PU07	–	POM07	√	√
Port 1	0	PM10	P10	PU10	–	–	√	√
	1	PM11	P11	PU11	–	–	√	√
	2	PM12	P12	PU12	–	–	√	√
	3	PM13	P13	PU13	–	–	√	√
	4	PM14	P14	PU14	–	–	√	√
	5	PM15	P15	PU15	PIM15	POM15	√	√
	6	PM16	P16	PU16	PIM16	POM16	√	√
	7	PM17	P17	PU17	–	POM17	√	√

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx Registers and the Bits Mounted on Each Product

Port		Bit Name						80 Pin	100 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register	PMCxx Register		
Port 2	0	PM20	P20	–	–	–	PMC20	√	√
	1	PM21	P21	–	–	–	PMC21	√	√
	2	PM22	P22	–	–	–	PMC22	√	√
	3	PM23	P23	–	–	–	PMC23	√	√
	4	PM24	P24	–	–	–	PMC24	–	√
	5	PM25	P25	–	–	–	PMC25	–	√
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

Table 4-4. PMxx, Pxx, PUxx, PIMxx, POMxx Registers and the Bits Mounted on Each Product (1/2)

Port		Bit Name					80 Pin	100 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register		
Port 3	0	PM30	P30	PU30	–	–	√	√
	1	PM31	P31	PU31	–	–	√	√
	2	PM32	P32	PU32	–	–	√	√
	3	PM33	P33	PU33	–	–	√	√
	4	PM34	P34	PU34	–	–	–	√
	5	PM35	P35	PU35	–	–	–	√
	6	PM36	P36	PU36	–	–	–	√
	7	PM37	P37	PU37	–	–	–	√
Port 4	0	PM40	P40	PU40	–	–	√	√
	1	–	–	–	–	–	–	–
	2	PM42	P42	PU42	PIM42	–	–	√
	3	PM43	P43	PU43	PIM43	–	–	√
	4	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–
Port 5	0	PM50	P50	PU50	–	–	–	√
	1	PM51	P51	PU51	–	POM51	–	√
	2	PM52	P52	PU52	PIM52	–	–	√
	3	PM53	P53	PU53	PIM53	–	–	√
	4	PM54	P54	PU54	–	POM54	–	√
	5	PM55	P55	PU55	PIM55	–	√	√
	6	PM56	P56	PU56	–	POM56	√	√
	7	PM57	P57	PU57	PIM57	POM57	–	√
Port 6	0	PM60	P60	–	–	–	√	√
	1	PM61	P61	–	–	–	√	√
	2	PM62	P62	–	–	–	√	√
	3	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–
Port 7	0	PM70	P70	PU70	–	–	√	√
	1	PM71	P71	PU71	–	–	√	√
	2	PM72	P72	PU72	–	–	√	√
	3	PM73	P73	PU73	–	–	√	√
	4	PM74	P74	PU74	–	–	√	√
	5	PM75	P75	PU75	–	–	√	√
	6	PM76	P76	PU76	–	–	√	√
	7	PM77	P77	PU77	–	–	√	√

Table 4-4. PMxx, Pxx, PUxx, PIMxx, POMxx Registers and the Bits Mounted on Each Product (2/2)

Port		Bit Name					80 Pin	100 Pin
		PMxx Register	Pxx Register	PUxx Register	PIMxx Register	POMxx Register		
Port 8	0	PM80	P80	PU80	PIM80	POM80	√	√
	1	PM81	P81	PU81	PIM81	POM81	√	√
	2	PM82	P82	PU82	–	POM82	√	√
	3	PM83	P83	PU83	–	–	√	√
	4	PM84	P84	PU84	PIM84	POM84	–	√
	5	PM85	P85	PU85	–	POM85	–	√
	6	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–
Port 9	0	PM90	P90	PU90	–	–	√	√
	1	PM91	P91	PU91	–	–	√	√
	2	PM92	P92	PU92	–	–	√	√
	3	PM93	P93	PU93	–	–	√	√
	4	PM94	P94	PU94	–	–	√	√
	5	PM95	P95	PU95	–	–	√	√
	6	PM96	P96	PU96	–	–	√	√
	7	PM97	P97	PU97	–	–	√	√
Port 12	0	–	–	–	–	–	–	–
	1	–	P121	–	–	–	√	√
	2	–	P122	–	–	–	√	√
	3	–	P123	–	–	–	√	√
	4	–	P124	–	–	–	√	√
	5	PM125	P125	PU125	–	–	√	√
	6	PM126	P126	PU126	–	–	√	√
	7	PM127	P127	PU127	–	–	√	√
Port 13	0	–	P130	–	–	–	√	√
	1	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–
	7	–	P137	–	–	–	√	√
Port 15	0	PM150	P150	–	–	–	√	√
	1	PM151	P151	–	–	–	√	√
	2	PM152	P152	–	–	–	√	√
	3	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–

4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

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

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4-1. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	1	1	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	1	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	1	1	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	1	1	1	FFF2CH	FFH	R/W
PM15	1	1	1	1	1	PM152	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 9, 12, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read[Note].

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

Reset signal generation clears these registers to 00H.

Note If P20 to P25 are set up as analog inputs of the 12-bit A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-2. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	0	0	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	0	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	0	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	0	0	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 1]
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Undefined[Note 2]	R/W[Note 1]
P15	0	0	0	0	0	P152	P151	P150	FFF0FH	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

Notes 1. P121 to P124 and P137 are read-only.

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

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

Remark m = 0 to 9, 12, 13, 15 ; n = 0 to 7

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to 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 cannot be connected to bits set to output mode regardless of the settings of these registers.

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

Reset signal generation clears these registers to 00H (PU4 is set to 01H).

Caution When a port with the PIMn register is input from different potential device to TTL buffer, pull up to the power supply of the different potential device via an external pull-up resistor by setting PUmn = 0.

Figure 4-3. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	0	0	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	PU43	PU42	0	PU40	F0034H	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	0	0	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	0	0	0	F003CH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 7 to 9, 12 ; 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 mounted to their initial values.

4.3.4 Port input mode registers (PIMxx)

These registers set the input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

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

Reset signal generation clears these registers to 00H.

Figure 4-4. Format of Port Input Mode Register

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	0	PIM16	PIM15	0	0	0	0	0	F0041H	00H	R/W
PIM4	0	0	0	0	PIM43	PIM42	0	0	F0044H	00H	R/W
PIM5	PIM57	0	PIM55	0	PIM53	PIM52	0	0	F0045H	00H	R/W
PIM8	0	0	0	PIM84	0	0	PIM81	PIM80	F0048H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 4, 5, 8 ; n = 0 to 7)
0	Normal input buffer
1	TTL input buffer

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

4.3.5 Port output mode registers (POMxx)

These registers set the output mode in 1-bit units.

N-ch open drain output (EV_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 and SDA10 pins during simplified I^2C communication with an external device of the same potential.

In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.

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

Reset signal generation clears these registers to 00H.

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

Figure 4-5. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM1	POM17	POM16	POM15	0	0	0	0	0	F0051H	00H	R/W
POM5	POM57	POM56	0	POM54	0	0	POM51	0	F0055H	00H	R/W
POM8	0	0	POM85	POM84	0	POM82	POM81	POM80	F0058H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 5, 8 ; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (EV_{DD} tolerance) mode

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

4.3.6 Port mode control register (PMCxx)

This register sets the digital I/O or analog input in 1-bit units.

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

Reset signal generation clears this register to FFH.

Figure 4-6. Format of Port Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC2	1	1	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W

PMCmn	Selection of digital I/O or analog input for Pmn pin (m = 2; n = 0 to 5)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Cautions 1. Select input mode by using port mode register 2 (PM2) for port pins which are to be set as analog inputs by the PMCxx register.

2. Do not use A/D channel select register A0 (ADANSA0) to set pins which are to be set as digital I/O pins by the PMCxx register.

3. Be sure to set bits corresponding to pins that are not mounted to their initial values.

4.3.7 Peripheral I/O redirection register (PIOR0)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

This function is used to switch ports to which alternate functions are assigned.

Use the PIOR0 register to assign a port to the function to redirect and enable the function.

In addition, can be changed the settings for redirection until its function enable operation.

The PIOR0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 4-7. Format of Peripheral I/O Redirection Register (PIOR0)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR0	0	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

Bit	Function	100-pin		80-pin	
		Setting value		Setting value	
		0	1	0	1
PIOR06	INTP8	P122	P55	P122	P55
	INTP9	P121	P53	P121	P06
PIOR05	COM0	P90	P37	These functions are not available for use. Set this bit to 0 (default value).	
	COM1	P91	P36		
	COM2	P92	P35		
	COM3	P93	P34		
	COM4/SEG0	P94	P33		
	COM5/SEG1	P95	P32		
	COM6/SEG2	P96	P31		
	COM7/SEG3	P97	P30		
	SEG24	P30	P97		
	SEG25	P31	P96		
	SEG26	P32	P95		
	SEG27	P33	P94		
	SEG28	P34	P93		
	SEG29	P35	P92		
	SEG30	P36	P91		
	SEG31	P37	P90		
PIOR04	INTP0	P137	P70	P137	P70
	INTP1	P125	P71	P125	P71
	INTP2	P07	P72	P07	P72
	INTP3	P05	P73	P05	P73
	INTP4	P04	P74	P04	P74
	INTP5	P02	P75	P02	P75
	INTP6	P42	P76	P62	P76
	INTP7	P43	P77	–	P77
PIOR03	PCLBUZ0	P43	P33	–	P33
	PCLBUZ1	P42	P32	P130	P32
	RTCOUT	P150	P40	P150	P40
PIOR02	SO10/TXD1	P04	P82	P04	P82
	SI10/RXD1/SDA10	P03	P81	P03	P81
	SCK10/SCL10	P02	P80	P02	P80
PIOR01	SO00/TXD0	P07	P17	P07	P17
	SI00/RXD0/SDA00	P06	P16	P06	P16
	SCK00/SCL00	P05	P15	P05	P15
PIOR00	TI00/TO00	P43	P60	–	P60
	TI01/TO01	P42	P61	P33	P61
	TI02/TO02	P07	P62	P07	P62
	TI03/TO03	P06	P127	P06	P127
	TI04/TO04	P05	P126	P05	P126
	TI05/TO05	P04	P125	P04	P125
	TI06/TO06	P03	P31	P03	P31
	TI07/TO07	P02	P30	P02	P30

4.3.8 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers set whether to use pins P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, and P90 to P97 as port pins (other than as segment and common output pins) or as segment and common output pins.

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

Reset signal generation sets these registers to FFH (PFSEG0 is set to 0FH, and PFSEG6 is set to 03H).

Remark The correspondence between the common output pins (COMx) and the PFSEG register (PFCOMx bits) and the existence of COMx pins in each product are shown in **Table 4-5 Common Output Pins in Each Product and Correspondence with PFSEG Register (PFCOM Bits)**.

The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in **Table 4-6 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**.

Figure 4-8. Format of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

Address: F0300H After reset: 0FH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	0	0	0	0	PFCOM3	PFCOM2	PFCOM1	PFCOM0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG07	PFSEG06	PFSEG05	PFSEG04	PFSEG03	PFSEG02	PFSEG01	PFSEG00

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0304H After reset: FFH (R5F10NPL), 0FH (R5F10NML) R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG31 Notes 1, 2	PFSEG30 Notes 1, 2	PFSEG29 Notes 1, 2	PFSEG28 Notes 1, 2	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0305H After reset: FFH (R5F10NPL), 3FH (R5F10NML) R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG39 Notes 1, 2	PFSEG38 Notes 1, 2	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0306H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6 Note 3	0	0	0	0	0	0	PFSEG41	PFSEG40

PFCOMx (x = 0 to 3)	Selection of Pmn pins to operate as port pins (other than as common outputs) or as common outputs (mn = 90 to 93)
0	Use the Pmn pin as a port pin (other than as a common output).
1	Use the Pmn pin as a common output.

PFSEGxx (xx = 00 to 41)	Selection of Pmn pins to operate as port pins (other than as segment and common outputs) or as segment and common outputs (mn = 02 to 07, 10 to 17, 30 to 37, 50 to 57, 70 to 77, 80 to 85, 94 to 97)
0	Use the Pmn pin as a port pin (other than as a segment or common output).
1	Use the Pmn pin as a segment or common output.

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

Notes 1. For the R5F10NML, the initial value is 0.
Writing 1 to this bit does not affect operation, and the value read is 0.

2. Be sure to set "1" for the 80-pin product.

3. R5F10NPL only

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

Remark To use the Pmn pins as segment or common output pins (PFSEGxx = 1 or PFCOMx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to “0”.

Table 4-5. Common Output Pins in Each Product and Correspondence with PFSEG Register (PFCOM Bits)

Bit Name of PFSEG Register	Corresponding COMx Pins	Alternate Port	100-pin	80-pin
PFCOM0	COM0	P90, (P37)	√	–
		P90	–	√
PFCOM1	COM1	P91, (P36)	√	–
		P91	–	√
PFCOM2	COM2	P92, (P35)	√	–
		P92	–	√
PFCOM3	COM3	P93, (P34)	√	–
		P93	–	√

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

Table 4-6. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits) (1/2)

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	100-pin	80-pin
PFSEG00	SEG0	P94, (P33)	√	–
		P94	–	√
PFSEG01	SEG1	P95, (P32)	√	–
		P95	–	√
PFSEG02	SEG2	P96, (P31)	√	–
		P96	–	√
PFSEG03	SEG3	P97, (P30)	√	–
		P97	–	√
PFSEG04	SEG4	P10	√	√
PFSEG05	SEG5	P11	√	√
PFSEG06	SEG6	P12	√	√
PFSEG07	SEG7	P13	√	√
PFSEG08	SEG8	P14	√	√
PFSEG09	SEG9	P15	√	√
PFSEG10	SEG10	P16	√	√
PFSEG11	SEG11	P17	√	√
PFSEG12	SEG12	P80	√	√
PFSEG13	SEG13	P81	√	√
PFSEG14	SEG14	P82	√	√
PFSEG15	SEG15	P83	√	√
PFSEG16	SEG16	P70	√	√
PFSEG17	SEG17	P71	√	√
PFSEG18	SEG18	P72	√	√
PFSEG19	SEG19	P73	√	√
PFSEG20	SEG20	P74	√	√
PFSEG21	SEG21	P75	√	√
PFSEG22	SEG22	P76	√	√
PFSEG23	SEG23	P77	√	√
PFSEG24	SEG24	P30, (P97)	√	–
		P30	–	√
PFSEG25	SEG25	P31, (P96)	√	–
		P31	–	√
PFSEG26	SEG26	P32, (P95)	√	–
		P32	–	√
PFSEG27	SEG27	P33, (P94)	√	–
		P33	–	√
PFSEG28	SEG28	P34, (P93)	√	–
PFSEG29	SEG29	P35, (P92)	√	–
PFSEG30	SEG30	P36, (P91)	√	–
PFSEG31	SEG31	P37, (P90)	√	–

(**Remark** is listed on the next page.)

Table 4-6. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits) (2/2)

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	100-pin	80-pin
PFSEG32	SEG32	P50	√	–
		P02	–	√
PFSEG33	SEG33	P51	√	–
		P03	–	√
PFSEG34	SEG34	P52	√	–
		P04	–	√
PFSEG35	SEG35	P53	√	–
		P05	–	√
PFSEG36	SEG36	P54	√	–
		P06	–	√
PFSEG37	SEG37	P55	√	–
		P07	–	√
PFSEG38	SEG38	P56	√	–
PFSEG39	SEG39	P57	√	–
PFSEG40	SEG40	P84	√	–
PFSEG41	SEG41	P85	√	–

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

4.3.9 LCD input switch control register (ISCLCD)

This register sets whether to use pins P125 to P127 as port pins (other than LCD function pins) or LCD function pins (V_{L3}, CAPL, CAPH).

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

Reset signal generation sets this register to 00H.

Figure 4-9. Format of LCD input switch control register (ISCLCD)

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	Control of schmitt trigger buffer of V_{L3}/P125 pin
0	Makes digital input invalid (used as LCD function pin (V_{L3}))
1	Makes digital input valid

ISCCAP	Control of schmitt trigger buffer of CAPL/ P126 and CAPH/P127 pins
0	Makes digital input invalid (used as LCD function pins (CAPL,CAPH))
1	Makes digital input valid

Caution If ISCVL3 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

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, 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 status 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 status 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 its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status 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 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V or 3 V) by switching I/O buffers with port input mode registers 0, 1, 4, 5, and 8 (PIM0, PIM1, PIM4, PIM5, and PIM8) and port output mode registers 0, 1, 5, and 8 (POM0, POM1, POM5, and POM8).

When receiving input from an external device with a different potential (1.8 V, 2.5 V or 3 V), set port input mode registers 0, 1, 4, 5, and 8 (PIM0, PIM1, PIM4, PIM5, and PIM8) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.

When outputting data to an external device with a different potential (1.8 V, 2.5 V or 3 V), set port output mode registers 0, 1, 5, and 8 (POM0, POM1, POM5, and POM8) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (EV_{DD} tolerance) switching.

The connection of a serial interface is described in the following.

(1) Setting procedure when using input pins of UART0 to UART4, CSI00, CSI10, and CSI30 functions for the TTL input buffer

In case of UART0:	P06 (P16)
In case of UART1:	P03 (P81)
In case of UART2:	P55
In case of UART3:	P84
In case of UART4:	P52
In case of CSI00:	P05, P06 (P15, P16)
In case of CSI10:	P02, P03 (P80, P81)
In case of CSI30:	P57, P84

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR0).

<1> Using an external resistor, pull up externally the 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, PIM1, PIM4, PIM5, and PIM8 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) Setting procedure when using output pins of UART0 to UART4, CSI00, CSI10, and CSI30 functions in N-ch open-drain output mode

In case of UART0:	P07 (P17)
In case of UART1:	P04 (P82)
In case of UART2:	P56
In case of UART3:	P85
In case of UART4:	P51
In case of CSI00:	P05, P07 (P15, P17)
In case of CSI10:	P02, P04 (P80, P82)
In case of CSI30:	P57, P85

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR0).

<1> Using an external resistor, pull up externally the 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 port mode changes to the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM1, POM5, and POM8 registers to 1 to set the N-ch open drain output (EV_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.
<6> Set the output mode by manipulating the PM0, PM1, PM5, and PM8 registers.
At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O pins of IIC00, IIC10, and IIC30 functions with a different potential (1.8 V, 2.5 V, 3 V)

In case of CSI00:	P05, P06 (P15, P16)
In case of CSI10:	P02, P03 (P80, P81)
In case of CSI30:	P57, P84

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR0).

<1> Using an external resistor, pull up externally the 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 port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM1, POM5,and POM8 registers to 1 to set the N-ch open drain output (EV_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM0, PIM1, PIM5, and PIM8 registers to 1 to switch 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 I^{2}C mode.
<7> Set the corresponding bit of the PM0, PM1, PM5, and PM8 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.

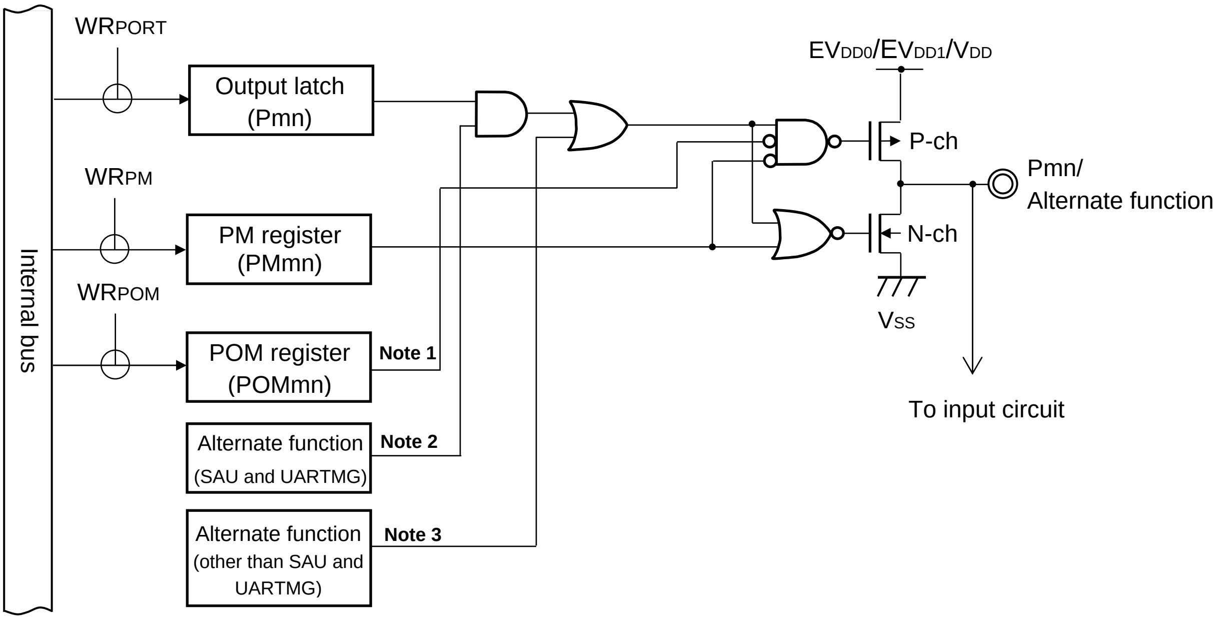
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog input, use the port mode control register (PMCxx) to specify whether to use the pin for analog input or digital input/output.

Figure 4-10 shows the basic configuration of an output circuit for pins used for digital input/output. The output from the output latch for the port-pin function and the output from the multiplexed SAU and UARTMG 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 SAU and UARTMG (timer array unit, realtime clock with independent power supply, clock/buzzer output, IICA, etc.) are connected to the other input pins of the OR gate. When such kind of pins are used by 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 4-7**.

Figure 4-10. Basic Configuration of Output Circuit for Pins



Notes 1. When there is no POM register, this signal should be considered to be low level (0).
2. When there is no alternate function, this signal should be considered to be high level (1).
3. When there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 0 to 9, 12, 13, 15); n: Bit number (n = 0 to 7)

Table 4-7. Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Port Function	Output Function for SAU and UARTMG	Output Function for other than SAU and UARTMG
Output function for port	–	Output is high (1)	Output is low (0)
Output function for SAU and UARTMG	High (1)	–	Output is low (0)
Output function for other than SAU and UARTMG	Low (0)	Output is high (1)	Output is low (0)[Note]

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

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

When the output of an alternate function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is the target, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR0). This allows usage of the port function or other alternate function assigned to the target pin.

(1) SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
When the serial output (SOp/TxDq) is not used, such as, a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(2) SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high). 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 bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

(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) TxDMGn = 1 (setting when UARTMGn output is not used)
When the UARTMGn is not to be used, set bit 7 to 0 (disabling operation of the UART operation clock) and bit 6 to 0 (disabling transmission) in UARTMGn operating mode setting register n0 (ASIMMGn0), and bit 0 in UARTMGn operating mode setting register n1 (ASIMMGn1) to 0 (positive logic). This is the same setting as the initial state.

(7) TRJIOn = 0/TRJOn = 0 (setting when timer RJn output is not used)
When TRJOn pin is not used for the pulse output function of timer RJ, set bit 2 (TOENAn) in the timer RJ I/O control register n (TRJIOCn) to 0 (disabling TRJOn output). This is the same setting as the initial state.
When TRJIOn pin of timer RJ is not used for the output function, set bits 2 to 0 (TMODn2 to TMODn0) in the timer RJ mode register n (TRJMRn) to the value other than 001b (pulse output mode). The initial setting is 000b (timer mode).

(8) SMOmj = 0 (setting when sampling clock output is not used)
When sampling clock output (SMOmj) is not used, set bit j (SMOTDOEmj) in the SMOTD output control register m (SMOTDOEm) to 0 (disabling SMOn output). This is the same setting as the initial state.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in **Table 4-8**. The registers used to control the port functions should be set as shown in **Table 4-8**. See the following remark for legends used in **Table 4-8**.

Remark	–:	Not supported
	×:	don’t care
	PIOR0:	Peripheral I/O redirection register
	POMxx:	Port output mode register
	PMCxx:	Port mode control register
	PMxx:	Port mode register
	Pxx:	Port output latch
	PUxx:	Pull-up resistor option register
	PIMcc:	Port input mode register
	PFSEG××:	LCD port function register
	ISCLCD:	LCD input switch control register

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (1/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)[Note]	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P02	P02	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	SCL10 = 1	TO07 = 0 SMO12 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	SCL10	Output	PIOR02 = 0	0/1	0	1	0	–	TO07 = 0 SMO12 = 0		
	SCK10	Input	PIOR02 = 0	×	1	×	0	–	–		
		Output	PIOR02 = 0	0/1	0	1	0	–	TO07 = 0 SMO12 = 0		
	TI07	Input	PIOR00 = 0	×	1	×	0	–	–		
	TO07	Output	PIOR00 = 0	0	0	0	0	SCL10 = 1	SMO12 = 0		
	INTP5	Input	PIOR04 = 0	×	1	×	0	–	–		
	SEG32	Output	×	0	0	0	1	–	–	√	–
	SMO12	Output	–	0	0	0	0	SCL10 = 1	TO07 = 0		
P03	P03	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	SDA10 = 1	TO06 = 0 TRJO1 = 0 SMO11 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	RxD1	Input	PIOR02 = 0	×	1	×	0	–	–		
	TI06	Input	PIOR00 = 0	×	1	×	0	–	–		
	TO06	Output	PIOR00 = 0	0	0	0	0	SDA10 = 1	–		
	SDA10	I/O	PIOR02 = 0	1	0	1	0	–	TO06 = 0 TRJO1 = 0 SMO11 = 0		
	SI10	Input	PIOR02 = 0	×	1	×	0	–	–		
	SEG33	Output	×	0	0	0	1	–	–	√	–
	TRJO1	Output	–	0	0	0	0	SDA10 = 1	TO06 = 0 SMO11 = 0		
	SMO11	Output	–	0	0	0	0	SDA10 = 1	TO06 = 0 TRJO1 = 0		
P04	P04	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	TxD1/SO10 = 1	TO05 = 0 TRJO0 = 0 SMO10 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	TxD1	Output	PIOR02 = 0	0/1	0	1	0	–	TO05 = 0 TRJO0 = 0 SMO10 = 0		
	TI05	Input	PIOR00 = 0	×	1	×	0	TxD1/SO10 = 1	–		
	TO05	Output	PIOR00 = 0	0	0	0	0	TxD1 = 1	TRJO0 = 0 SMO10 = 0		
	INTP4	Input	PIOR04 = 0	×	1	×	0	–	–		
	SO10	Output	PIOR02 = 0	0/1	0	1	0	–	TO05 = 0 TRJO0 = 0 SMO10 = 0		
	SEG34	Output	×	0	0	0	1	–	–	√	–
	TRJO0	Output	–	0	0	0	0	TxD1/SO10 = 1	TO05 = 0 SMO10 = 0		
	SMO10	Output	–	0	0	0	0	TxD1/SO10 = 1	TO05 = 0 TRJO0 = 0		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (2/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P05	P05	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	SCK00/SCL00 = 1	TO04 = 0 TRJIO0 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	SCK00	Input	PIOR01 = 0	×	1	×	0	–	–		
		Output		0/1	0	1	0	–	TO04 = 0 TRJIO0 = 0		
	SCL00	Output	PIOR01 = 0	0/1	0	1	0	–	TO04 = 0 TRJIO0 = 0		
	TI04	Input	PIOR00 = 0	×	1	×	0	–	–		
	TO04	Output	PIOR00 = 0	0	0	0	0	SCK00/SCL00 = 1	–		
	INTP3	Input	PIOR04 = 0	×	1	×	0	–	–		
	SEG35	Output	×	0	0	0	1	–	–	√	–
	TRJIO0	Input	–	0	1	×	0	–	–		
		Output	–	0	0	0	0	SCK00/SCL00 = 1	TO04 = 0		
P06	P06	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	SDA00 = 1	TO03 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	SI00	Input	PIOR01 = 0	×	1	×	0	–	–		
	RxD0	Input	PIOR01 = 0	×	1	×	0	–	–		
	TI03	Input	PIOR00 = 0	×	1	×	0	–	–		
	TO03	Output	PIOR00 = 0	0	0	0	0	SDA00 = 1	–		
	SDA00	I/O	PIOR01 = 0	1	0	1	0	–	TO03 = 0		
	TOOLRxD	Input	×	×	1	×	0	–	–		
	SEG36	Output	×	0	0	0	1	–	–	√	–
	RxDMG1	Input	–	–	1	×	–	–	–		
	(INTP9)	Input	PIOR06 = 1	–	1	×	0	–	–		
P07	P07	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	SO00/TxD0/TxDMG1 = 1	TO02 = 0		
		N-ch open drain output	×	1	0	0/1	0				
	SO00	Output	PIOR01 = 0	0/1	0	1	0	–	TO02 = 0		
	TxD0	Output	PIOR01 = 0	0/1	0	1	0	–	TO02 = 0		
	TI02	Input	PIOR00 = 0	×	1	×	0	–	–		
	TO02	Output	PIOR00 = 0	0	0	0	0	SO00/TxD0/TxDMG1 = 1	–		
	INTP2	Input	PIOR04 = 0	×	1	×	0	–	–		
	TOOLTxD	Output	×	0/1	0	1	0	–	–		
	SEG37	Output	×	0	0	0	1	–	–	√	–
	TxDMG1	Output	–	0/1	0	1	0	–	TO02 = 0		
P10	P10	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG4	Output	–	–	0	0	1	–	–		
P11	P11	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG5	Output	–	–	0	0	1	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (3/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)[Note]	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P12	P12	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	SCK10/SCL10 = 1	–		
	SEG6	Output	×	–	0	0	1	SCK10/SCL10 = 1	–		
P13	P13	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	SDA10 = 1	–		
	SEG7	Output	×	–	0	0	1	–	–		
P14	P14	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	SO10/TxD1 = 1	–		
	SEG8	Output	×	–	0	0	1	–	–		
P15	P15	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(SCK00/SCL00) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG9	Output	×	0	0	0	1	–	–		
	(SCK00)	Input	PIOR01 = 1	×	1	×	0	–	–		
		Output	PIOR01 = 1	0/1	0	1	0	–	–		
	(SCL00)	Output	PIOR01 = 1	0/1	0	1	0	–	–		
P16	P16	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(SDA00) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG10	Output	×	0	0	0	1	–	–		
	(SI00)	Input	PIOR01 = 1	×	1	×	0	–	–		
	(RxD0)	Input	PIOR01 = 1	×	1	×	0	–	–		
	(SDA00)	I/O	PIOR01 = 1	1	0	0	0	–	–		
P17	P17	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(SO00/TxD0) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG11	Output	×	0	0	0	1	–	–		
	(SO00)	Output	PIOR01 = 1	0/1	0	1	0	–	–		
	(TxD0)	Output	PIOR01 = 1	0/1	0	1	0	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (4/12)

Pin Name	Used Function		PMCxx	PMxx	Pxx	80-pin	100-pin
	Function Name	I/O					
P20	P20	Input	0	1	×	√	√
		Output	0	0	0/1		
	ANI3	Analog input	1	1	×		
	AV$_{\mathrm{REFP}}$	Reference voltage input	1	1	×		
	VREFOUT	Analog output					
P21	P21	Input	0	1	×	√	√
		Output	0	0	0/1		
	ANI4	Analog input	1	1	×		
	AV$_{\mathrm{REFM}}$	Reference voltage input	1	1	×		
P22	P22	Input	0	1	×	√	√
		Output	0	0	0/1		
	ANI5	Analog input	1	1	×		
	EXLVD	Analog input	1	1	×		
P23	P23	Input	0	1	×	√	√
		Output	0	0	0/1		
	ANI0	Analog input	1	1	×		
P24	P24	Input	0	1	×	–	√
		Output	0	0	0/1		
	ANI1	Analog input	1	1	×		
P25	P25	Input	0	1	×	–	√
		Output	0	0	0/1		
	ANI2	Analog input	1	1	×		

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (5/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P30	P30	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	(TO07) = 0		
	SEG24	Output	×	–	0	0	1	–	–		
	(TI07)	Input	PIOR00 = 1	–	1	×	0	–	–		
	(TO07)	Output	PIOR00 = 1	–	0	0	0	–	–		
	(COM7)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	(SEG3)	Output	PIOR05 = 1	–	0	0	1	–	–		
	SMP3	Input	–	–	1	×	0	–	–	√	–
P31	P31	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	TxD2/IrTxD = 1	(TO06) = 0		
	SEG25	Output	×	–	0	0	1	–	–		
	(TI06)	Input	PIOR00 = 1	–	1	×	0	–	–		
	(TO06)	Output	PIOR00 = 1	–	0	0	0	–	–		
	(COM6)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	(SEG2)	Output	PIOR05 = 1	–	0	0	1	–	–		
	SMP2	Input	–	–	1	×	0	–	–	√	–
P32	P32	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	(PCLBUZ1) = 0 SMO01 = 0		
	SEG26	Output	×	–	0	0	1	–	–		
	(PCLBUZ1)	Output	PIOR03 = 1	–	0	0	0	–	–		
	(COM5)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	(SEG1)	Output	PIOR05 = 1	–	0	0	1	–	–		
	SMP1	Input	–	–	1	×	0	–	–	√	–
	SMO01	Output	–	–	0	0	0	–	(PCLBUZ1) = 0		
P33	P33	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	(PCLBUZ0) = 0 SMO00 = 0 TO01 = 0		
	SEG27	Output	×	–	0	0	1	–	–		
	(PCLBUZ0)	Output	PIOR03 = 1	–	0	0	0	–	–		
	(COM4)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	(SEG0)	Output	PIOR05 = 1	–	0	0	1	–	–		
	SMP0	Input	–	–	1	×	0	–	–	√	–
	SMO00	Output	–	–	0	0	0	–	(PCLBUZ0) = 0 TO01 = 0		
	TI01	Input	PIOR00 = 0	–	1	×	–	–	–		
	TO01	Output	PIOR00 = 0	–	0	0	–	–	(PCLBUZ0) = 0 SMO00 = 0		
P34	P34	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	–		
	SEG28	Output	–	–	0	0	1	–	–		
	(COM3)	Output	PIOR05 = 1	–	0	0	1	–	–		
P35	P35	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	–		
	SEG29	Output	–	–	0	0	1	–	–		
	(COM2)	Output	PIOR05 = 1	–	0	0	1	–	–		
P36	P36	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	–		
	SEG30	Output	–	–	0	0	1	–	–		
	(COM1)	Output	PIOR05 = 1	–	0	0	1	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (6/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)[Note]	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P37	P37	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	–		
	SEG31	Output	–	–	0	0	1	–	–		
	(COM0)	Output	PIOR05 = 1	–	0	0	1	–	–		
P40	P40	Input	–	–	1	×	–	–	–	√	√
		Output	–	–	0	0/1	–	–	–		
	TOOL0	I/O	–	–	×	×	–	–	–		
	(RTCOUT)	Output	PIOR03 = 1	–	0	0	–	–	–		
P42	P42	Input	–	–	1	×	–	–	–	–	√
		Output	×	–	0	0/1	–	–	TO01 = 0 PCLBUZ1 = 0		
	TI01	Input	PIOR00 = 0	–	1	×	–	–	–		
	TO01	Output	PIOR00 = 0	–	0	0	–	–	PCLBUZ1 = 0		
	PCLBUZ1	Output	PIOR03 = 0	–	0	0	–	–	TO01 = 0		
	INTP6	Input	PIOR04 = 0	–	1	×	–	–	–		
P43	P43	Input	–	–	1	×	–	–	–	–	√
		Output	×	–	0	0/1	–	–	TO00 = 0 PCLBUZ0 = 0		
	TI00	Input	PIOR00 = 0	–	1	×	–	–	–		
	TO00	Output	PIOR00 = 0	–	0	0	–	–	PCLBUZ0 = 0		
	PCLBUZ0	Output	PIOR03 = 0	–	0	0	–	–	TO00 = 0		
	INTP7	Input	PIOR04 = 0	–	1	×	–	–	–		
P50	P50	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	SMO00 = 0		
	SEG32	Output	–	–	0	0	1	–	–		
	SMO00	Output	–	–	0	0	0	–	–		
P51	P51	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	TxD4 = 1	SMO01 = 0		
		N-ch open drain output	–	1	0	0/1	0				
	SEG33	Output	–	–	0	0	1	–	–		
	SMO01	Output	–	–	0	0	0	TxD4 = 1	–		
	TxD4	Output	–	0/1	0	1	0	–	SMO01 = 0		
P52	P52	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	SMO02 = 0		
	SEG34	Output	–	–	0	0	1	–	–		
	SMO02	Output	–	–	0	0	0	–	–		
	RxD4	Input	–	×	1	×	0	–	–		
P53	P53	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	–	TRJO0 = 0		
	SEG35	Output	–	–	0	0	1	–	–		
	TRJO0	Output	–	×	1	×	0	–	–		
	RxDMG1	Input	–	×	1	×	0	–	–		
	(INTP9)	Input	PIOR06 = 1	–	1	×	–	–	–		
P54	P54	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	TxDMG1 = 0	TRJO1 = 0		
	SEG36	Output	–	–	0	0	1	–	–		
	TRJO1	Output	–	×	1	×	0	TxDMG1 = 0	–		
	TxDMG1	Output	–	0/1	0	1	0	–	TRJO1 = 0		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (7/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)[Note]	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P55	P55	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	RxD2	Input	–	–	1	×	0	–	–		
	IrRxD	Input	–	–	1	×	0	–	–		
	SEG37	Output	–	–	0	0	1	–	–	–	
	RxDMG0	Input	–	×	1	×	0	–	–	√	
	(INTP8)	Input	PIOR06 = 1	–	1	×	–	–	–		
P56	P56	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	TxD2/IrTxD/ TxDMG0 = 1	–		
		N-ch open drain output	–	1	0	0/1	0				
	TxD2	Output	–	0/1	0	1	0	–	–		
	IrTxD	Output	–	0/1	0	1	0	–	–		
	SEG38	Output	–	–	0	0	1	–	–	–	
	TxDMG0	Output	–	0/1	0	1	0	–	–	√	
P57	P57	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	SCK30/SCL30 = 1	TRJIO0 = 0		
		N-ch open drain output	–	1	0	0/1	0				
	SCK30	Input	–	×	1	×	0	–	–		
		Output	–	0/1	0	1	0	–	–		
	SCL30	Output	–	0/1	0	1	0	–	–		
	SEG39	Output	–	–	0	0	1	–	–		
	TRJIO0	Input	–	0	1	×	0	–	–		
		Output	–	0	0	0	0	SCK30/SCL30 = 1	–		
P60	P60	Input	–	–	1	×	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	×	–	0	0/1	–	–	SCLA0 = 0 (TO00) = 0		
	SCLA0	I/O	×	–	0	0	–	–	(TO00) = 0		
	(TI00)	Input	PIOR00 = 1	–	1	×	–	–	–		
	(TO00)	Output	PIOR00 = 1	–	0	0	–	–	SCLA0 = 0		
P61	P61	Input	–	–	1	×	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	×	–	0	0/1	–	–	SDAA0 = 0 (TO01) = 0		
	SDAA0	I/O	×	–	0	0	–	–	(TO01) = 0		
	(TI01)	Input	PIOR00 = 1	–	1	×	–	–	–		
	(TO01)	Output	PIOR00 = 1	–	0	0	–	–	SDAA0 = 0		
P62	P62	Input	–	–	1	×	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	×	–	0	0/1	–	–	(TO02) = 0 TRJIO1 = 0		
	(TI02)	Input	PIOR00 = 1	–	1	×	–	–	–		
	(TO02)	Output	PIOR00 = 1	–	0	0	–	–	TRJIO1 = 0		
	TRJIO1	Input	–	0	1	×	0	–	–	√	–
		Output	–	0	0	0	0	–	(TO02) = 0		
	INTP6	Input	PIOR04 = 0	–	1	×	0	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (8/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P70	P70	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	SEG16	Output	×	–	0	0	1	–	–		
	KR0	Input	×	–	1	×	0	–	–		
	(INTP0)	Input	PIOR04 = 1	–	1	×	0	–	–		
P71	P71	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	SEG17	Output	×	–	0	0	1	–	–		
	KR1	Input	×	–	1	×	0	–	–		
	(INTP1)	Input	PIOR04 = 1	–	1	×	0	–	–		
P72	P72	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	KR2	Input	×	–	1	×	0	–	–		
	SEG18	Output	×	–	0	0	1	–	–		
	(INTP2)	Input	PIOR04 = 1	–	1	×	0	–	–		
P73	P73	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	KR3	Input	×	–	1	×	0	–	–		
	SEG19	Output	×	–	0	0	1	–	–		
	(INTP3)	Input	PIOR04 = 1	–	1	×	0	–	–		
P74	P74	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	(PCLBUZ0) = 0		
	KR4	Input	×	–	1	×	0	–	–		
	SEG20	Output	×	–	0	0	1	–	–		
	(INTP4)	Input	PIOR04 = 1	–	1	×	0	–	–		
P75	P75	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	KR5	Input	×	–	1	×	0	–	–		
	SEG21	Output	×	–	0	0	1	–	–		
	(INTP5)	Input	PIOR04 = 1	–	1	×	0	–	–		
P76	P76	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	KR6	Input	×	–	1	×	0	–	–		
	SEG22	Output	×	–	0	0	1	–	–		
	(INTP6)	Input	PIOR04 = 1	–	1	×	0	–	–		
	SMP5	Input	–	–	1	×	0	–	–	√	–
P77	P77	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	KR7	Input	×	–	1	×	0	–	–		
	SEG23	Output	×	–	0	0	1	–	–		
	(INTP7)	Input	PIOR04 = 1	–	1	×	0	–	–		
	SMP4	Input	–	–	1	×	0	–	–	√	–
P80	P80	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(SCK10/SCL10) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG12	Output	×	×	0	0	1	–	–		
	(SCL10)	Output	PIOR02 = 1	0/1	0	1	0	–	–		
	(SCK10)	Input	PIOR02 = 1	×	1	×	0	–	–		
		Output	PIOR02 = 1	0/1	0	1	0	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (9/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P81	P81	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(SDA10) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG13	Output	×	×	0	0	1	–	–		
	(RxD1)	Input	PIOR02 = 1	×	1	×	0	–	–		
	(SDA10)	I/O	PIOR02 = 1	1	0	1	0	–	–		
	(SI10)	Input	PIOR02 = 1	×	1	×	0	–	–		
P82	P82	Input	–	×	1	×	0	–	–	√	√
		Output	×	0	0	0/1	0	(TxD1/SO10) = 1	–		
		N-ch open drain output	×	1	0	0/1	0				
	SEG14	Output	×	×	0	0	1	–	–		
	(TxD1)	Output	PIOR02 = 1	0/1	0	1	0	–	–		
	(SO10)	Output	PIOR02 = 1	0/1	0	1	0	–	–		
P83	P83	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG15	Output	–	–	0	0	1	–	–		
P84	P84	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	SDA30 = 1	TRJIO1 = 0		
		N-ch open drain output	–	1	0	0/1	0				
	SEG40	Output	–	–	0	0	1	–	–		
	SI30	Input	–	×	1	×	0	–	–		
	RxD3	Input	–	×	1	×	0	–	–		
	SDA30	I/O	–	1	0	1	0	–	–		
	TRJIO1	Input	–	0	1	×	0	–	–		
		Output	–	0	0	0	0	SDA30 = 1	–		
P85	P85	Input	–	–	1	×	0	–	–	–	√
		Output	–	–	0	0/1	0	SO30/TxD3 = 1	–		
		N-ch open drain output	–	1	0	0/1	0				
	SEG41	Output	–	–	0	0	1	–	–		
	SO30	Output	–	0/1	0	1	0	–	–		
	TxD3	Output	–	0/1	0	1	0	–	–		
	SMP0	Input	–	–	1	×	0	–	–		
P90	P90	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	COM0	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG31)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMP1	Input	–	–	1	×	0	–	–		
P91	P91	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	–		
	COM1	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG30)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMP2	Input	–	–	1	×	0	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (10/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	Alternate Function Output		80-pin	100-pin
	Function Name	I/O						SAU and UARTMG	Other than SAU and UARTMG		
P92	P92	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	SMO10 = 0		
	COM2	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG29)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMO10	Output	–	–	0	0	0	–	–		
P93	P93	Input	–	–	1	×	0	–	–	√	√
		Output	×	–	0	0/1	0	–	SMO11 = 0		
	COM3	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG28)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMO11	Output	–	–	0	0	0	–	–		
P94	P94	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	SMO12 = 0		
	SEG0	Output	PIOR05 = 0	–	0	0	1	–	–		
	COM4	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG27)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMO12	Output	–	–	0	0	0	–	–		
P95	P95	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG1	Output	PIOR05 = 0	–	0	0	1	–	–		
	COM5	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG26)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMP3	Input	–	–	1	×	0	–	–		
P96	P96	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG2	Output	PIOR05 = 0	–	0	0	1	–	–		
	COM6	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG25)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMP4	Input	–	–	1	×	0	–	–		
P97	P97	Input	–	–	1	×	0	–	–	√	√
		Output	–	–	0	0/1	0	–	–		
	SEG3	Output	PIOR05 = 0	–	0	0	1	–	–		
	COM7	Output	PIOR05 = 0	–	0	0	1	–	–		
	(SEG24)	Output	PIOR05 = 1	–	0	0	1	–	–	–	√
	SMP5	Input	–	–	1	×	0	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (11/12)

Pin Name	Used Function		PIOR0	Pxx	CMC Register (EXCLK, OSCSEL)	SCMC Register (EXCLKS, OSCSELS)	80-pin	100-pin
	Function Name	I/O						
P121	P121	Input	–	×	00/10/11	–	√	√
	X1	–	–	–	01			
	INTP9	Input	PIOR06 = 0	–	00/10/11			
P122	P122	Input	–	×	00/10		√	√
	X2	–	–	–	01			
	EXCLK	Input	–	–	11			
	INTP8	Input	PIOR06 = 0	–	00/10			
P123	P123	Input	–	×	–	00/10/11	√	√
	XT1	–	–	–		01		
P124	P124	Input	–	×		00/10	√	√
	XT2	–	–	–		01		
	EXCLKS	Input	–	–		11		

<R>

Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (12/12)

Pin Name	Used Function		PIOR0	POM××	PM××	P××	PFSEG×× (ISCVL3, ISCCAP)Note	TCEN bit	Alternate Function Output		80-pin	100-pin
	Function Name	I/O							SAU and UARTMG	Other than SAU and UARTMG		
P125	P125	Input	–	–	1	×	1	–	–	–	√	√
		Output	×	–	0	0/1	1	–	–	(TO05) = 0 PCLBUZ1 = 0		
	V_{L3}	–	×	–	1	×	0	–	–	–		
	INTP1	Input	PIOR04 = 0	–	1	×	1	–	–	–		
	(TI05)	Input	PIOR00 = 1	–	1	×	1	–	–	–		
	(TO05)	Output	PIOR00 = 1	–	0	0	1	–	–	PCLBUZ1 = 0		
P126	P126	Input	–	–	1	×	1	–	–	–	√	√
		Output	×	–	0	0/1	1	–	–	(TO04) = 0		
	CAPL	–	×	–	1	×	0	–	–	–		
	(TI04)	Input	PIOR00 = 1	–	1	×	1	–	–	–		
	(TO04)	Output	PIOR00 = 1	–	0	0	1	–	–	–		
P127	P127	Input	–	–	1	×	1	–	–	–	√	√
		Output	×	–	0	0/1	1	–	–	(TO03) = 0		
	CAPH	–	×	–	1	×	0	–	–	–		
	(TI03)	Input	PIOR00 = 1	–	1	×	1	–	–	–		
	(TO03)	Output	PIOR00 = 1	–	0	0	1	–	–	–		
P130	P130	Output	–	–	–	0/1	–	–	–	PCLBUZ1 = 0 SMO02 = 0	√	√
	PCLBUZ1	Output	PIOR03 = 0	–	–	0	–	–	–	SMO02 = 0	√	–
	SMO02	Output	–	–	–	0	0	–	–	PCLBUZ1 = 0		
P137	P137	Input	–	–	–	×	–	–	–	–	√	√
	INTP0	Input	PIOR04 = 0	–	–	×	–	–	–	–		
P150	P150	Input	–	×	1	×	–	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	×	–	0	0/1	–	0	–	–		
	RTCOUT	Output	PIOR03 = 0	–	0	0	–	0	–	–		
	RTCIC0	Input	×	–	1	×	–	1	–	–		
	INTP14	Input	–	–	–	×	–	–	–	–		
P151	P151	Input	–	×	1	×	–	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	–	–	0	0/1	–	0	–	–		
	RTCIC1	Input	–	–	1	×	–	1	–	–		
	INTP13	Input	–	–	–	×	–	–	–	–		
P152	P152	Input	–	–	1	×	–	–	–	–	√	√
		N-ch open drain output (6 V tolerance)	–	–	0	0/1	–	0	–	–		
	RTCIC2	Input	–	–	1	×	–	1	–	–		
	INTP12	Input	–	–	–	×	–	–	–	–		

Note ISCVL3 and ISCCAP are registers that correspond to P125, and P126 and P127, respectively.

4.5.4 Operation of ports that have multiplexed SEGxx or COMx pin functions

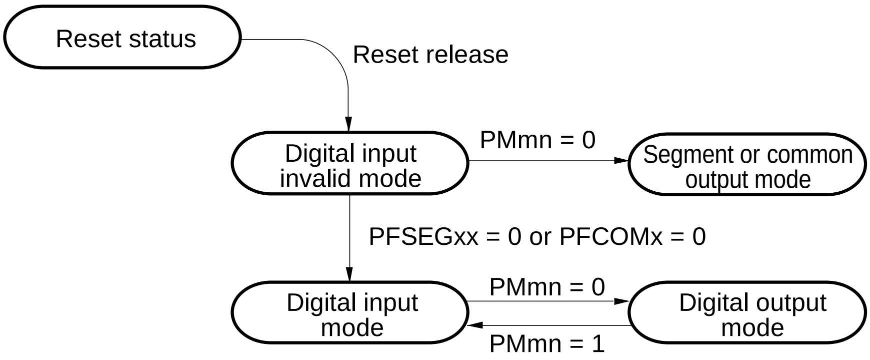
The functions of port pins that also serve as segment output pins (SEGxx) or common output pins (COMx) can be selected by using the port mode register (PMxx), and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

Table 4-9. Setting of the SEGxx or COMx and Port Pin Functions

PFSEGxx or PFCOMx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input invalid mode	√
0	0	Digital output mode	–
0	1	Digital input mode	–
1	0	Segment or common output mode	–

The following shows the state transitions of the SEGxx or COMx and port pin functions.

Figure 4-11. State Transition Diagram of SEGxx or COMx and Port Pin Functions



Caution Be sure to set the segment or common output mode before segment or common output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

4.5.5 Operation of ports that alternately function as V_{L3}, CAPL, CAPH pins

The functions of the V_{L3}/P125, CAPL/P126, CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

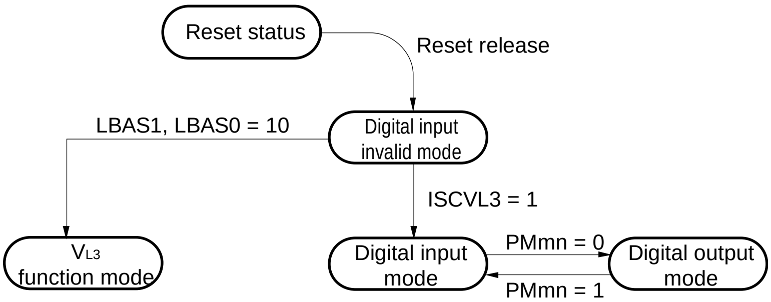
(1) V_{L3}/P125

Table 4-10. Settings of V_{L3}/P125 Pin Function

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register)	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	V_{L3} function mode	–
Other than above			Setting prohibited	

The following shows the V_{L3}/P125 pin function status transitions.

Figure 4-12. V_{L3}/P125 Pin Function Status Transition Diagram



Caution Be sure to set the V_{L3} function mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

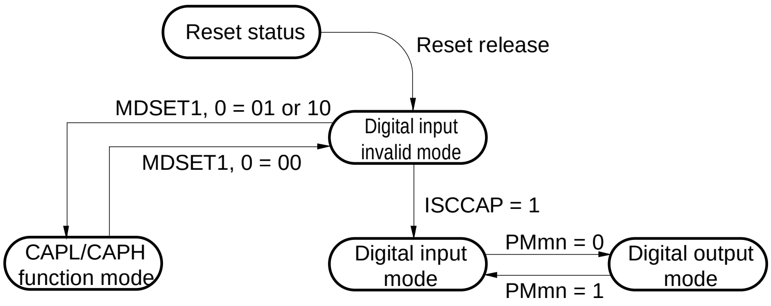
(2) CAPL/P126, CAPH/P127

Table 4-11. Settings of CAPL/P126, CAPH/P127 Pins Function

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126, PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	–
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pins function status transitions.

Figure 4-13. CAPL/P126 and CAPH/P127 Pins Function Status Transition Diagram



Caution Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC of LCD mode register 1 (LCDM1) is 0).

4.6 Cautions When Using Port Function

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

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

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

<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port 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 is FFH.

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

A 1-bit manipulation instruction is executed in the following order in the RL78/I1C (512 KB).

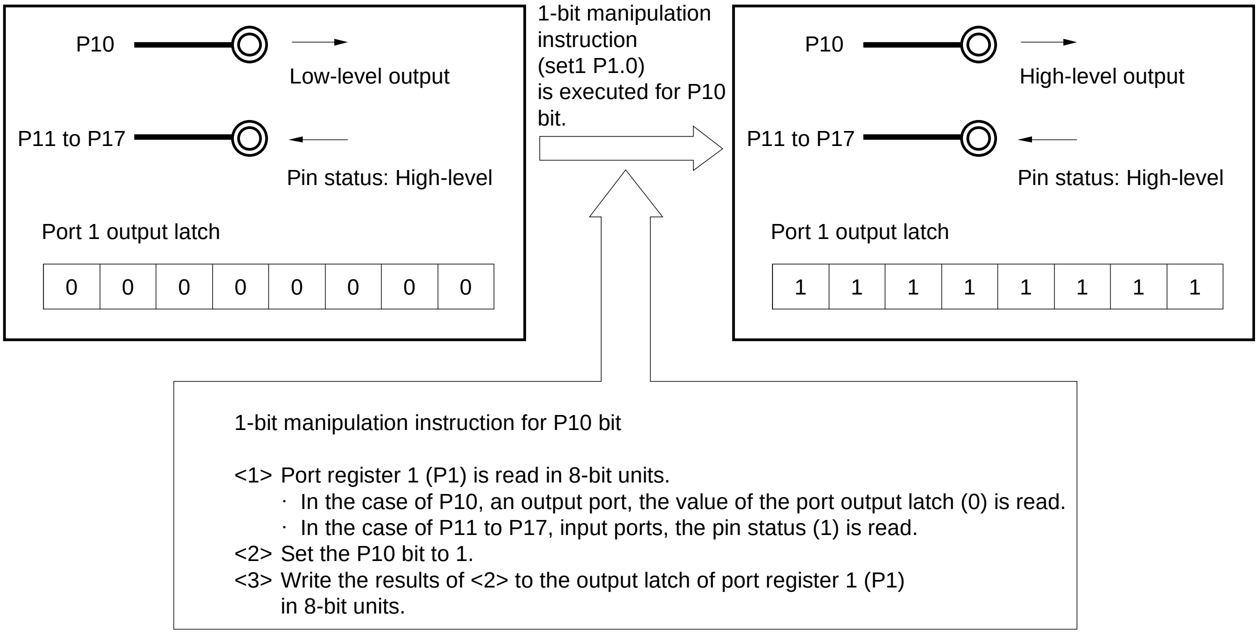
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses 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 4-14. Bit Manipulation Instruction (P10)



0	0	0	0	0	0	0	0

1	1	1	1	1	1	1	1

1-bit manipulation instruction for P10 bit

<1> Port register 1 (P1) is read in 8-bit units.
· In the case of P10, an output port, the value of the port output latch (0) is read.
· In the case of P11 to P17, input ports, the pin status (1) is read.
<2> Set the P10 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

4.6.2 Notes on specifying the pin settings

If the output function of an alternate function is assigned to a pin that is also used as an output pin, the output of the unused alternate function must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR0). For details about the alternate output function, see **4.5 Register Settings When Using Alternate Function**.

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

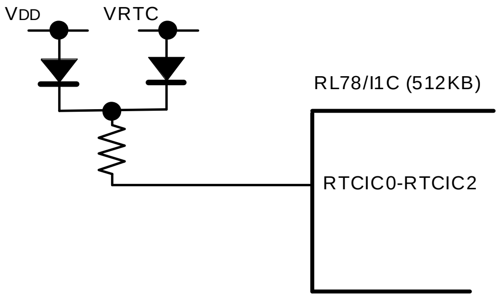
Disabling the unused functions, including blocks that are only used for input or do not have output, is recommended to lower power consumption.

<R> 4.6.3 Point to Note on the P150 to P152

Note the following points when using a P150/RTCIC0- P152/RTCIC2 pin under any of conditions (1) to (4) stated below.

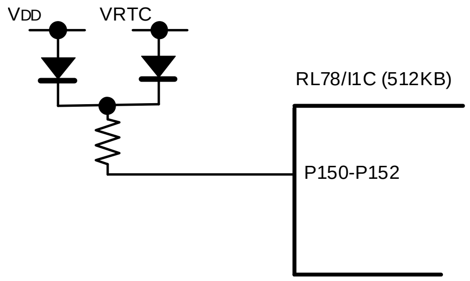
(1) A pin is in use as a time-capture event input RTCICn (n = 0 to 2).

Setting a TCEN bit in the RTCCRy (y = 0 to 2) register enables operation of the pin as a time-capture input pin (RTCICn). Each of the P150/RTCIC0 to P152/RTCIC2 pins has two input buffers powered by V_{DD} and VRTC. Since the input buffers are always on, a through-current might flow when an intermediate potential is input to the input buffers. Accordingly, configure the circuit as follows such that each pin selected for time-capture input is always individually connected to either V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V. This is to prevent the input of an intermediate potential to the given pin even in cases where, for example, the V_{DD} voltage fluctuates from being greater than or equal to VRTC to being less than VRTC.



(2) A P150 to P152 pin is in use as an input port pin.

Each of the P150 to P152 pins has two input buffers powered by V_{DD} and VRTC. Since the input buffers are always on, a through-current might flow when an intermediate potential is input to the input buffers. Accordingly, when a high-level signal is to be input to any pin among P150 to P152, configure the circuit as follows so that the pin is always individually connected to either V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V. This is to prevent the input of an intermediate potential to the given pin even in cases where, for example, the V_{DD} voltage fluctuates from being greater than or equal to VRTC to being less than VRTC.



(3) A P150 to P152 pin is in use as an output port (N-ch open-drain output) pin.

When a P150 to P152 pin is in use in the output mode, set the corresponding TCEN bit in the RTCCRy (y = 0 to 2) register to 0. When a high-level signal is to be output from any pin among P150 to P152, handling of the pin is the same as in the input mode. Configure the circuit so that the pin is always individually connected to V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V. When a reset occurs due to the V_{DD} power dropping or being cut off while the output from any of these pins is a low-level signal, they are all set for the input mode.

(4) A P150/RTCIC0 to P152/RTCIC2 pin is not in use.

Set the P150 to P152 pins to the input mode, and individually connect them via a resistor to V_{SS}, in accord with the recommended connection of unused pins.

CHAPTER 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 according to the supply voltage range and clock frequencies used to operate the MCU.

The flash operation mode set by the option byte is selected for operation immediately after a reset release. Then, each mode is selected according to register settings.

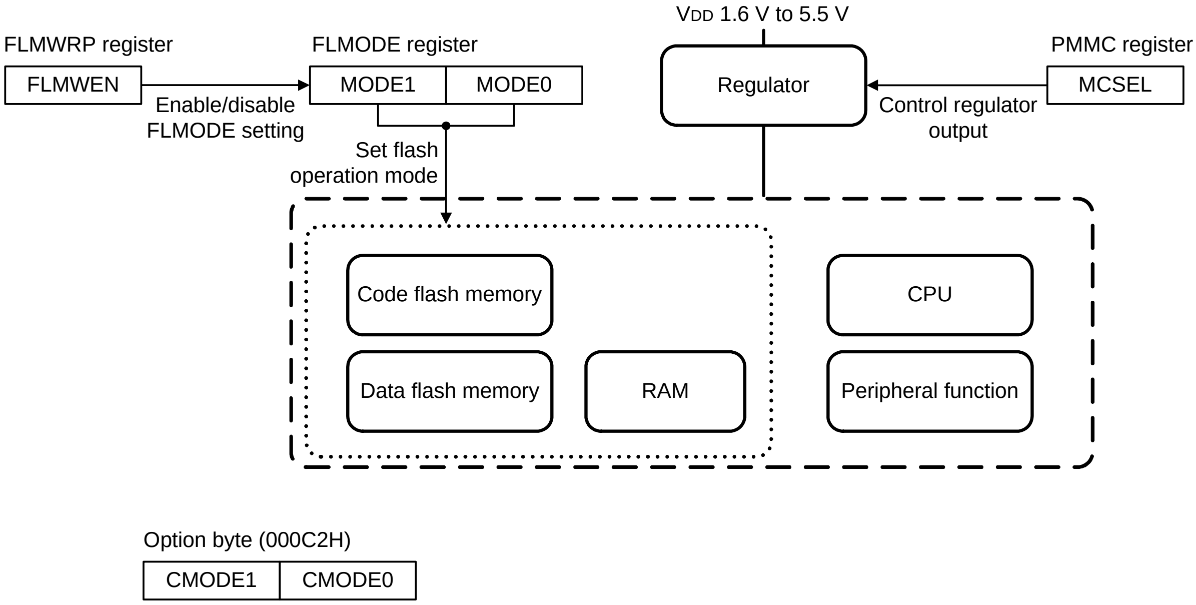
5.1 Configuration of Operation State Control

Operation state control is supported by the following hardware.

Table 5-1. Configuration of Operation State Control

Item	Configuration
Option byte	● User option byte address: 000C2H
Control registers	● Flash operating mode select register (FLMODE) ● Flash operating mode protect register (FLMWRP) ● Regulator mode control register (PMMC)

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
- ● LV (low-voltage main) mode
- ● LP (low-power main) 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	Regulator Mode	Recommended Operating Range		Operating Current (typ.)	Description
HS (high-speed main) mode	Normal setting only (MCSEL = 0)	2.7 V to 5.5 V	1 MHz to 32 MHz	5.6 mA (during operation at 32 MHz$^{\text{Note 1}}$)	High-speed CPU operation (at 32 MHz (max.)) is possible in this mode. Suitable when CPU processing capacity is required.
		2.4 V to 5.5 V	1 MHz to 16 MHz		
		2.1 V to 5.5 V	1 MHz to 6 MHz		
LS (low-speed main) mode	Normal setting (MCSEL = 0)	1.8 V to 5.5 V	1 MHz to 8 MHz	1.5 mA (during operation at 8 MHz$^{\text{Note 1}}$)	The operating current and CPU operation processing (at 8 MHz (max.)) are well-balanced in this mode. To operate the CPU at 4 to 8 MHz, set regulator mode to the normal setting. When operating the CPU at 1 to 4 MHz, the operating current can be reduced by setting regulator mode to the low-power consumption setting.
	Low-power consumption setting (MCSEL = 1)	1.8 V to 5.5 V	1 MHz to 4 MHz	0.8 mA (during operation at 4 MHz$^{\text{Note 2}}$)	
LP (low-power main) mode	Low-power consumption setting only (MCSEL = 1)	1.8 V to 5.5 V	1 MHz	170 μA (during operation at 1 MHz$^{\text{Note 2}}$)	The CPU operates at 1 MHz in this mode. Low operating current is realized at 1 MHz.
LV (low-voltage main) mode$^{\text{Note 1}}$	Normal setting only (MCSEL = 0)	1.6 V to 5.5 V	1 MHz to 4 MHz	1.7 mA (during operation at 4 MHz)	Low-voltage operation up to 1.6 V is possible in this mode. To operate the CPU at the supply voltage range of 1.6 to 1.8 V, select this mode.

Notes 1. Operable only with the high-speed on-chip oscillator.

2. When the middle-speed on-chip oscillator operates.

5.2 Registers Controlling Operation State Control

Operation state control is controlled by the following registers.

- ● Flash operating mode select register (FLMODE)
- ● Flash operating mode protect register (FLMWRP)
- ● Regulator mode control register (PMMC)

5.2.1 Flash operating mode select register (FLMODE)

The FLMODE register is an 8-bit register used to control flash operation modes and operation of the code flash memory.

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 while the setting of the FLMWEN bit of the flash operating mode protect register (FLMWRP) is 0.

Reset generation updates MODE1 and MODE0 with the set value of CMODE1 and CMODE0 in the option byte (address: 000C2H).

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

Address: F00AAH After reset: 00H/80H/C0H[Note 1] 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	LV (low-voltage main) mode (Selectable when 1 MHz ≤ f_{CLK} ≤ 4 MHz in LS mode.)
0	1	LP (low-power main) mode (Selectable when 1.8 V ≤ V_{DD} ≤ 5.5 V and f_{CLK} = 1 MHz in LS mode. [Note 2])
1	0	LS (low-speed main) mode (Selectable when 1.8 V ≤ V_{DD} ≤ 5.5 V and 1 MHz ≤ f_{CLK} ≤ 8 MHz in HS mode, LP mode, or LV mode.)
1	1	HS (high-speed main) mode Selectable when 2.4 V ≤ V_{DD} ≤ 5.5 V in LS mode.)

Notes 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).

2. After LP (low-power main) mode is selected, set the MCSEL bit in the regulator mode control register (PMMC) to 1.

(**Cautions** are on the next page.)

Cautions 1. The value of the FLMODE register cannot be changed when the FLMWEN bit in the flash operation mode protect register (FLMWRP) is 0. Also, do not change the value of the FLMODE register when the MCSEL bit in the regulator mode control register is 1.

When changing the value of the FLMODE register, first set the FLMWEN bit in the FLMERP register to 1 while MCSEL is 0. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.

2. The MODE1 and MODE0 bits cannot be set when the CSS bit in the system clock control register (CKC) is 1 (CPU/peripheral function operates on subsystem clock).

3. Do not change the value of the MODE1 and MODE0 bits using the DTC.

4. When changing the flash operation mode, make sure that operation is possible within the voltage range and operating frequency range in the changed flash operation mode before changing the mode.

5. The middle-speed on-chip oscillator cannot be used in LV (low-voltage main) mode. When entering LV mode, first switch the operating clock to an oscillator other than the middle-speed on-chip oscillator, and then enter LV mode.

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

Table 5-3. Flash Operation Mode Change Time

Flash Operation Mode Change	Change Time
LS (low-speed main) mode → HS (high-speed main) mode	225 cycles[Note 1]
LS (low-speed main) mode → LV (low-voltage main) mode	99 cycles[Notes 1, 2]
LP (low-power main) mode → LS (low-speed main) mode	10 cycles[Note 1]
LS (low-speed main) mode → LP (low-power main) mode	10 cycles[Note 1]
LV (low-voltage main) mode → LS (low-speed main) mode	20 cycles[Note 1]
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles[Note 1]

Notes 1. The cycle of the CPU/peripheral hardware clock (f_{CLK})

2. Switching of the mode from LS (low-speed main) mode to LV (low-voltage main) mode must proceed while oscillation of the high-speed on-chip oscillator is stable.

Cautions 7. When rewriting the FLMODE register, insert one or more clock cycles after rewriting the FLMODE register and before writing to this register. Do not write to the FLMODE register successively.

8. Do not change the FLMODE register when rewriting the flash memory.

5.2.2 Flash operating mode protect register (FLMWRP)

The FLMWRP register is an 8-bit register used to control access to the flash operation mode select register. This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset generation sets this register to 00H.

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

Address: F00ABH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
FLMWRP	0	0	0	0	0	0	0	FLMWEN

FLMWEN	Control of flash operation mode select register (FLMODE)
0	Rewriting the FLMODE register is disabled
1	Rewriting the FLMODE register is enabled

5.2.3 Regulator mode control register (PMMC)

The PMMC register is an 8-bit register used to control the mode of the on-chip regulator. This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset generation sets this register to 00H.

Figure 5-4. Format of Regulator Mode Control Register (PMMC)

Address: F00F8H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	1	0
PMMC	0	MCSEL	0	0	0	0	0	0

MCSEL	Control of regulator mode
0	Normal setting
1	Low-power consumption setting

Cautions 1. Do not change the flash operation mode select register (FLMODE) when MCSEL is 1.

2. Do not set MCSEL to 1 in HS (high-speed main) mode and LV (low-voltage main) mode.

3. In LS (low-speed main) mode, transition to the STOP mode is prohibited when MCSEL is 1.

5.3 Initial Setting of Flash Operation Modes

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

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

When a reset is released, the value of CMODE1 and CMODE0 is updated in MODE1 and MODE0 in the flash operation mode select register (FLMODE) and the value of FRQSEL2 to FRQSEL0 is updated in the high-speed on- chip oscillator frequency select register (HOCODIV).

Figure 5-5. Format of User Option Byte (000C2H)

Address: 000C2H

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

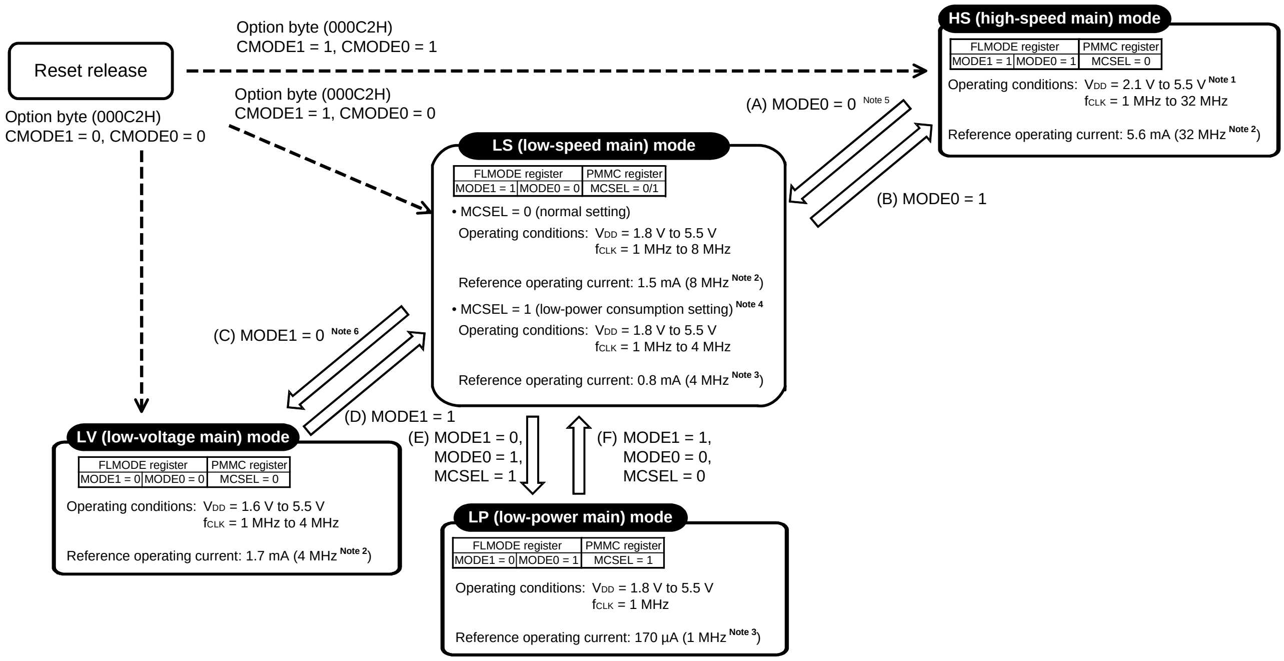
CMODE1	CMODE0	Selection of flash operation mode after reset release
0	0	LV (low-voltage main) mode
1	0	LS (low-speed main) mode
1	1	HS (high-speed main) mode
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator frequency
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
0	1	0	0	1.5 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

5.4 Transitions between Flash Operation Modes

HS (high-speed main) mode, LS (low-speed main) mode, or LV (low-voltage main) mode can be selected as the flash operation mode immediately after a reset release, by setting CMODE1 and CMODE0 in the option byte (000C2H). The value of CMODE1 and CMODE0 is updated in the MODE1 and MODE0 bits in the flash operation mode select register (FLMODE). After that, the flash operation mode can be changed by changing the value of the FLMODE register during CPU operation.

Figure 5-6. State Transitions between Flash Operation Modes


Reset release
Option byte (000C2H)
CMODE1 = 1, CMODE0 = 1
Option byte (000C2H)
CMODE1 = 1, CMODE0 = 0
Option byte (000C2H)
CMODE1 = 0, CMODE0 = 0
HS (high-speed main) mode
FLMODE register | PMMC register
MODE1 = 1 | MODE0 = 1 | MCSEL = 0
Operating conditions: VDD = 2.1 V to 5.5 V Note 1
fCLK = 1 MHz to 32 MHz
Reference operating current: 5.6 mA (32 MHz Note 2)
(A) MODE0 = 0 Note 5
(B) MODE0 = 1
LS (low-speed main) mode
FLMODE register | PMMC register
MODE1 = 1 | MODE0 = 0 | MCSEL = 0/1
• MCSEL = 0 (normal setting)
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz to 8 MHz
Reference operating current: 1.5 mA (8 MHz Note 2)
• MCSEL = 1 (low-power consumption setting) Note 4
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz to 4 MHz
Reference operating current: 0.8 mA (4 MHz Note 3)
(C) MODE1 = 0 Note 6
(D) MODE1 = 1
LV (low-voltage main) mode
FLMODE register | PMMC register
MODE1 = 0 | MODE0 = 0 | MCSEL = 0
Operating conditions: VDD = 1.6 V to 5.5 V
fCLK = 1 MHz to 4 MHz
Reference operating current: 1.7 mA (4 MHz Note 2)
(E) MODE1 = 0,
MODE0 = 1,
MCSEL = 1
(F) MODE1 = 1,
MODE0 = 0,
MCSEL = 0
LP (low-power main) mode
FLMODE register | PMMC register
MODE1 = 0 | MODE0 = 1 | MCSEL = 1
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz
Reference operating current: 170 μA (1 MHz Note 3)

Notes 1. The operating frequency and operating voltage range are as follows.

$1\text{ MHz} \le f_{CLK} \le 6\text{ MHz}$ ($2.1\text{ V} \le V_{DD} \le 5.5\text{ V}$)

$1\text{ MHz} \le f_{CLK} \le 16\text{ MHz}$ ($2.4\text{ V} \le V_{DD} \le 5.5\text{ V}$)

$1\text{ MHz} \le f_{CLK} \le 32\text{ MHz}$ ($2.7\text{ V} \le V_{DD} \le 5.5\text{ V}$)

2. Current when the high-speed on-chip oscillator operates.

3. Current when the middle-speed on-chip oscillator operates.

4. Transitions between flash operation modes or transition to the STOP mode cannot be made when MCSEL = 1 (low- power consumption setting). When changing the flash operation mode or making transition to the STOP mode, be sure to set MCSEL = 0 (normal setting) before changing the mode.

5. When CMODE1 and CMODE0 of the option byte (000C2H) are set to 1, operation is not guaranteed if a reset is generated while the operating voltage is 2.4 V or lower after entry to the LS (low-speed main) mode.

6. When CMODE1 and CMODE0 of the option byte (000C2H) are set to 1 and 0 respectively, operation is not guaranteed if a reset is generated while the operating voltage is 1.8 V or lower after entry to the LV (low-voltage main) mode.

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

5.5 Details of Flash Operation Modes

5.5.1 Details of HS (high-speed main) mode

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

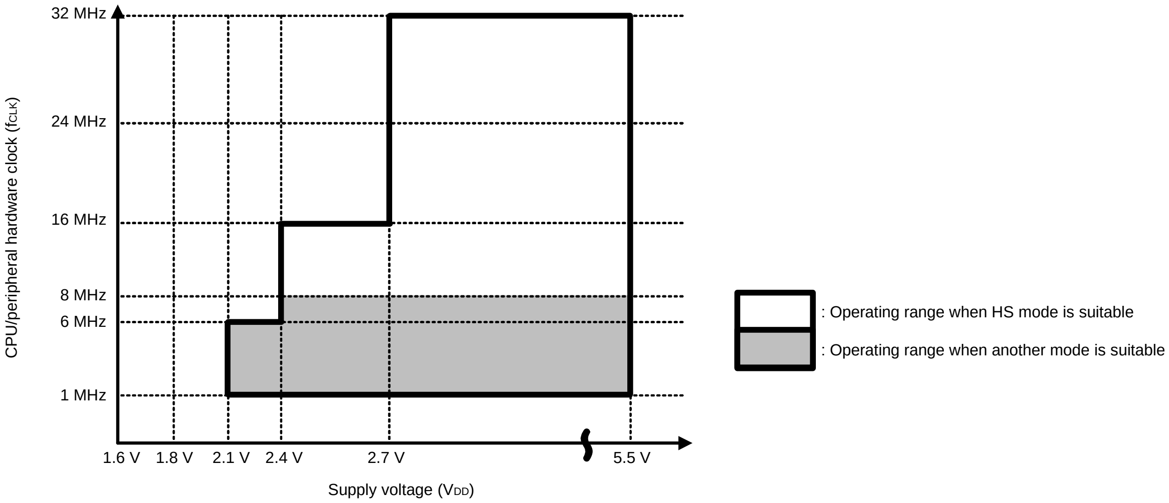
HS mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode. When entering HS mode, make sure that the supply voltage is 2.4 V ≤ V_{DD} ≤ 5.5 V in LS mode.

Operating in HS mode is suitable when the power supply voltage and operating frequency meet any of the following conditions.

- 2.4 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 8 MHz < f_{CLK} ≤ 16 MHz
- 2.7 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 8 MHz < f_{CLK} ≤ 32 MHz

When 8 MHz or lower is used for operation, another mode can be used as the suitable flash operation mode.

Figure 5-7. Operating Range in HS Mode



5.5.2 Details of LS (low-speed main) mode

LS (low-speed main) mode supports both CPU processing capacity and operating voltage performance, suitable for applications that require low-power consumption at 1 to 8 MHz.

LS mode can be operated immediately after a reset release. Also, this mode can be entered from HS (high-speed main) mode, LV (low-voltage main) mode, or LP (low-power main) mode. When entering from HS mode to LS mode, make sure that the operating frequency is 1 MHz ≤ f_{CLK} ≤ 8 MHz.

In LS mode, low-power consumption can be set by the MCSEL bit in the regulator mode control register (PMMC). When setting low-power consumption, set the MCSEL bit to 1 while the operating frequency is 1 MHz ≤ f_{CLK} ≤ 4 MHz.

The suitable operating range in LS mode is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 4 MHz < f_{CLK} ≤ 8 MHz if MCSEL = 0, and when the supply voltage is 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 1 MHz < f_{CLK} ≤ 4 MHz if MCSEL = 1.

Figure 5-8. Operating Range in LS Mode (MCSEL = 0)

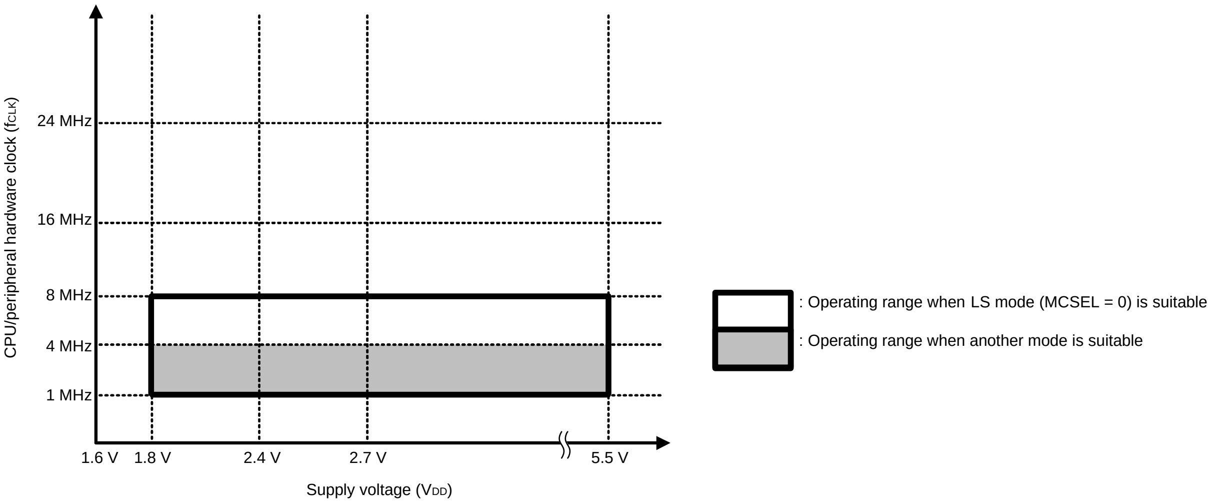
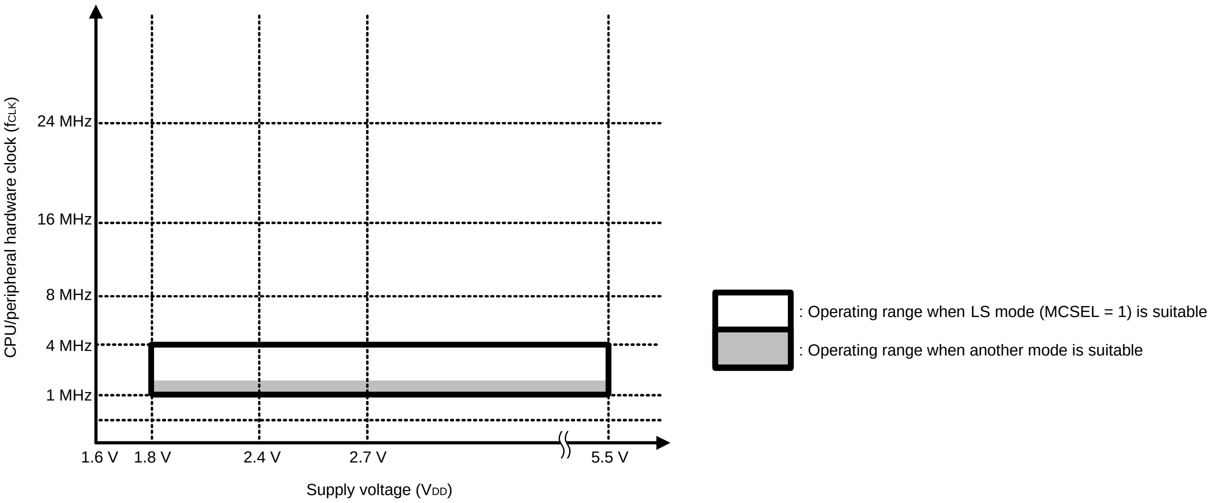


Figure 5-9. Operating Range in LS Mode (MCSEL = 1)



Caution When entering another flash operation mode, make sure that MCSEL = 0.

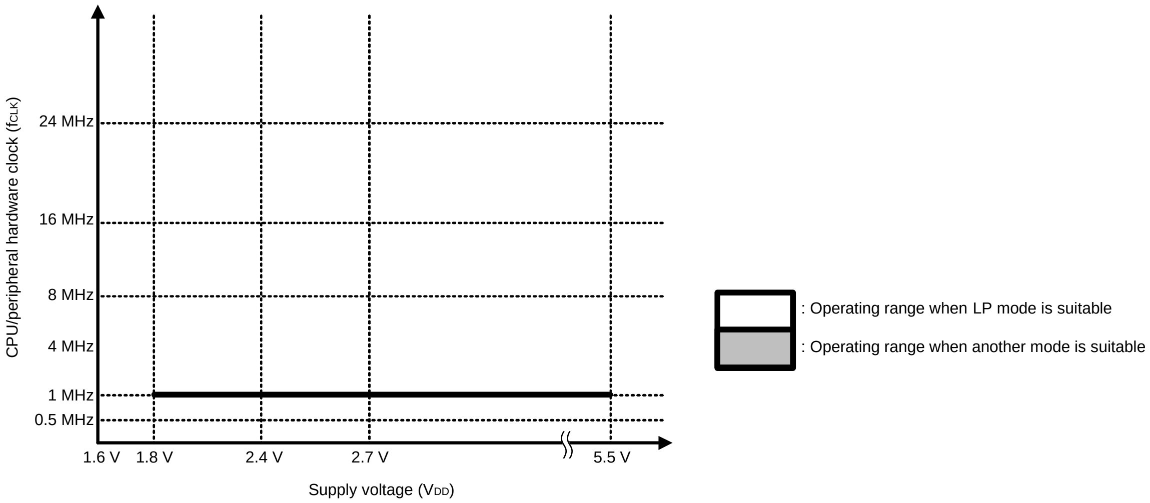
5.5.3 Details of LP (low-power main) mode

LP (low-power main) mode can be use to operate the CPU on low power at a 1-MHz frequency.

LP mode can be entered from LS (low-speed main) mode. When entering from LS mode to LP mode, make sure the operating frequency is f_{CLK} = 1 MHz. After the mode is entered, set the MCSEL bit in the regulator mode control register to 1.

The suitable operating range in LP mode is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 1 MHz.

Figure 5-10. Operating Range in LP Mode



Cautions 1. When entering LS (low-speed main) mode, make sure that MCSEL = 0.

2. The 24-bit ΔΣ A/D converter cannot be used in LP (low-power main) mode or LV (low-voltage main) mode.

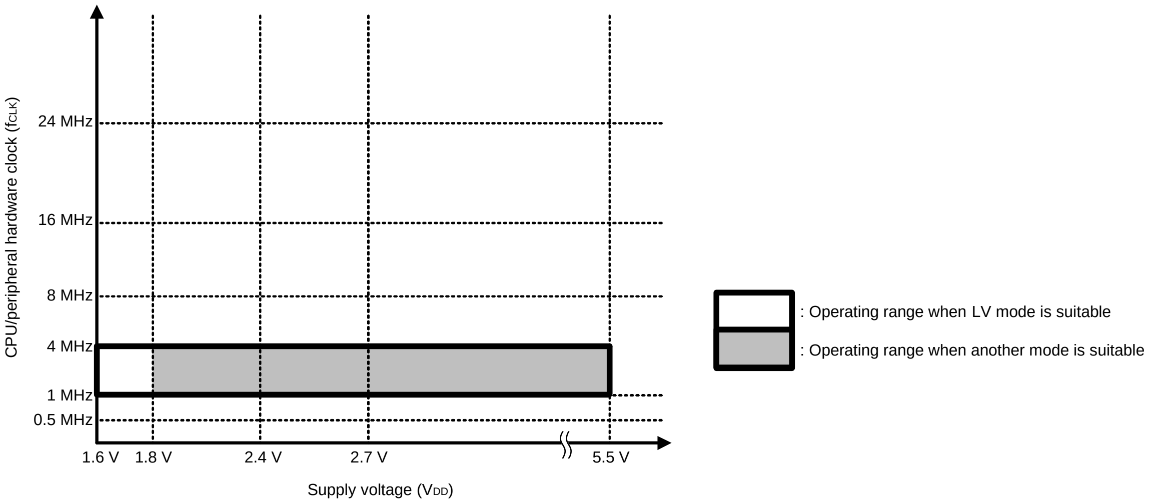
5.5.4 Details on LV (low-voltage main) mode

LV (low-voltage main) mode is suitable for applications that require operation at 1.8 V or lower.

LV mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode. When entering from LS mode to LV mode, make sure that the operating frequency is 1 MHz ≤ f_{CLK} ≤ 4 MHz.

The suitable operating range in LV mode is when the supply voltage is 1.6 V ≤ V_{DD} < 1.8 V. When a supply voltage of 1.8 V ≤ V_{DD} ≤ 5.5 V is used for operation, another mode can be used as the suitable flash operation mode.

Figure 5-11. Operating Range in LV Mode



Cautions 1. The middle-speed on-chip oscillator cannot be used in LV (low-voltage main) mode. When entering LV mode, first switch the operating clock to an oscillator other than the middle-speed on-chip oscillator, and then enter LV mode.

2. The 24-bit ΔΣ A/D converter cannot be used in LP (low-power main) mode or LV (low-voltage main) mode.

CHAPTER 6 CLOCK GENERATOR

6.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

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

Note When the high-speed system clock (f_{MX}) is supplied as the operating clock for the 24-bit ΔΣ A/D converter, only the 12-MHz crystal resonator can be used.

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 24, 12, 6, 3, or 1.5 MHz (when the FRQSEL3 bit is set to 0) or f_{IH} = 32, 16, 8, 4, 2, or 1 MHz (when the FRQSEL3 bit is set to 1) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 6-14 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	1.5	2	3	4	6	8	12	16	24	32
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	√	√
2.4 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	—	—
1.8 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	—	—	—	—
1.6 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	—	—	—	—	—	—

Caution To supply the high-speed on-chip oscillator clock as the operating clock for the 24-bit ΔΣ A/D converter, set the FRQSEL3 bit to 0 (f_{HOCO} = 24 MHz). $f_{HOCO}/2$ = 12 MHz is supplied regardless of the frequency setting of the high-speed on-chip oscillator clock (f_{IH}).

<3> Middle-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IM} = 4, 2, 1 MHz (TYP.) by setting of the MOCODIV bit (bits 0, 1 of the MOCODIV register). Oscillation can be stopped by executing the STOP instruction or clearing of the MIOEN bit (bit 1 of the CSC register).

<4> PLL oscillator

Selecting the PLL clock by setting the main clock control register (MCKC) and setting the PLL control register (DSCCTL) causes the PLL to oscillate and produce a 32-MHz (f_{PLL}) clock signal.

An external main system clock (f_{EX} = 1 to 20 MHz$^{\text{Note}}$) can also be supplied from the EXCLK/X2/P122/INTP8 pin. An external main system clock input can be disabled by executing the 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), high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or PLL clock can be selected.

The available frequency range of the main system clock varies depending on the power supply voltage V_{DD}. The operation voltage mode of the flash memory must be set with CMODE0 and CMODE1 of the option byte (000C2H) or the flash operating mode select register (FLMODE) (see **CHAPTER 37 OPTION BYTE** or **CHAPTER 5 OPERATION STATE CONTROL**).

Note When the high-speed system clock (f_{MX}) is supplied as the operating clock for the 24-bit ΔΣ A/D converter, supply a 12-MHz signal as the external main system clock (f_{EX}).

(2) Subsystem clock

<1> XT1 clock oscillator

This circuit oscillates a clock of f_{XT} = 32.768 or 38.4 kHz by connecting a 32.768 or 38.4 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 or 38.4 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by the setting of the XTSTOP bit.

Caution The XT1 clock oscillator runs on the VRTC power-supply. It can operate after release from the RTC power-on reset following the power supply to the VRTC pin being turned on. If the voltage on the VRTC pin falls below the detection voltage (V_{PDR}), an RTC power-on reset is generated and the XT1 clock oscillator stops.

<2> Low-speed on-chip oscillator

This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).

The low-speed on-chip oscillator clock can be used as the CPU clock. The following peripheral hardware is driven by the low-speed on-chip oscillator clock.

- Watchdog timer
- 12-bit interval timer
- 8-bit interval timer
- Frequency measurement circuit
- Oscillation stop detection circuit
- LCD controller/driver
- Timers RJ0 and RJ1

This clock operates when any bit among bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply option control register (OSMC), or bit 0 of the subsystem clock select register (CKSEL) is set to 1 (including multiple bits).
However, when WDTON = 1, WUTMMCK0 = 0, SELLOSC = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Remark

f_X:	X1 clock oscillation frequency
f_{IH}:	High-speed on-chip oscillator clock frequency (32 MHz max.)
f_{IM}:	Middle-speed on-chip oscillator clock frequency
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 frequency
f_{PLL}:	PLL clock frequency

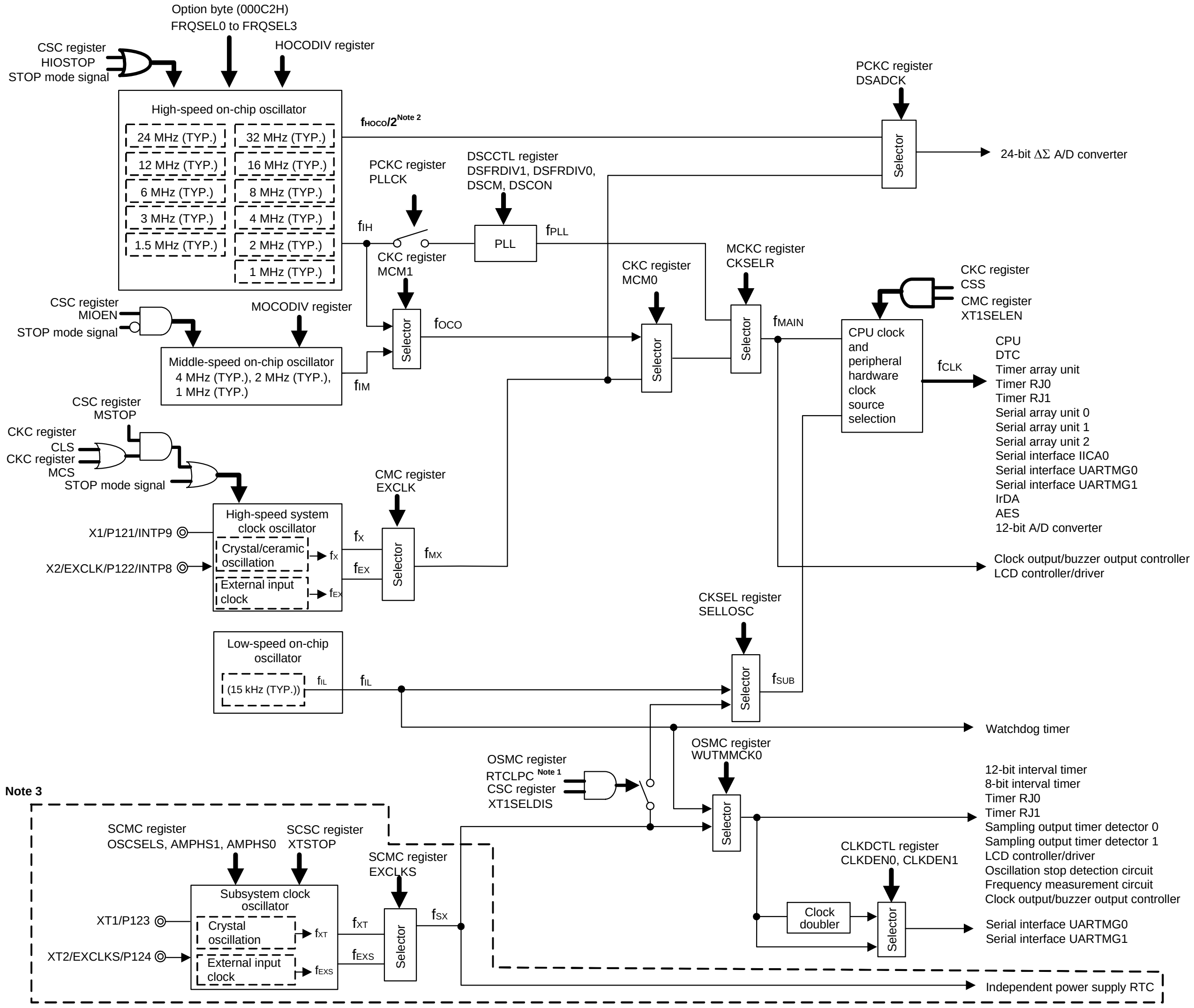
6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 6-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2) Subsystem clock supply option control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) Subsystem clock select register (CKSEL) Middle-speed on-chip oscillator frequency select register (MOCODIV) Frequency measurement clock select register (FMCKS) Peripheral clock control register (PCKC) PLL control register (DSCCTL) Main clock control register (MCKC) Sub clock operation mode control register (SCMC) Sub clock operation status control register (SCSC)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Middle-speed on-chip oscillator Low-speed on-chip oscillator PLL oscillator

Figure 6-1. Block Diagram of Clock Generator



Notes 1. Setting the RTCLPC bit to 1 during a period in the STOP mode or in the HALT mode while the CPU is being driven by the sub clock (f_{SX}) stops supply of the clock signal.

2. To supply the high-speed on-chip oscillator clock as the operating clock for the 24-bit ΔΣ A/D converter, set the FRQSEL3 bit to 0 (f_{HOCO} = 24 MHz). $f_{HOCO}/2$ = 12 MHz is supplied regardless of the frequency setting of the high-speed on-chip oscillator clock (f_{IH}).

3. The blocks in the dotted lines run on the VRTC power-supply. They can operate after release from the RTC power-on reset following the power supply to the VRTC pin being turned on. If the voltage on the VRTC pin falls below the detection voltage (V_{PDR}), an RTC power-on reset is generated and the blocks in the dotted lines stop.

Caution **The low-speed on-chip oscillator clock (f_{IL}) cannot be selected as the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1. When the serial interface UARTMG0 or UARTMG1, sampling output timer detector 0, or sampling output timer detector 1 is to be used, select the sub clock (f_{SX}) as the operating clock by setting the WUTMMCK0 bit to 0.**

(**Remark** is listed on the next page.)

Remark f_{X}: X1 clock oscillation frequency

f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)

f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)

f_{EX}: External main system clock frequency

f_{MX}: High-speed system clock frequency

f_{MAIN}: Main system clock frequency

f_{XT}: XT1 clock oscillation frequency

f_{EXS}: External subsystem clock frequency

f_{SX}: Sub clock frequency

f_{SUB}: Subsystem clock frequency

f_{CLK}: CPU/peripheral hardware clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

f_{PLL}: PLL clock frequency

f_{HOCO}: High-speed on-chip oscillator clock oscillation frequency (24 MHz when FRQSEL3 = 0 and 32 MHz when FRQSEL3 = 1)

6.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Sub clock operation mode control register (SCMC)
- Sub clock operation status control register (SCSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)
- Subsystem clock supply option control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Subsystem clock select register (CKSEL)
- Middle-speed on-chip oscillator frequency select register (MOCODIV)
- Frequency measurement clock select register (FMCKS)
- PLL control register (DSCCTL)
- Main clock control register (MCKC)
- Peripheral clock control register (PCKC)

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

6.3.1 Clock operation mode control register (CMC)

This register is used to set the operating mode of the X1/P121/INTP9 and X2/EXCLK/P122/INTP8 pins, to select the gain of the X1 oscillator, and to permit or prohibit selection of the XT1 oscillation clock or external subsystem clock.

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.

Generation of reset signals other than an RTC power-on reset clears this register to 00H.

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

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	XT1SELEN	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121/INTP9 pin	X2/EXCLK/P122/INTP8 pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

XT1SELEN	Permits or prohibits selection of the XT1 oscillation clock (f_{XT}) or external subsystem clock (f_{EXS}) as the CPU/peripheral hardware clock (f_{CLK})Notes 1, 2, 3
0	Prohibited (switching the clock by setting the CSS bit in the CKC register is disabled).
1	Permitted (switching the clock by setting the CSS bit in the CKC register is enabled).

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ f_X ≤ 10 MHz
1	10 MHz < f_X ≤ 20 MHz

Notes 1. This bit only permits switching the clock by setting the CSS bit in the CKC register. Simply setting this bit does not change the CPU/peripheral hardware clock (f_{CLK}).

2. Setting this bit is not required if the low-speed on-chip oscillator clock (f_{IL}) is selected as the CPU/peripheral hardware clock (f_{CLK}).

3. Be sure to write the same value as that of the OSCSELS bit in the SCMC register.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.

2. After reset release, set the CMC register before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC) or the sub clock operation status control register (SCSC).

3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.

4. Specify the settings for the AMPH bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{SUB}).

5. Although the maximum system clock frequency is 32 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Remark f_X: X1 clock frequency

6.3.2 System clock control register (CKC)

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

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

Generation of reset signals other than an RTC power-on reset clears this register to 00H.

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

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

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

MCS	Status of Main system clock (f_{MAIN})
0	Main on-chip oscillator clock (f_{OCO})
1	High-speed system clock (f_{MX})

MCM0[Note 2]	Main system clock (f_{MAIN}) operation control
0	Selects the main on-chip oscillator clock (f_{OCO}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

MCS1	Status of Main on-chip oscillator clock (f_{OCO})
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

MCM1[Note 2]	Main on-chip oscillator clock (f_{OCO}) operation control
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

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

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

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})

Cautions 1. Be sure to set bits 2 and 3 of the CKC register to 0.

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 with independent power supply, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, timers RJ0 and RJ1, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the 12-bit A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 43 ELECTRICAL SPECIFICATIONS.

6.3.3 Clock operation status control register (CSC)

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

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

Generation of reset signals other than an RTC power-on reset sets this register to C0H.

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

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
CSC	MSTOP	XT1SELDIS	0	0	0	0	MIOEN	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XT1SELDIS	Control supply of the sub clock (f_{SX})[Note 3] as the CPU/peripheral hardware clock (f_{CLK})[Notes 1, 2]
0	Enables clock supply.
1	Stops clock supply.

MIOEN	Middle-speed on-chip oscillator clock operation control
0	Middle-speed on-chip oscillator stopped
1	Middle-speed on-chip oscillator operating

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator operating
1	High-speed on-chip oscillator stopped

Notes 1. This bit only controls supply of the sub clock (f_{SX}). It does not control oscillation of the XT1 oscillator.

2. Be sure to write 0 to this bit when the XT1 oscillation clock (f_{XT}) or external subsystem clock (f_{EXS}) is to be used as the CPU/peripheral hardware clock (f_{CLK}).

3. This does not include supply of the clock signal to the realtime clock with independent power supply, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

(**Cautions** are listed on the next page.)

Cautions 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.

2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

3. To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

4. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) with the OSC register.

5. The preconditions for stopping oscillation of the various clocks (and for disabling the external clock inputs) and the register settings to be made to stop oscillation of each clock or disable input of the given clock are listed in Table 6-2.

Before stopping the oscillation of a clock, check that the precondition for stopping clock oscillation is satisfied.

Table 6-2. Preconditions for Stopping Clock Oscillation and Register Settings

Clock	Precondition for Stopping Clock Oscillation (or Disabling External Clock Input)	Settings of the SFR Register
High-speed on-chip oscillator clock	MCS1 = 1, MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock or PLL clock.)	CSC.HIOSTOP = 1
Middle-speed on-chip oscillator clock	MCS1 = 0, MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the middle-speed on-chip oscillator clock.)	CSC.MIOEN = 0
PLL clock	CKSTR = 0 (The CPU is operating on a clock other than the PLL clock.)	DSCCTL.DSCON = 0
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	CSC.MSTOP = 1
External main system clock		

XT1 clock	CLS = 0
	(The CPU is operating on a clock other than the subsystem clock.)
		SCSC.XTSTOP = 1
		CMC.XT1SELEN = 0
		CSC.XT1SELDIS = 1

External subsystem clock		
Low-speed on-chip oscillator clock	CLS = 0 (The CPU is operating on a clock other than the low-speed on-chip oscillator clock.)	CKSEL.SELLOSC = 0

6.3.4 Sub clock operation mode control register (SCMC)

This register is used to set the operating mode of the XT1/P123 and XT2/EXCLKS/P124 pins, and to select the gain of the oscillator.

After release from an RTC power-on reset or a reset from any other source, the SCMC register can be written only once by an 8-bit memory manipulation instruction. This register can be read by an 8-bit memory manipulation instruction.

The SCMC register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal clears this register to 00H. This register is not reset by other reset sources (including the power-on reset of the V_{DD} power supply).

Figure 6-5. Format of Sub Clock Operation Mode Control Register (SCMC)

Address: F0384H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SCMC	0	0	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	0

EXCLKS[Note]	OSCSELS[Note]	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	
0	1	XT1 oscillation mode	Crystal oscillator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1[Note]	AMPHS0[Note]	XT1 oscillator oscillation mode selection
0	0	Low-power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

Note The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are only initialized by an RTC power-on reset; they retain their values following a reset due to another source (including the power-on reset of the V_{DD} power supply).

Cautions 1. After the CPU is released from the reset state, the SCMC register can be written only once by an 8-bit memory manipulation instruction. When using the SCMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop.

2. After the CPU is released from the reset state, set the SCMC register before XT1 oscillation is started as set by the sub clock operation status control register (SCSC).

3. Specify the settings for the AMPHS1 and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

4. Count the f_{XT} oscillation stabilization time by using software.

5. After the CPU is released from the reset state following writing to the SCMC register and then a reset other than an RTC power-on reset, set the same value as the value before the reset to prevent incorrect operation in the case of an endless loop or runaway execution.

Cautions 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 the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 6.7 Resonator and Oscillator Constants. Using ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is not recommended for applications (e.g. utility meters) which require securing wide margins for oscillation. In such cases, we recommend the use of normal oscillation (AMPHS1, AMPHS0 = 0, 1).**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) 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 potential 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 resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high 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.**

7. The XT1 clock oscillator runs on the VRTC power-supply. It can operate after release from the RTC power-on reset following the power supply to the VRTC pin being turned on.

8. Be sure to clear bits 7, 6, 3, and 0 to 0.

6.3.5 Sub clock operation status control register (SCSC)

This register is used to control the operation of the sub clock.

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

The SCSC register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal sets this register to 40H.

Figure 6-6. Format of Sub Clock Operation Status Control Register (SCSC)

Address: F0386H After reset: 40H R/W

Symbol	7	<6>	5	4	3	2	1	0
SCSC	0	XTSTOP	0	0	0	0	0	0

XTSTOP[Note]	Control of XT1 oscillator operation
0	XT1 oscillation mode: XT1 oscillator operating External clock input mode: External clock from EXCLKS pin is valid. Input port mode: Input port
1	XT1 oscillation mode: XT1 oscillator stopped External clock input mode: External clock from EXCLKS pin is invalid. Input port mode: Input port

Note The XTSTOP bit is only initialized by an RTC power-on reset; it retains its value following a reset due to another source (including the power-on reset of the V_{DD} power supply).

Cautions 1. When starting XT1 oscillation by setting the XTSTOP bit, use software to wait for oscillation of the sub clock to become stable.

2. Be sure to clear the bits 7 and 5 to 0 to 0.

6.3.6 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case,

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

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

The generation of reset signal, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

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

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

Figure 6-7. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
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.

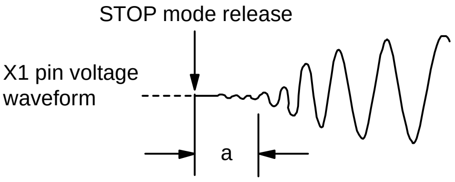
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

6.3.7 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time.

When the X1 clock is made to oscillate by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

When switching the CPU clock from the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock to the X1 clock, and when using the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

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

Reset signal generation sets the OSTS register to 07H.

Figure 6-8. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				fX = 10 MHz	fX = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204 μs	102 μs
1	0	0	$2^{13}/f_X$	819 μs	409 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.63 ms
1	1	0	$2^{17}/f_X$	13.1 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.2 ms	13.1 ms

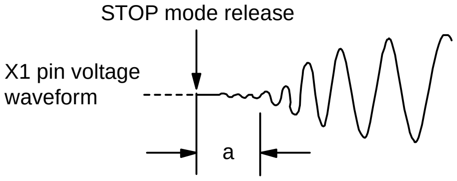
Cautions 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fX: X1 clock oscillation frequency

6.3.8 Subsystem clock select register (CKSEL)

The CKSEL register is used to select the sub clock or low-speed on-chip oscillator clock as the subsystem clock.

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

Reset signal generation sets this register to 00H.

Figure 6-9. Format of Subsystem Clock Select Register (CKSEL)

Address: FFFA7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
CKSEL	0	0	0	0	0	0	0	SELLOSC

SELLOSC	Selection of subsystem clock (f_{SUB})
0	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL})[Note]

Note Do not set SELLOSC to 1 when the sub clock (f_{SX}) operates.

Caution When changing SELLOSC, be sure to set CSS to 0 (f_{MAIN} selected) and change the value of SELLOSC while CLS is 0.

6.3.9 Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)

These registers are used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

To use the peripheral functions below, which are controlled by these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Timer array unit
- Serial array unit 0
- Serial array unit 1
- Serial array unit 2
- Serial interface IICA0
- 12-bit A/D converter
- IrDA
- 24-bit ΔΣ A/D converter
- DTC
- Frequency measurement circuit
- Independent power supply real-time clock
- 32-bit multiplier and accumulator
- Oscillation stop detection circuit
- 12-bit interval timer
- Serial interface UARTMG0
- Serial interface UARTMG1
- Sampling output timer detector 0
- Sampling output timer detector 1
- Timer RJ0
- Timer RJ1

The PER0, PER1, and PER2 registers can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 6-10. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

IRDAEN	Control of IrDA input clock supply
0	Stops input clock supply. ● SFR used by IrDA cannot be written. The read value is 00H. However, the SFR is not initialized. Note 1
1	Enables input clock supply. ● SFR used by IrDA can be read and written.

ADCEN	Control of 12-bit A/D converter input clock supply
0	Stops input clock supply. ● SFR used by the 12-bit A/D converter cannot be written. The read value is 00H. However, the SFR is not initialized. Note 2
1	Enables input clock supply. ● SFR used by the 12-bit A/D converter can be read and written.

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. ● SFR used by the serial interface IICA0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 3
1	Enables input clock supply. ● SFR used by the serial interface IICA0 can be read and written.

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. ● SFR used by the serial array unit 1 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 4
1	Enables input clock supply. ● SFR used by the serial array unit 1 can be read and written.

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. ● SFR used by the serial array unit 0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 5
1	Enables input clock supply. ● SFR used by the serial array unit 0 can be read and written.

SAU2EN	Control of serial array unit 2 input clock supply
0	Stops input clock supply. ● SFR used by the serial array unit 2 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 6
1	Enables input clock supply. ● SFR used by the serial array unit 2 can be read and written.

(**Notes** and **Cautions** are listed on the next page.)

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. ● SFR used by the timer array unit 0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 7
1	Enables input clock supply. ● SFR used by the timer array unit 0 can be read and written.

Notes 1. To initialize the IrDA and the SFR used by the IrDA, use bit 6 (IRDARES) of PRR0.

2. To initialize the 12-bit A/D converter and the SFR used by the 12-bit A/D converter, use bit 5 (ADCRES) of PRR0.

3. To initialize the serial interface IICA0 and the SFR used by the serial interface IICA0, use bit 4 (IICA0RES) of PRR0.

4. To initialize the serial array unit 1 and the SFR used by the serial array unit 1, use bit 3 (SAU1RES) of PRR0.

5. To initialize the serial array unit 0 and the SFR used by the serial array unit 0, use bit 2 (SAU0RES) of PRR0.

6. To initialize the serial array unit 2 and the SFR used by the serial array unit 2, use bit 1 (SAU2RES) of PRR0.

7. To initialize the timer array unit 0 and the SFR used by the timer array unit 0, use bit 0 (TAU0RES) of PRR0.

Cautions 1. Be sure to clear bit 7 to 0.

2. Do not change the target bit in the PER0 register while operation of each peripheral function is enabled. Change the setting specified by PER0 while operation of each peripheral function assigned to PER0 is stopped.

Figure 6-11. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

FMCEN	Control of frequency measurement circuit input clock supply
0	Stops input clock supply. ● SFR used by the frequency measurement circuit cannot be written. The read value is 00H. ● The frequency measurement circuit is in the reset state.
1	Enables input clock supply. ● SFR used by the frequency measurement circuit can be read and written.

SMOTD1EN	Control of sampling output timer detector 1 input clock supply
0	Stops input clock supply. ● SFR used by the sampling output timer detector 1 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 1
1	Enables input clock supply. ● SFR used by the sampling output timer detector 1 can be read and written.

SMOTD0EN	Control of sampling output timer detector 0 input clock supply
0	Stops input clock supply. ● SFR used by the sampling output timer detector 0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 2
1	Enables input clock supply. ● SFR used by the sampling output timer detector 0 can be read and written.

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. ● DTC cannot run.
1	Enables input clock supply. ● DTC can run.

TRJ1EN	Control of timer RJ1 input clock supply
0	Stops input clock supply. ● SFR used by the timer RJ1 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 3
1	Enables input clock supply. ● SFR used by the timer RJ1 can be read and written.

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. ● SFR used by the timer RJ0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 4
1	Enables input clock supply. ● SFR used by the timer RJ0 can be read and written.

(**Notes** and **Cautions** are listed on the next page.)

DSADCEN	Control of 24-bit ΔΣ A/D converter input clock supply
0	Stops input clock supply. ● SFR used by the 24-bit ΔΣ A/D converter cannot be written. The read value is 00H. However, the SFR is not initialized. Note 5
1	Enables input clock supply. ● SFR used by the 24-bit ΔΣ A/D converter can be read and written.

Notes 1. To initialize the sampling output timer detector 1 and the SFR used by the sampling output timer detector 1, use bit 5 (SMOTD1RES) of PRR1.

2. To initialize the sampling output timer detector 0 and the SFR used by the sampling output timer detector 0, use bit 4 (SMOTD0RES) of PRR1.

3. To initialize the timer RJ1 and the SFR used by the timer RJ1, use bit 2 (TRJ1RES) of PRR1.

4. To initialize the timer RJ0 and the SFR used by the timer RJ0, use bit 1 (TRJ0RES) of PRR1.

5. To initialize the 24-bit ΔΣ A/D converter and the SFR used by the 24-bit ΔΣ A/D converter, use bit 0 (DSADRES) of PRR1.

Cautions 1. Be sure to clear bit 7 to 0.

2. Do not change the target bit in the PER1 register while operation of each peripheral function is enabled. Change the setting specified by PER1 while operation of each peripheral function assigned to PER1 is stopped.

Figure 6-12. Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. ● SFR used by the 12-bit interval timer cannot be written. The read value is 00H. However, the SFR is not initialized. Note 1
1	Enables input clock supply. ● SFR used by the 12-bit interval timer can be read and written.

OSDCEN	Control of oscillation stop detection circuit input clock supply
0	Stops input clock supply. ● SFR used by the oscillation stop detection circuit cannot be written. The read value is 00H. However, the SFR is not initialized. Note 2
1	Enables input clock supply. ● SFR used by the oscillation stop detection circuit can be read and written.

UARTMG1EN	Control of serial interface UARTMG1 input clock supply
0	Stops input clock supply. ● SFR used by the serial interface UARTMG1 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 3
1	Enables input clock supply. ● SFR used by the serial interface UARTMG1 can be read and written.

UARTMG0EN	Control of serial interface UARTMG0 input clock supply
0	Stops input clock supply. ● SFR used by the serial interface UARTMG0 cannot be written. The read value is 00H. However, the SFR is not initialized. Note 4
1	Enables input clock supply. ● SFR used by the serial interface UARTMG0 can be read and written.

(**Notes** and **Cautions** are listed on the next page.)

MACEN	Control of 32-bit multiplier and accumulator input clock supply
0	Stops input clock supply. ● SFR used by the 32-bit multiplier and accumulator cannot be written. The read value is 00H. However, the SFR is not initialized. Note 5
1	Enables input clock supply. ● SFR used by the 32-bit multiplier and accumulator can be read and written.

VRTCEN	Control of independent power supply RTC input clock supply
0	Stops input clock supply. ● SFR used by the independent power supply RTC cannot be written. The read value is 00H. The independent power supply RTC is able to operate with the sub clock (f_{SX}) as the source for counting.
1	Enables input clock supply. ● SFR used by the independent power supply RTC can be read and written. Note 6

Notes 1. To initialize the 12-bit interval timer and the SFR used by the 12-bit interval timer, use bit 7 (TMKARES) of PRR2.

2. To initialize the oscillation stop detection circuit and the SFR used by the oscillation stop detection circuit, use bit 6 (OSDCRES) of PRR2.

3. To initialize the serial interface UARTMG1 and the SFR used by the serial interface UARTMG1, use bit 5 (UARTMG1RES) of PRR2.

4. To initialize the serial interface UARTMG0 and the SFR used by the serial interface UARTMG0, use bit 4 (UARTMG0RES) of PRR2.

5. To initialize the 32-bit multiplier and accumulator and the SFR used by the 32-bit multiplier and accumulator, use bit 2 (MACRES) of PRR2.

6. Set the VRTCEN bit to 0 except during reading or writing of the SFRs of the independent power supply RTC.

Cautions 1. Be sure to clear the following bits to 0.
Bits 1 and 3

2. Do not change the target bit in the PER2 register while operation of each peripheral function is enabled. Change the setting specified by PER2 while operation of each peripheral function assigned to PER2 is stopped.

6.3.10 Subsystem clock supply option control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 is stopped in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock.

In addition, the OSMC register can be used to select the operating clock of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

The low-speed on-chip oscillator clock cannot be selected as the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1. When the serial interface UARTMG0 or UARTMG1, sampling output timer detector 0, or sampling output timer detector 1 is to be used, select the sub clock (f_{SX}) as the operating clock by setting the WUTMMCK0 bit to 0.

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

Reset signal generation clears this register to 00H.

Figure 6-13. Format of Subsystem Clock Supply Option Control Register (OSMC)

Address: F00F3H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC[Note 4]	Setting in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock
0	Enables supply of sub clock (f_{SX}) to peripheral functions (See **Tables 29-1** to **29-3** for peripheral functions whose operations are enabled.)
1	Stops supply of sub clock (f_{SX}) to peripheral functions other than the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL}) [Notes 2, 3, 6, 7]	Low-speed on-chip oscillator clock (f_{IL}) selected[Note 6]	Clock output is prohibited. [Note 5]	Setting prohibited

Notes 1. Be sure to set bits 0 to 3, 5, and 6 to 0.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (f_{IL}) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the sub clock (f_{SX}) 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.

5. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

6. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

7. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

6.3.11 High-speed on-chip oscillator frequency select register (HOCODIV)

The frequency of the high-speed on-chip oscillator which is set by an option byte (000C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL3 bit of the option byte (000C2H).

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

Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 6-14. Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H After reset: the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) 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 high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1[Note 2]
0	0	0	f_{IH} = 24 MHz[Note 1]	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	f_{IH} = 1.5 MHz	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

Notes 1. When 32 MHz is selected as the frequency of the CPU/peripheral hardware clock (f_{CLK}) while the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter (the DSADCK bit of the PCKC register is set to 0), set the high-speed on-chip oscillator clock (f_{IH}) to 24 MHz and select the PLL clock (at 32 MHz).

2. When the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter (the DSADCK bit in the PCKC register is set to 0), the 24-bit ΔΣ A/D converter cannot be used.

Cautions 1. Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 6 MHz	2.1 to 5.5 V
			1 to 16 MHz	2.4 to 5.5 V
			1 to 32 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

3. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

6.3.12 Middle-speed on-chip oscillator frequency select register (MOCODIV)

The MOCODIV register is used to select the division ratio of the middle-speed on-chip oscillator.

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

Reset signal generation clears this register to 00H.

Figure 6-15. Format of Middle-speed On-chip Oscillator Frequency Select Register (MOCODIV)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MOCODIV	0	0	0	0	0	0	MOCODIV1	MOCODIV0

MOCODIV1	MOCODIV0	Selection of middle-speed on-chip oscillator clock frequency
0	0	4 MHz
0	1	2 MHz
1	0	1 MHz
Other than above		Setting prohibited

6.3.13 Frequency measurement clock select register (FMCKS)

The FMCKS register is used to select the operating clock and frequency count clock to be input to the frequency measurement circuit.

The FMCKS register can be used by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears the FMCKS register to 00H.

Figure 6-16. Format of Frequency Measurement Clock Select Register (FMCKS)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FMCKS	0	0	0	0	0	0	FMCKSEL1	FMCKSEL0

FMCKSEL1	FMCKSEL0	Selection of frequency count clock
0	0	f_{MX} selected
0	1	f_{IM} selected
1	×	f_{IH} selected

6.3.14 PLL control register (DSCCTL)

This register is used to control operation of the PLL oscillator.

To select 32 MHz for the CPU/peripheral hardware clock (f_{CLK}) while the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter (the DSADCK bit in the PCKC register is set to 0), divide the high-speed on-chip oscillator clock (f_{IH} = 24 MHz) by 6 to obtain the 4-MHz PLL reference clock. After that, select multiplication by 16 and division by 2 for the PLL (multiplication by 8) to obtain a 32-MHz clock signal from the PLL (f_{PLL}).

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

Reset signal generation clears this register to 00H.

Figure 6-17. Format of PLL Control Register (DSCCTL)

Address: F02E5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	DSFRDIV1	DSFRDIV0	DSCM	DSCON

DSFRDIV1	DSFRDIV0	Control over frequency division of the PLL reference clock[Notes 3, 4]
1	1	Frequency division by 6 (f_{IH}/6)
Other than above		Setting prohibited

DSCM	Selection of multiplication by the PLL[Note 1]
1	Multiplication by 16 and division by 2 (multiplication by 8)
Other than above	Setting prohibited

DSCON	Control of PLL oscillation and output[Note 5]
0	PLL stops
1	PLL oscillates and the result is output[Note 2]

Notes 1. The multiplier in parentheses "()" is because of division by 2 at the last stage of the PLL oscillator.

2. After the PLL starts operation, at least 40 μs of waiting for lock-up is required for the frequency to become stable.

3. Do not change the division setting of the PLL reference clock while the PLL is operating. Stop the PLL before changing the setting.

4. The only signal used as the PLL reference clock is f_{IH}, which runs at 24 MHz.

5. When making a transition to the STOP mode, switch the CPU/peripheral hardware clock (f_{CLK}) to the high-speed on-chip oscillator clock (f_{IH} = 24 MHz) and stop the PLL.

Caution Be sure to clear bits 4 to 7 to 0.

6.3.15 Main clock control register (MCKC)

This register is used to control the operation of the main clock.

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

Reset signal generation clears this register to 00H.

Figure 6-18. Format of Main Clock Control Register (MCKC)

Address: F02E6H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
MCKC	CKSTR	0	0	0	0	0	0	CKSELR

CKSTR	State of switching the main system clock (f_{MAIN})
0	On-chip oscillator clock (f_{OCO})/high-speed system clock (f_{MX})[Note 2]
1	PLL clock (f_{PLL})

CKSELR	Selection of the main system clock (f_{MAIN})
0	On-chip oscillator clock (f_{OCO})/high-speed system clock (f_{MX})[Note 2]
1	PLL clock (f_{PLL})[Note 3]

Notes 1. Bit 7 is read only.

2. Select with the bit 4 (MCM0) of the system clock control register (CKC).

3. The high-speed on-chip oscillator clock must not be stopped while the PLL is selected.

Caution Be sure to clear bits 1 to 6 to 0.

Remark Because the MCM0 bit is given priority in the case of the conflict between MCM0 and CKSELR bits, the main system clock is selected.

6.3.16 Peripheral clock control register (PCKC)

This register is used to select the peripheral clock (operating clock for the 24-bit ΔΣ A/D converter or operating clock for the PLL).

When this is used as the operating clock for the 24-bit ΔΣ A/D converter, only the 12-MHz crystal resonator can be used as the high-speed system clock (f_{MX}).

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

Reset signal generation clears this register to 00H.

Figure 6-19. Format of Peripheral Clock Control Register (PCKC)

Address:F0098H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PCKC	0	0	0	0	0	0	PLLCK	DSADCK

PLLCK	Control of operating clock supplied to the PLL
0	Stops supply of the high-speed on-chip oscillator clock (f_{IH}).
1	Enables supply of the high-speed on-chip oscillator clock (f_{IH}).

DSADCK	Selection of operating clock for the 24-bit ΔΣ A/D converter
0	Enables supply of the high-speed on-chip oscillator clock (f_{HOCO}/2)[Notes 2, 3] (stops supply of f_{MX}).
1	Enables supply of the high-speed system clock (f_{MX}).[Note 1]

Notes 1. Only the 12-MHz crystal resonator can be used as the high-speed system clock (f_{MX}).

2. When the high-speed on-chip oscillator clock (f_{HOCO}/2) is to be selected, set f_{IH} to 24, 12, 6, 3, or 1.5 MHz (FRQSEL3 = 0).

3. Even when the PLL clock (f_{PLL}) is selected as the CPU/peripheral hardware clock (f_{CLK}), the high-speed oscillator clock (f_{HOCO}/2 = 12 MHz) is supplied as the operating clock for the 24-bit ΔΣ A/D converter.

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.

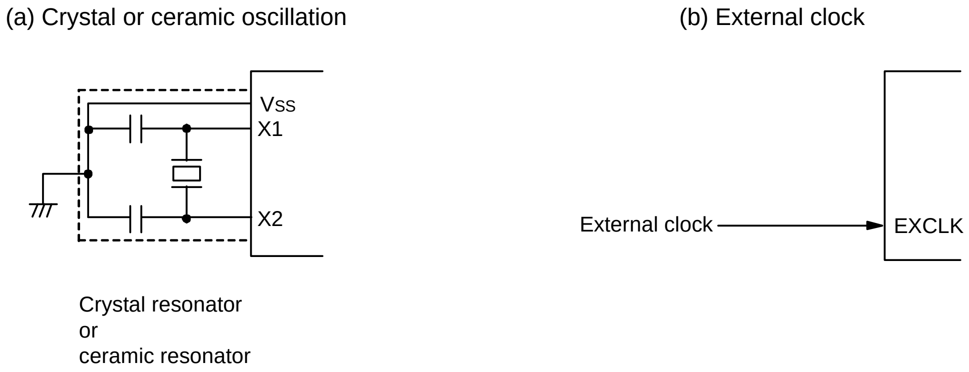
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins**.

Figure 6-20 shows an example of the external circuit of the X1 oscillator.

Figure 6-20. Example of External Circuit of X1 Oscillator



Caution is listed on the next page.

6.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 or 38.4 kHz (TYP.)) connected to the XT1 and XT2 pins.

The XT1 clock oscillator runs on the VRTC power-supply. To use the XT1 oscillator, after turning on the power supply to the VRTC pin and release from the RTC power-on reset, set both bit 4 (OSCSELS) of the sub clock operation mode control register (SCMC) and bit 4 (XT1SELEN) 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, after turning on the power supply to the VRTC pin and release from the RTC power-on reset, set bits 5 and 4 (EXCLKS, OSCSELS) of the sub clock operation mode control register (SCMC) and bit 4 (XT1SELEN) of the clock operation mode control register (CMC) as follows.

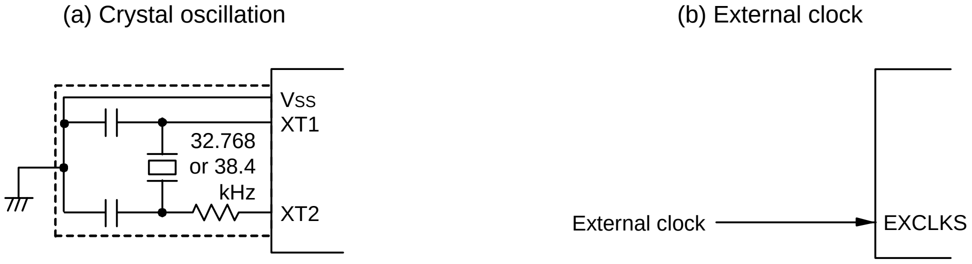
- Crystal oscillation: EXCLKS, OSCSELS = 0, 1, XT1SELEN = 1
- External clock input: EXCLKS, OSCSELS = 1, 1, XT1SELEN = 1

When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins**.

Figure 6-21 shows an example of the external circuit of the XT1 oscillator.

Figure 6-21. Example of External Circuit of XT1 Oscillator



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6-20 and 6-21 to avoid an adverse effect from wiring capacitance.

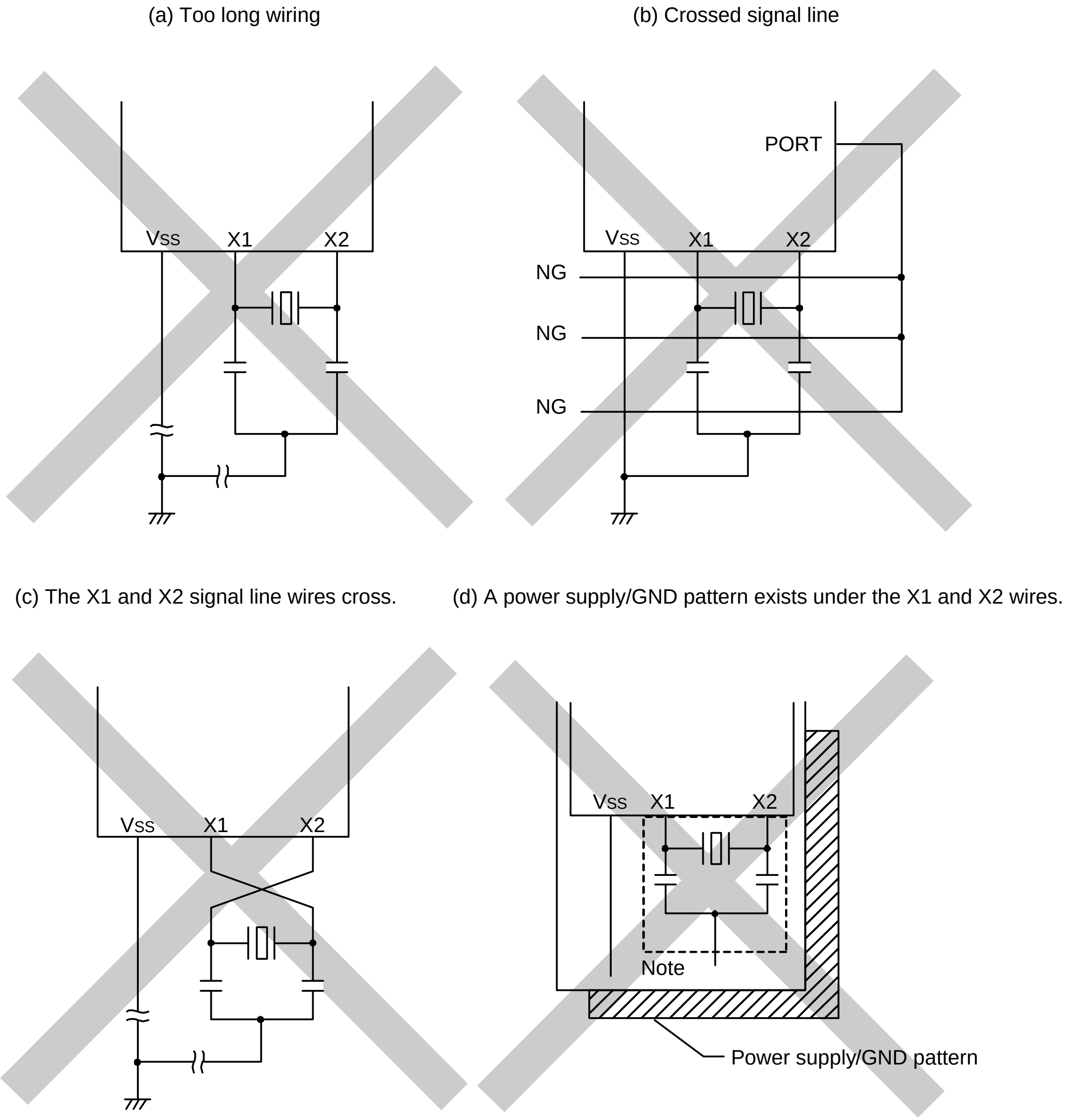
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 6.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high 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 6-22 shows examples of incorrect resonator connection.

Figure 6-22. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

(b) Crossed signal line

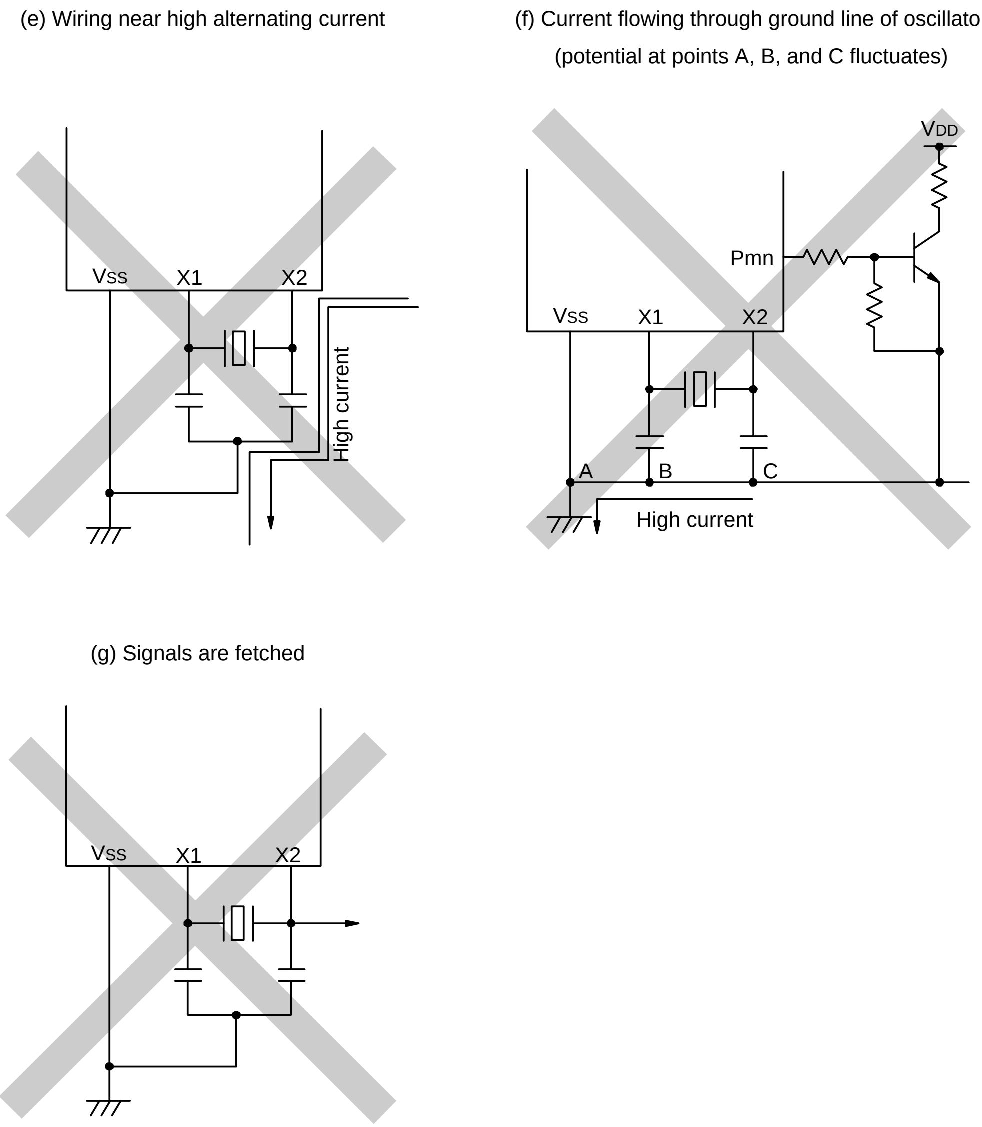
(c) The X1 and X2 signal line wires cross.

(d) A power supply/GND pattern exists under the X1 and X2 wires.

Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators 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.

Figure 6-22. Examples of Incorrect Resonator Connection (2/2)



(e) Wiring near high alternating current

(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)

(g) Signals are fetched

Caution **When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.**

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively.

6.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/I1C (512 KB). The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, 1.5, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after reset release.

6.4.4 Middle-speed on-chip oscillator

The middle-speed on-chip oscillator is incorporated in the RL78/I1C (512 KB). 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/I1C (512 KB).

The low-speed on-chip oscillator runs while the watchdog timer is operating or when the setting of either or both of the following bits is 1: bit 4 (WUTMMCK0) in the subsystem clock supply mode control register (OSMC) or bit 0 (SELLOSC) in the subsystem clock select register (CKSEL).

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and both WUTMMCK0 and SELLOSC are set to 0.

6.4.6 Phase-locked loop (PLL)

The RL78/I1C (512 KB) incorporates a PLL circuit.

The high-speed on-chip oscillator clock (f_{IH} = 24 MHz) is divided by 6 to obtain the 4-MHz PLL reference clock and then multiplied by 16 and divided by 2 (multiplied by 8) to obtain a 32-MHz clock signal from the PLL (f_{PLL}).

Operation is controlled by bit 0 (DSCON) in the PLL control register (DSCCTL).

Caution When making a transition to the STOP mode, switch to the high-speed on-chip oscillator clock and stop the PLL. Do not stop the high-speed on-chip oscillator clock while the PLL is in use.

6.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6-1**).

CPU/peripheral hardware clock f_{CLK}

(1) Main system clock f_{MAIN}

(1-1) High-speed system clock f_{MX}

- X1 clock f_X
- External main system clock f_{EX}

(1-2) Main on-chip oscillator clock f_{OCO}

- High-speed on-chip oscillator clock f_{IH}
- Middle-speed on-chip oscillator clock f_{IM}

(1-3) PLL clock f_{PLL}

(2) Subsystem clock f_{SUB}

(2-1) Sub clock f_{SX}

- XT1 clock f_{XT}
- External subsystem clock f_{EXS}

(2-2) Low-speed on-chip oscillator clock f_{IL}

Clock for the peripheral hardware

(1) Clock for the watchdog timer

- Low-speed on-chip oscillator clock f_{IL}

(2) Clock for the realtime clock with independent power supply, sampling output timer detector 0, sampling output timer detector 1, and serial interfaces UARTMG0 and UARTMG1

- Sub clock f_{SX}

(3) Clock for the 12-bit interval timer, 8-bit interval timer, oscillation stop detection circuit, frequency measurement circuit, and timers RJ0 and RJ1

- Sub clock f_{SX}
- Low-speed on-chip oscillator clock f_{IL}

(4) Operating clock for the 24-bit ΔΣ A/D converter

- High-speed on-chip oscillator clock $f_{HOCO}/2$
- High-speed system clock f_{MX}

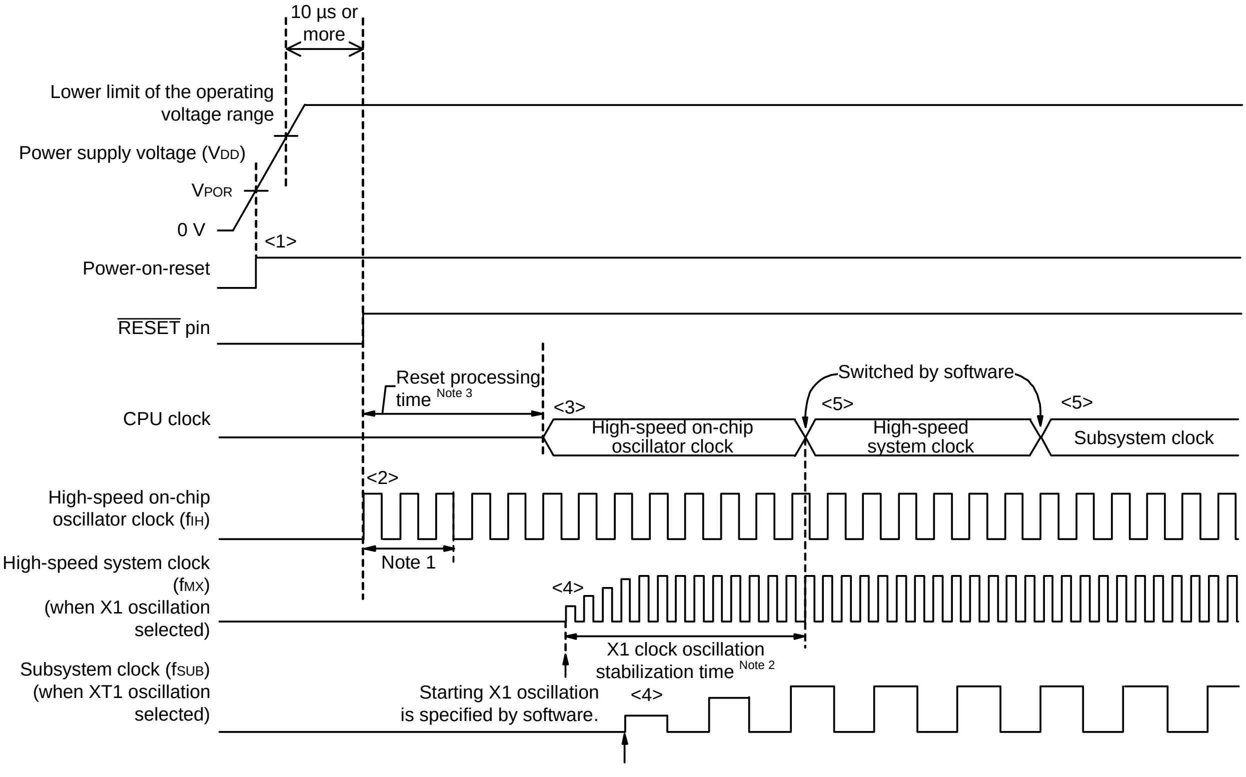
(5) Clock for the LCD controller/driver and clock output/buzzer output controller

- Main system clock f_{MAIN}
- Sub clock f_{SX}
- Low-speed on-chip oscillator clock f_{IL}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/I1C (512 KB).

When the power supply voltage is turned on, the clock generator operation is shown in **Figure 6-23**.

Figure 6-23. Clock Generator Operation When Power Supply Voltage Is Turned On


10 µs or more
Lower limit of the operating voltage range
Power supply voltage (VDD)
VPOR
0 V
<1>
Power-on-reset
RESET pin
CPU clock
Reset processing time Note 3
<3>
High-speed on-chip oscillator clock
Switched by software
<5>
High-speed system clock
<5>
Subsystem clock
<2>
High-speed on-chip oscillator clock (fIH)
Note 1
High-speed system clock (fMX) (when X1 oscillation selected)
<4>
X1 clock oscillation stabilization time Note 2
Subsystem clock (fSUB) (when XT1 oscillation selected)
Starting X1 oscillation is specified by software.
<4>

Starting XT1 oscillation is specified by software.

<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 **43.4 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 on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on chip oscillator clock.

2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

3. For the reset processing time, see **CHAPTER 31 POWER-ON-RESET CIRCUIT**.

Cautions 1. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

2. The subsystem clock must be set after turning on the power supply to the VRTC pin and release from the RTC power-on reset.

6.6 Controlling Clock

6.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 32, 24, 16, 12, 8, 6, 4, 3, 2, 1.5, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). In addition, Oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	Setting of flash operation mode	
0	0	LV (low-voltage main) mode	V_{DD} = 1.6 V to 5.5 V @ 1 MHz to 4 MHz
1	0	LS (low-speed main) mode	V_{DD} = 1.8 V to 5.5 V @ 1 MHz to 8 MHz
1	1	HS (high-speed main) mode	V_{DD} = 2.1 V to 5.5 V @ 1 MHz to 6 MHz V_{DD} = 2.4 V to 5.5 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 32 MHz
Other than above		Setting prohibited	

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
0	1	0	0	1.5 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency	
			FRQSEL3 = $0^{\text{Note 2}}$	FRQSEL3 = $1^{\text{Note 2}}$
0	0	0	f_{IH} = 24 $\text{MHz}^{\text{Note 1}}$	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	f_{IH} =1.5 MHz	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

Notes 1. When 32 MHz is selected as the frequency of the CPU/peripheral hardware clock (f_{CLK}) while the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter (the DSADCK bit of the PCKC register is set to 0), set the high-speed on-chip oscillator clock (f_{IH}) to 24 MHz and select the PLL clock (at 32 MHz).

2. When the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter (the DSADCK bit in the PCKC register is set to 0), set the FRQSEL3 bit to 0.

6.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation 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 oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases where the f_X is equal to or more than 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	0	XT1SELEN 0	0	0	0	AMPH 0/1

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

Example: Setting values when a wait of at least 102 μs is set based on 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 (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	XT1SELDIS 1	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on 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 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	0	0	MCS1 0	MCM1 0

Caution Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 6 MHz	2.1 to 5.5 V
			1 to 16 MHz	2.4 to 5.5 V
			1 to 32 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

6.6.3 Example of setting XT1 oscillation clock

After release from the reset state (except that following the RTC power-on reset), the high-speed on-chip oscillator clock is always the initial CPU/peripheral hardware clock (f_{CLK}).

After turning on the power supply to the VRTC pin and release from the RTC power-on reset, the XT1 oscillator and RTC circuit can operate.

To subsequently change the CPU/peripheral hardware clock (f_{CLK}) to the XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply option control register (OSMC), sub clock operation mode control register (SCMC), clock operation mode control register (CMC), clock operation status control register (CSC), and sub clock operation status control register (SCSC), set the XT1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <10> below.

<1> Turn on the power supply to the VRTC pin, and release the RTC power-on reset.

<2> To operate the frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output, oscillation stop detection circuit, LCD controller/driver, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 with the sub clock (f_{SX}) (ultra-low current consumption) in STOP mode or in HALT mode while CPU is operating with the sub clock (f_{SX}), set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 0	0	0	0	0

<3> Set (1) the VRTCEN bit in the PER2 register to permit access to the SFRs in the VRTC power-supply domain.

<4> Set the EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits in the SCMC register, and set the XT1 oscillation mode and the gain for the XT1 oscillator.

	7	6	5	4	3	2	1	0
SCMC	0	0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<5> Set (1) the XT1SELEN bit in the CMC register to permit selection of the XT1 clock as the CPU clock.

<6> Clear (0) the XTSTOP bit of the SCSC register to start oscillation of the XT1 oscillator.

	7	6	5	4	3	2	1	0
SCSC	0	XTSTOP 0	0	0	0	0	0	0

<7> Use the timer function or another function to wait for oscillation of the XT1 oscillation clock to stabilize by using software.

<8> Clear (0) the XT1SELDIS bit of the CSC register to permit selection of the XT1 clock as the CPU/peripheral hardware clock.

<9> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as 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

<10> After completion of setting the SFRs in the VRTC power-supply domain, clear (0) the VRTCEN bit in the PER2 register to prohibit access to SFRs in the VRTC power-supply domain.

Caution Once the settings in steps <4>, <6>, and <7> are made, the values are retained except in case of an RTC power-on reset, so making the settings again will not be necessary unless this occurs.

6.6.4 Procedure for settings when the XT1 oscillator is not to be used as the CPU/peripheral hardware clock

[Register settings] Set the register in the order of <1> to <4> below.

<1> Turn on the power supply to the VRTC pin, and release the RTC power-on reset.

<2> Set the EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits in the SCMC register, and set the XT1 oscillation mode and the gain for the XT1 oscillator. When the XT1 clock is not to be used, set the SCMC register to 00H and set the XT1/P123 and XT2/EXCLKS/P124 pins to the input port mode.

<3> Clear (0) the XT1SELEN bit of the CMC register to prohibit selection of the XT1 clock as the CPU clock.

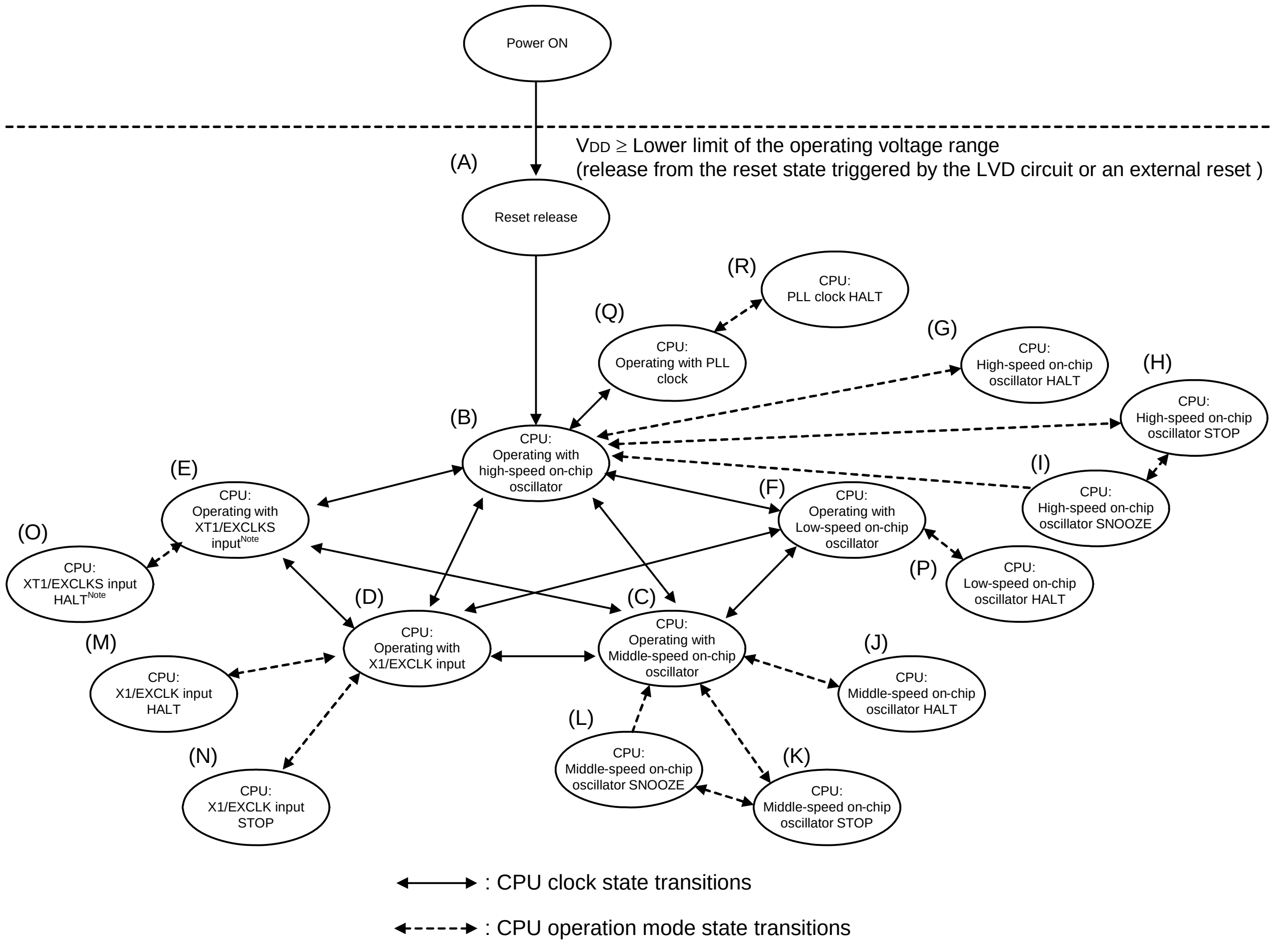
<4> Set (1) the XT1SELDIS bit of the CSC register to prohibit selection of the XT1 clock as the CPU/peripheral hardware clock.

Caution Once the settings in step <2> are made, the values are retained except in case of an RTC power-on reset, so making the settings again will not be necessary unless this occurs.

6.6.5 CPU clock status transition diagram

Figure 6-24 shows the CPU clock status transition diagram of this product.

Figure 6-24. CPU Clock Status Transition Diagram


Power ON
VDD ≥ Lower limit of the operating voltage range
(release from the reset state triggered by the LVD circuit or an external reset)
(A)
Reset release
(R)
CPU: PLL clock HALT
(Q)
CPU: Operating with PLL clock
(G)
CPU: High-speed on-chip oscillator HALT
(H)
CPU: High-speed on-chip oscillator STOP
(B)
CPU: Operating with high-speed on-chip oscillator
(I)
CPU: High-speed on-chip oscillator SNOOZE
(E)
CPU: Operating with XT1/EXCLKS input Note
(F)
CPU: Operating with Low-speed on-chip oscillator
(O)
CPU: XT1/EXCLKS input HALT Note
(P)
CPU: Low-speed on-chip oscillator HALT
(D)
CPU: Operating with X1/EXCLK input
(C)
CPU: Operating with Middle-speed on-chip oscillator
(M)
CPU: X1/EXCLK input HALT
(J)
CPU: Middle-speed on-chip oscillator HALT
(L)
CPU: Middle-speed on-chip oscillator SNOOZE
(N)
CPU: X1/EXCLK input STOP
(K)
CPU: Middle-speed on-chip oscillator STOP
: CPU clock state transitions
: CPU operation mode state transitions

Note Operation is possible after release from RTC power-on-reset.

Table 6-3 shows show transition of the CPU clock and examples of setting the SFR registers.

Table 6-3. CPU Clock Transition and SFR Register Setting Examples (1/5)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)
Target state transition: (A) → (B)

Clock After Change	SFR Register Setting
High-speed on-chip oscillator clock	SFR registers do not have to be set (default status after reset release).

(2) Changing to high-speed on-chip oscillator clock operation (B)
Target state transition: (C) → (B), (D) → (B), (E) → (B), (F) → (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
Clock After Change	HIOSTOP		CSS	MCM0	MCM1
High-speed on-chip oscillator clock	0	65 μs	0	0	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

Target state transition: (Q) → (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	MCKC Register	Waiting for the clocks to be switched	DSCCTL Register	PCKC Register
Clock After Change	CKSELR		DSCON	PLLCK
High-speed on-chip oscillator	0	Confirm that the CKSTR in the MCKC register is set to 0.	0	0

(3) Changing to middle-speed on-chip oscillator clock operation (C)
Target state transition: (B) → (C), (D) → (C), (E) → (C), (F) → (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
Clock After Change	MIOEN		CSS	MCM0	MCM1
Middle-speed on-chip oscillator clock	1	4 μs	0	0	1

Unnecessary if the CPU is operating with the middle-speed on-chip oscillator clock

Remark (A) to (R) in **table 6-3** correspond to (A) to (R) in **Figure 6-24**.

Table 6-3. CPU Clock Transition and SFR Register Setting Examples (2/5)

(4) Changing to PLL clock operation (Q)
Target state transition: (B) → (Q)

(Setting sequence of SFR registers)

Setting Flag of SFR Register / Clock After Change	DSCCTL Register			PCKC Register	DSCCTL Register
	DSFRDIV1	DSFRDIV0	DSCM	PLLCK	DSCON
PLL clock	1	1	1	1	1

Waiting time for lock-up	MCKC Register	Waiting for the clocks to be switched
	CKSELR	
40 µs	1	Confirm that the CKSTR in the MCKC register is set to 1.

Caution 24 MHz must be selected as the frequency of the high-speed on-chip oscillator clock (f_{IH}).

Remark (A) to (R) in **Table 6-3** correspond to (A) to (R) in **Figure 6-24**.

Table 6-3. CPU Clock Transition and SFR Register Setting Examples (3/5)

(5) Changing the CPU to high-speed system clock operation (D)
Target state transition: (B) → (D), (C) → (D), (E) → (D), (F) → (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Clock After Change	CMC Register[Note 1]			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
Changing to X1 clock: 1 MHz ≤ fx ≤ 10 MHz	0	1	0	**Note 2**	0	Must be checked	0	1
Changing to X1 clock: 10 MHz < fx ≤ 20 MHz	0	1	1	**Note 2**	0	Must be checked	0	1
Changing to external main clock	1	1	×	**Note 2**	0	Need not be checked	0	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Notes 1. The clock operation mode control register (CMC) can be changed only once after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 43 ELECTRICAL SPECIFICATIONS).

Remark (A) to (R) in **Table 6-3** correspond to (A) to (R) in **Figure 6-24**.

Table 6-3. CPU Clock Transition and SFR Register Setting Examples (4/5)

(6) Changing the CPU to subsystem clock operation (E)
Target state transition: (B) → (E), (C) → (E), (D) → (E)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	PER2 Register	SCMC Register[Note]				SCSC Register	CMC Register	Waiting for Oscillation Stabilization	CSC Register	CKC Register	PER2 Register
Clock After Change	VRTCEN	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP	XT1SELEN		XT1SELDIS	CSS	VRTCEN
Changing to XT1 clock	1	0	1	0/1	0/1	0	1	Necessary	0	1	0
Changing to external sub clock	1	1	1	×	×	0	1	Necessary	0	1	0

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the subsystem clock

Note After release from the reset state, the sub clock operation mode control register (SCMC) can be written only once by an 8-bit memory manipulation instruction.

Operation is possible after turning on the power supply to the VRTC pin and release from the RTC power-on reset.

(7) Changing to low-speed on-chip oscillator clock operation (F)
Target state transition: (B) → (F), (C) → (F), (D) → (F)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register	CKSEL	Oscillation accuracy stabilization time	CKC Register
Clock After Change	SELLOSC		CSS
Changing to low-speed on-chip oscillator	1	210 μs	1

Unnecessary
if the CPU is operating with
the low-speed on-chip
oscillator clock

Remarks 1. ×: don’t care

2. (A) to (R) in **Table 6-3** correspond to (A) to (R) in **Figure 6-24**.

Table 6-3. CPU Clock Transition and SFR Register Setting Examples (5/5)

(8) Changing from CPU operation mode (B), (C), (D), (E), (F) and (Q) to HALT mode (G), (J), (M), (O), (P) and (R)
Target state transition: (B) → (G), (C) → (J), (D) → (M), (E) → (O), (F) → (P), (Q) → (R)

Mode After Change	Setting
HALT mode	Executing HALT instruction

(9) Changing from CPU operation mode (B), (C), and (D) to STOP mode (H), (K), and (N)
Target state transition: (B) → (H), (C) → (K), (D) → (N)

(Setting sequence) →

Mode After Change	Setting		
STOP mode	Stopping peripheral functions that cannot operate in STOP mode	Sets the OSTS register	Executing STOP instruction

Settings are unnecessary if the CPU does not enter STOP mode while it is operating with the high-speed system clock

(10) Changing from STOP mode (H) and (K), to SNOOZE mode (I) and (L)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **20.5.7 SNOOZE mode function** and **20.6.3 SNOOZE mode function**.

Remark (A) to (R) in **Table 6-3** correspond to (A) to (R) in **Figure 6-24**.

6.6.6 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 6-4. Changing CPU Clock (1/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator ● MIOEN = 1	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	X1 clock	Stabilization of X1 oscillation ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin ● OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	PLL clock	Oscillation of the high-speed on-chip oscillator with f_{IH} at 24 MHz and supply as the operating clock for the PLL ● FRQSEL3 to FRQSEL0 = 0000B ● HIOSTOP = 0 ● PLLCK = 1 Stabilization of PLL oscillation ● DSFRDIV1 = 1, DSFRDIV0 = 1, DSCM = 1, DSCON = 1 ● After waiting time for lock-up (40 μs) elapses	-
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator ● SELLOSC = 1	
Middle-speed on-chip oscillator clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator HIOSTOP = 0	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock has been switched to the clock after transition.
	X1 clock	Stabilization of X1 oscillation ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time	
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	-
	External main system clock	Enabling input of external clock from the EXCLK pin OSCSEL = 1, EXCLK = 1, MSTOP = 0	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock has been switched to the clock after transition.
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator ● SELLOSC = 1	

Table 6-4. Changing CPU Clock (2/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
PLL clock	High-speed on-chip oscillator clock	Oscillation of the high-speed on-chip oscillator clock and selection of the high-speed on-chip oscillator as the main system clock ● HIOSTOP = 0, MCS = 0, MCS1 = 0	Operating current can be reduced by stopping the PLL (DSCON = 0, PLLCK = 0) after checking that the CPU clock has been switched to the clock after transition.
	Middle-speed on-chip oscillator clock	Transition not possible	-
	X1 clock	Transition not possible	-
	External main system clock	Transition not possible	-
	XT1 clock	Transition not possible	-
	External subsystem clock	Transition not possible	-
	Low-speed on-chip oscillator clock	Transition not possible	-

Table 6-4. Changing CPU Clock (3/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator ● MIOEN = 1	
	External main system clock	Transition not possible	—
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Stopped XT1 oscillation Selection of low-speed on-chip oscillator ● SELLOSC = 1	
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator ● MIOEN = 1	
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	—
	X1 clock	Transition not possible	
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Stopped XT1 oscillation Selection of low-speed on-chip oscillator ● SELLOSC = 1	

Table 6-4. Changing CPU Clock (4/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock ● HIOSTOP = 0, MCS = 0, MCS1 = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock as main system clock ● MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	—
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time ● MCS = 1	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 1, MSTOP = 0 ● MCS = 1	
	External subsystem clock	Transition not possible	—
	Low-speed on-chip oscillator clock	Transition not possible	

Table 6-4. Changing CPU Clock (5/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock ● HIOSTOP = 0, MCS = 0, MCS1 = 0	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock ● MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	—
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time ● MCS = 1	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock has been switched to the clock after transition.
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 1, MSTOP = 0 ● MCS = 1	
	XT1 clock	Transition not possible	—
	Low-speed on-chip oscillator clock	Transition not possible	

Table 6-4. Changing CPU Clock (6/6)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Low-speed on-chip oscillator clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock ● HIOSTOP = 0, MCS = 0, MCS1 = 0	—
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock ● MIOEN = 1, MCS = 0, MCS1 = 1	
	PLL clock	Setting prohibited (switch to the high-speed on-chip oscillator clock first, and then switch to the PLL clock)	
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time ● MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock ● OSCSEL = 1, EXCLK = 1, MSTOP = 0 ● MCS = 1	
	XT1 clock	Transition not possible	
	External subsystem clock	Transition not possible	

6.6.7 Time required for switchover of CPU clock and main system clock

By setting bits 0, 4, 6 (MCM0, MCM1, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), main system clock can be switched (between the on-chip oscillator clock and the high-speed system clock), and on-chip oscillator clock can be switched (between the high-speed on-chip oscillator clock and the middle-speed on-chip oscillator clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Tables 6-5** to **6-8**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. Whether the main system clock is operating on the high-speed system clock or main on-chip oscillator clock or PLL clock can be ascertained using bit 5 (MCS) of the CKC register. Whether the main system clock is operating on the high-speed on-chip oscillator clock or PLL clock can be ascertained using bit 7 (CKSTR) of the MCKC register. Whether the main on-chip oscillator clock is operating on the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock can be ascertained using bit 1 (MCS1) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 6-5. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{OCO}	↔	f_{MX}	See **Table 6-6**
f_{IH}	↔	f_{IM}	See **Table 6-7**
f_{MAIN}	↔	f_{SUB}	See **Table 6-8**

Table 6-6. Maximum Number of Clocks Required for f_{OCO} ↔ f_{MX}

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{OCO}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{OCO}$)	$f_{MX} \geq f_{OCO}$		2 clocks
	$f_{MX} < f_{OCO}$		2 f_{OCO}/f_{MX} clocks
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{OCO}$	2 f_{MX}/f_{OCO} clocks	
	$f_{MX} < f_{OCO}$	2 clocks	

Table 6-7. Maximum Number of Clocks Required for f_{IH} ↔ f_{IM}

Set Value Before Switchover		Set Value After Switchover	
MCM1		MCM1	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{IM}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{IM} \geq f_{IH}$		2 clocks
	$f_{IM} < f_{IH}$		1 + f_{IH}/f_{IM} clock
1 ($f_{MAIN} = f_{IM}$)	$f_{IM} \geq f_{IH}$	2 f_{IM}/f_{IH} clocks	
	$f_{IM} < f_{IH}$	2 clocks	

Table 6-8. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{SUB}

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAIN}$)	1 ($f_{CLK} = f_{SUB}$)
0 ($f_{CLK} = f_{MAIN}$)		1 + 2 f_{MAIN}/f_{SUB} clock
1 ($f_{CLK} = f_{SUB}$)	3 clock	

Remarks 1. The number of clocks listed in **Tables 6-6** to **6-8** is the number of CPU clocks before switchover.

2. Calculate the number of clocks in **Tables 6-6** to **6-8** 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) to the high- speed system clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} =10 MHz)

$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2$ clocks

6.6.8 Preconditions for stopping clock oscillation

Before stopping the oscillation of a clock, check that the precondition for stopping clock oscillation is satisfied.

For the preconditions for stopping oscillation of the various clocks (and for disabling the external clock inputs) and the register settings to be made to stop oscillation of each clock or disable input of the given clock, see **Table 6-2 Preconditions for Stopping Clock Oscillation and Register Settings**.

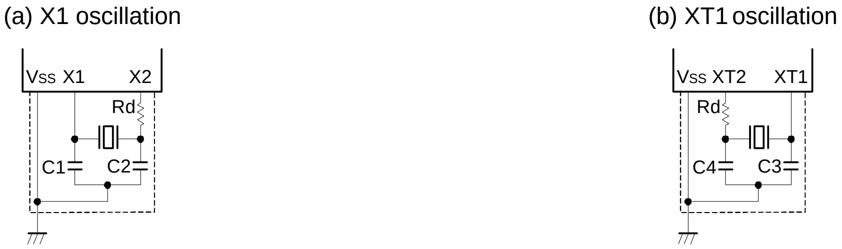
6.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants, see the target product page on the Renesas Web site.

Cautions 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.

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 6-25. Example of External Circuit



(a) X1 oscillation

(b) XT1 oscillation

CHAPTER 7 HIGH-SPEED ON-CHIP OSCILLATOR CLOCK FREQUENCY CORRECTION FUNCTION

7.1 High-speed On-chip Oscillator Clock Frequency Correction Function

Using the subsystem clock f_{SUB} (32.768 kHz)**Note 1** as a reference, the frequency of high-speed on-chip oscillator is measured, and the accuracy of the high-speed on-chip oscillator clock (f_{IH}) frequency is corrected in real time.

Table 7-1 lists the operation specifications of high-speed on-chip oscillator clock frequency correction function and **Figure 7-1** shows the block diagram of high-speed on-chip oscillator clock frequency correction function.

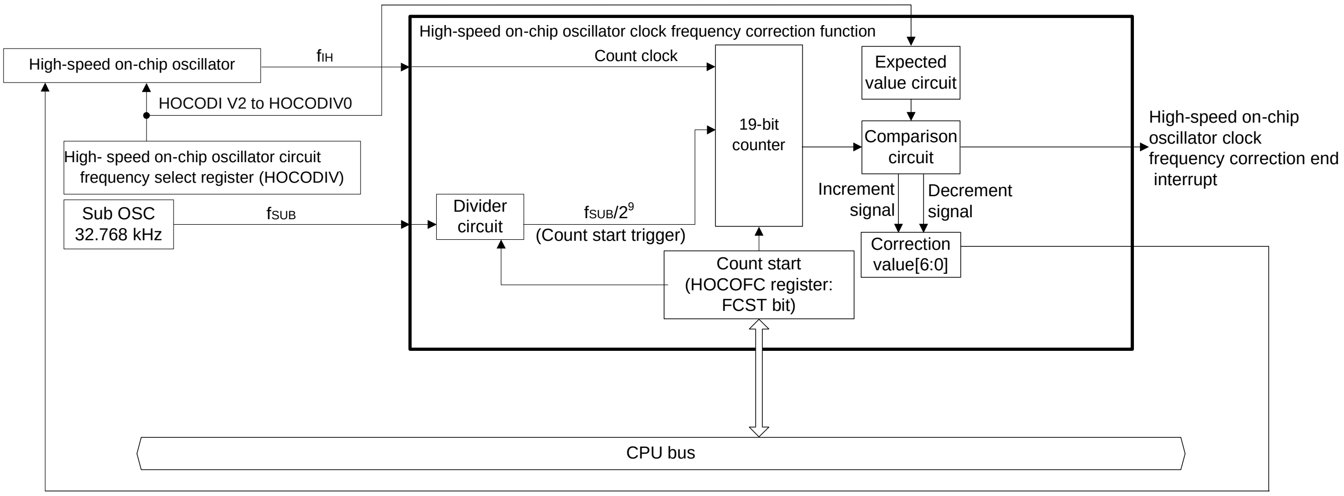
Table 7-1. Operation Specifications of High-speed On-chip Oscillator Clock Frequency Correction Function

Item	Description
Reference clock	● $f_{SUB}/2^9$ (subsystem clock: 32.768 kHz)**Note 1**
Clock to be corrected	● f_{IH} (high-speed on-chip oscillator clock)
Operating modes	● Continuous operating mode The high-speed on-chip oscillator clock frequency is corrected continuously. ● Intermittent operating mode The high-speed on-chip oscillator clock frequency is corrected intermittently using a timer interrupt, etc.
Clock accuracy correction function	● Correction time: Correction cycle (31.2 ms) × (number of corrections – 0.5)**Note 2**
Interrupt	● Output when high-speed on-chip oscillator clock frequency correction is completed (while interrupt output is enabled).

Notes 1. The sub clock (f_{SX}, 38.4 kHz) cannot be used as the reference clock.

2. Correction time: Varies depending on the number of corrections.
Correction cycle: Total time of the frequency measurement phase and the frequency correction phase
Number of corrections: The number of repeated correction cycles until the frequency is adjusted to the expected value range.

Figure 7-1. Block Diagram of High-speed On-chip Oscillator Clock Frequency Correction Function


High-speed on-chip oscillator clock frequency correction function
High-speed on-chip oscillator
fIH
Count clock
Expected value circuit
HOCODI V2 to HOCODIV0
19-bit counter
Comparison circuit
High-speed on-chip oscillator clock frequency correction end interrupt
High- speed on-chip oscillator circuit frequency select register (HOCODIV)
Increment signal
Decrement signal
Sub OSC 32.768 kHz
fSUB
Divider circuit
fSUB/2^9
(Count start trigger)
Correction value[6:0]
Count start (HOCOFC register: FCST bit)
CPU bus

Cautions 1. A subsystem clock is necessary to use the high-speed on-chip oscillator clock frequency correction function. Connect a sub clock oscillator to XT1 and XT2.

2. Use this function as necessary to select a high-speed on-chip oscillator as the operating clock when using a 24 bit ΔΣ type A/D converter.

7.2 Register

Table 7-2 lists the register used for the high-speed on-chip oscillator clock frequency correction function.

Table 7-2. Register for High-speed On-chip Oscillator Clock Frequency Correction Function

Item	Configuration
Control registers	High-speed on-chip oscillator clock frequency correction control register (HOCOFC)

7.2.1 High-speed on-chip oscillator clock frequency correction control register (HOCOFC)

This register is used to control the high-speed on-chip oscillator clock frequency correction function.

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

Reset signal generation clears this register to 00H.

Figure 7-2. Format of High-Speed On-Chip Oscillator Clock Frequency Correction Control Register (HOCOFC)

Address: F02D8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
HOCOFC	FCMD	FCIE	0	0	0	0	0	FCST

FCMD[Note 1]	High-speed on-chip oscillator clock frequency correction function operating mode
0	Continuous operating mode
1	Intermittent operating mode

FCIE	Control of high-speed on-chip oscillator clock frequency correction end interrupt
0	No interrupt is generated when high-speed on-chip oscillator clock frequency correction is completed
1	An interrupt is generated when high-speed on-chip oscillator clock frequency correction is completed

FCST[Note 2]	High-speed on-chip oscillator clock frequency correction circuit operation control/status
0	High-speed on-chip oscillator clock frequency correction circuit stops operating/frequency correction is completed
1	High-speed on-chip oscillator clock frequency correction circuit starts operating/frequency correction is operating
In continuous operating mode, operation is stopped by writing 0 to this bit by software. In intermittent operating mode, the FCST bit is cleared by hardware after correction is completed.	

Notes 1. Do not rewrite the FCMD bit when the FCST bit is 1.

2. When writing 1 to the FCST bit, confirm that the FCST bit is 0 before writing 1 to FCST. However, when writing 1 to the FCST bit immediately after intermittent operation is completed (an interrupt is generated when high-speed on-chip oscillator clock frequency correction is completed), wait for at least one f_{IH} cycle to elapse after the high-speed on-chip oscillator clock frequency correction end interrupt is generated because clearing by hardware has priority.

After writing 0 (high-speed on-chip oscillator clock frequency correction circuit stops operating) to the FCST bit, do not write 1 (high-speed on-chip oscillator clock frequency correction circuit starts operating) to the FCST bit for two f_{IH} cycles.

Caution Be sure to clear bits 5 to 1 to "0".

7.3 Operation

7.3.1 Operation overview

In high-speed on-chip oscillator clock frequency correction, a correction cycle is generated using the subsystem clock (f_{SUB}) as a reference, the frequency of the high-speed on-chip oscillator is measured, and the accuracy of the high-speed on-chip oscillator clock frequency is corrected in real time. In clock correction, operations of the frequency measurement and frequency correction phases are repeated. In the frequency measurement phase, correction is calculated. In the frequency correction phase, the output of the correction value that reflects the correction calculation result is retained.

Table 7-3 lists the high-speed on-chip oscillator input frequency and correction cycle and **Figure 7-3** shows the operation timing (details) of high-speed on-chip oscillator clock frequency correction.

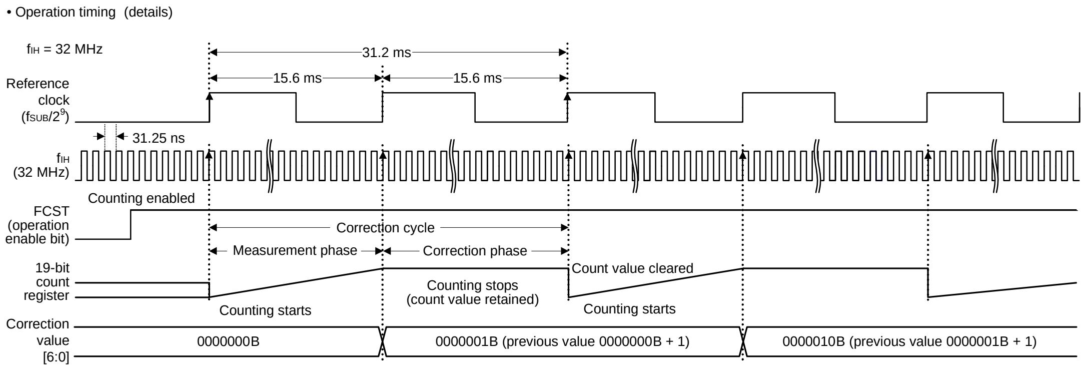
Table 7-3. High-Speed On-Chip Oscillator Input Frequency and Correction Cycle

f_{IH} (MHz)	FRQSEL3, HOCODIV2 to HOCODIV0[Note] (HOCODIV Register)	Correction Cycle (ms)
24	0000	31.2 (frequency measurement phase + frequency correction phase)
12	0001	
6	0010	
3	0011	
1.5	0100	
32	1000	
16	1001	
8	1010	
4	1011	
2	1100	
1	1101	
Other than above	Setting prohibited	

Note Be sure to change the high-speed on-chip oscillator frequency select register (HOCODIV) only when the high-speed on-chip oscillator clock frequency correction function is not used.

The frequency measurement phase period for the correction cycle is counted using the high-speed on-chip oscillator clock, and the high-speed on-chip oscillator frequency is corrected depending on the count result and whether it is greater or smaller than the expected value.

Figure 7-3. Operation Timing (Details) of High-speed On-chip Oscillator Clock Frequency Correction



Remark Basic operation is the same in both continuous and intermittent operating modes. Only the difference between these modes is clearing the FCST bit is controlled by either software or hardware. In addition, the correction value is not cleared until a reset is applied to the system.

(1) Continuous operating mode

In continuous operating mode, the high-speed on-chip oscillator clock frequency is corrected continuously. This mode is selected by setting the FCMD bit in the HOCOFC register to 0.

Operation of high-speed on-chip oscillator clock frequency correction is started by setting the FCST bit in the HOCOFC register to 1. Similarly, operation of high-speed on-chip oscillator clock frequency correction is stopped by setting the FCST bit in the HOCOFC register to 0.

When operation of high-speed on-chip oscillator clock frequency correction is started, frequency counting starts at the rising edge of the reference clock ($f_{SUB}/2^9$) and stops at the next rising edge of the reference clock ($f_{SUB}/2^9$) in the frequency measurement phase.

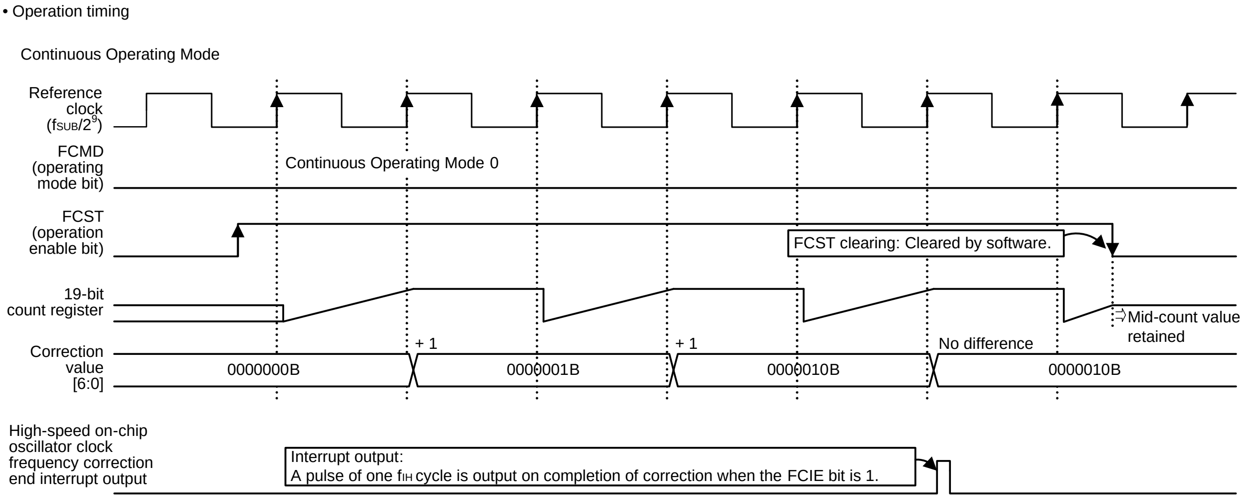
Next, the count value and the expected value are compared, and the correction value is adjusted as follows in the frequency correction phase:

- When the count value is greater than the expected value: Correction value – 1
- When the count value is smaller than the expected value: Correction value + 1
- When the count value is in the range of the expected value: The correction value is retained (high-speed on-chip oscillator clock frequency correction is completed)

When the FCIE bit in the HOCOFC register is set to 1, a high-speed on-chip oscillator clock frequency correction end interrupt is output every time high-speed on-chip oscillator clock frequency correction is completed. In continuous operating mode, the frequency measurement phase and the frequency correction phase are repeated until the high-speed on-chip oscillator clock frequency correction function is stopped.

Figure 7-4 shows the continuous operating mode timing.

Figure 7-4. Continuous Operating Mode Timing


• Operation timing
Continuous Operating Mode
Reference clock ($f_{SUB}/2^9$)
FCMD (operating mode bit)
Continuous Operating Mode 0
FCST (operation enable bit)
FCST clearing: Cleared by software.
19-bit count register
Mid-count value retained
Correction value [6:0]
+ 1
+ 1
No difference
0000000B
0000001B
0000010B
0000010B
High-speed on-chip oscillator clock frequency correction end interrupt output
Interrupt output:
A pulse of one f_{IH} cycle is output on completion of correction when the FCIE bit is 1.

(2) Intermittent operating mode

In intermittent operating mode, the high-speed on-chip oscillator clock frequency is corrected intermittently using a timer interrupt, etc. This mode is selected by setting the FCMD bit in the HOCOFC register to 1.

Operation of high-speed on-chip oscillator clock frequency correction is started by setting the FCST bit in the HOCOFC register to 1.

When operation of high-speed on-chip oscillator clock frequency correction is started, frequency counting starts at the rising edge of the reference clock ($f_{SUB}/2^9$) and stops at the next rising edge of the reference clock ($f_{SUB}/2^9$) in the frequency measurement phase.

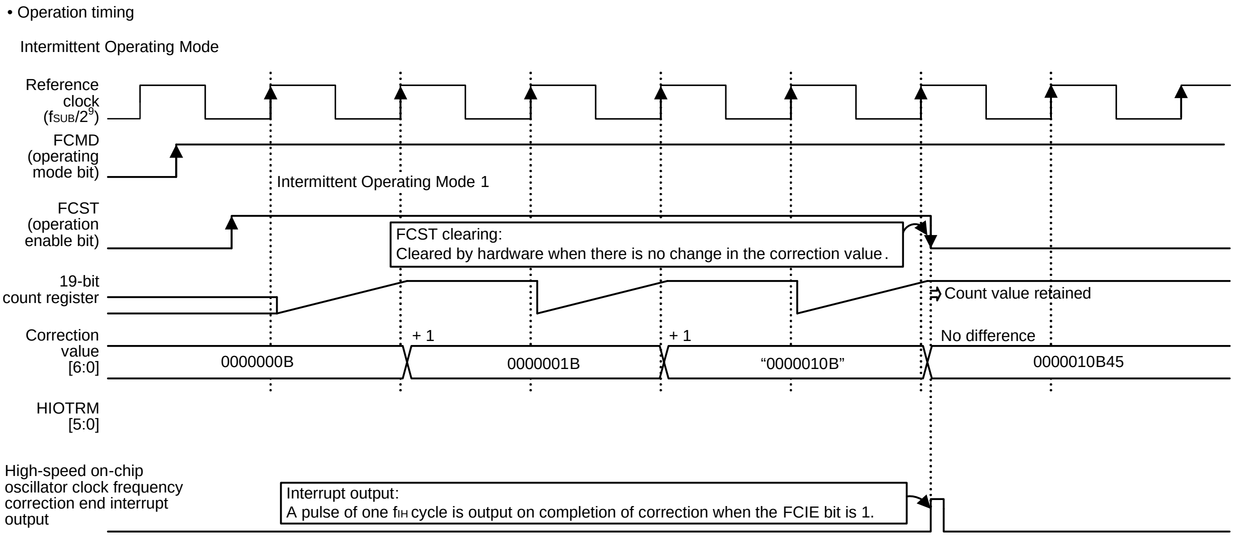
Next, the count value and the expected value are compared, and the correction value is adjusted as follows in the frequency correction phase:

- When the count value is greater than the expected value: Correction value – 1
- When the count value is smaller than the expected value: Correction value + 1
- When the count value is in the range of the expected value: The correction value is retained and the FCST bit is cleared (high-speed on-chip oscillator clock frequency correction is completed)

While the FCIE bit in the HOCOFC register is set to 1, a high-speed on-chip oscillator clock frequency correction end interrupt is output when high-speed on-chip oscillator clock frequency correction is completed. In intermittent operating mode, the frequency measurement phase and the frequency correction phase are repeated, and high-speed on-chip oscillator clock frequency correction operation is stopped after high-speed on-chip oscillator clock frequency correction is completed.

Figure 7-5 shows the intermittent operating mode timing.

Figure 7-5. Intermittent Operating Mode Timing


• Operation timing
Intermittent Operating Mode
Reference clock ($f_{SUB}/2^9$)
FCMD (operating mode bit)
Intermittent Operating Mode 1
FCST (operation enable bit)
19-bit count register
Count value retained
Correction value [6:0]
+ 1
+ 1
No difference
0000000B
0000001B
“0000010B”
0000010B45
HIOTRM [5:0]
High-speed on-chip oscillator clock frequency correction end interrupt output

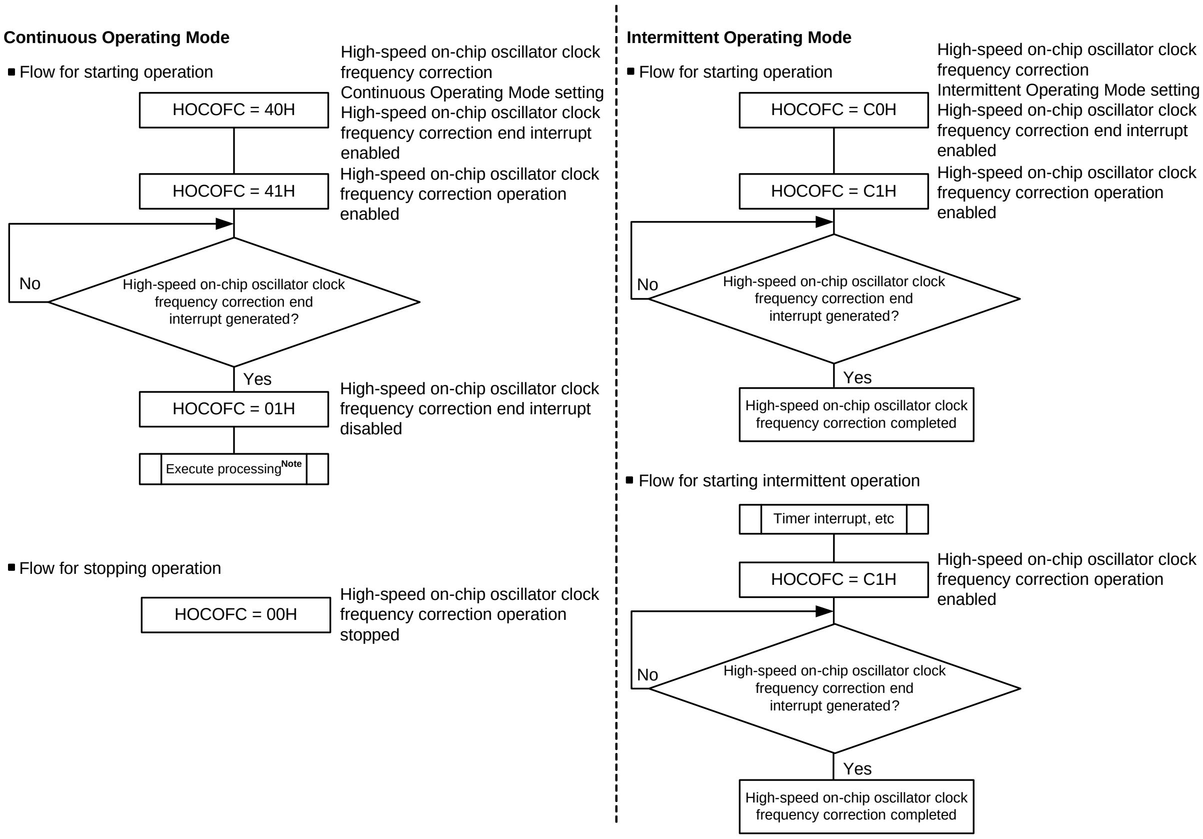
FCST clearing:
Cleared by hardware when there is no change in the correction value.

Interrupt output:
A pulse of one f_{IH} cycle is output on completion of correction when the FCIE bit is 1.

7.3.2 Operation procedure

The following shows the flow for starting and stopping operation when the high-speed on-chip oscillator clock frequency correction function is used.

Figure 7-6. Example of Procedure for Setting Operating Mode


Continuous Operating Mode
■ Flow for starting operation
HOCOFC = 40H
High-speed on-chip oscillator clock frequency correction Continuous Operating Mode setting High-speed on-chip oscillator clock frequency correction end interrupt enabled
HOCOFC = 41H
High-speed on-chip oscillator clock frequency correction operation enabled
No
High-speed on-chip oscillator clock frequency correction end interrupt generated?
Yes
HOCOFC = 01H
High-speed on-chip oscillator clock frequency correction end interrupt disabled
Execute processing Note
■ Flow for stopping operation
HOCOFC = 00H
High-speed on-chip oscillator clock frequency correction operation stopped
Intermittent Operating Mode
■ Flow for starting operation
HOCOFC = C0H
High-speed on-chip oscillator clock frequency correction Intermittent Operating Mode setting High-speed on-chip oscillator clock frequency correction end interrupt enabled
HOCOFC = C1H
High-speed on-chip oscillator clock frequency correction operation enabled
No
High-speed on-chip oscillator clock frequency correction end interrupt generated?
Yes
High-speed on-chip oscillator clock frequency correction completed
■ Flow for starting intermittent operation
Timer interrupt, etc
HOCOFC = C1H
High-speed on-chip oscillator clock frequency correction operation enabled
No
High-speed on-chip oscillator clock frequency correction end interrupt generated?
Yes
High-speed on-chip oscillator clock frequency correction completed

Note The high-speed on-chip oscillator clock frequency correction is repeated until the high-speed on-chip oscillator clock frequency correction function is stopped.

7.4 Usage Notes

7.4.1 SFR access

When writing 1 to FCST to control the FCST bit in intermittent operating mode, confirm that the FCST bit is 0 before writing 1 to the FCST bit. However, when writing 1 to the FCST bit immediately after intermittent operation is completed, wait for at least one f_{IH} cycle after a high-speed on-chip oscillator clock frequency correction end interrupt is generated because clearing by hardware has priority.

7.4.2 Operation during standby state

Be sure to stop operation of high-speed on-chip oscillator clock frequency correction before executing the STOP instruction.

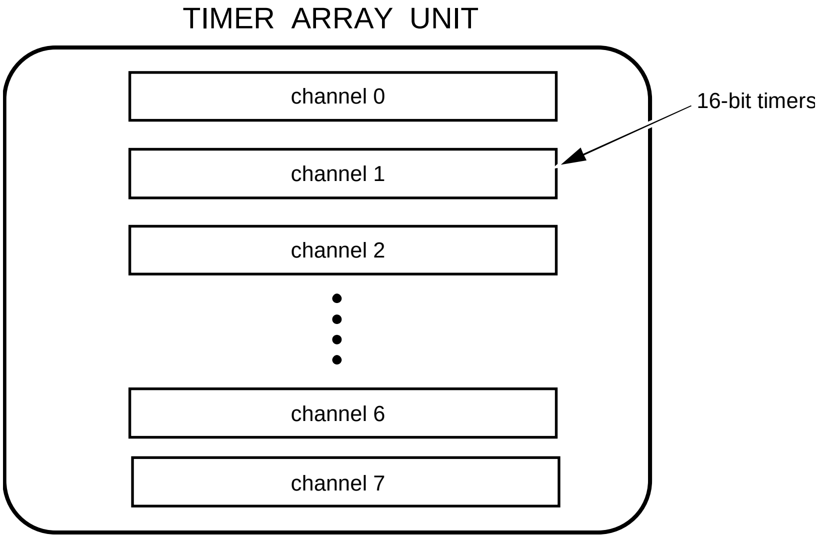
7.4.3 Changing high-speed on-chip oscillator frequency select register (HOCODIV)

Be sure to change the high-speed on-chip oscillator frequency select register (HOCODIV) only when the high-speed on-chip oscillator clock frequency correction function is not used.

CHAPTER 8 TIMER ARRAY UNIT

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 (→ see **8.8.1**) ● Square wave output (→ see **8.8.1**) ● External event counter (→ see **8.8.2**) ● Input pulse interval measurement (→ see **8.8.3**) ● Measurement of high-/low-level width of input signal (→ see **8.8.4**) ● Delay counter (→ see **8.8.5**)	● One-shot pulse output(→ see **8.9.1**) ● PWM output(→ see **8.9.2**) ● Multiple PWM output(→ see **8.9.3**)

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Channel 7 can be used to realize LIN-bus communication operating in combination with UART0 of the serial array unit.

8.1 Functions of Timer Array Unit

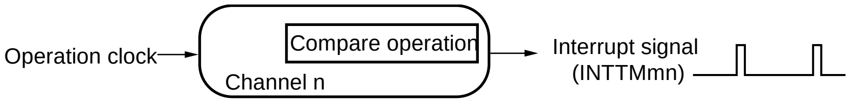
Timer array unit has the following functions.

8.1.1 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.

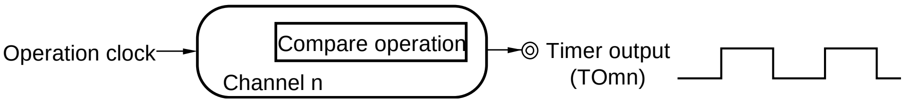
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



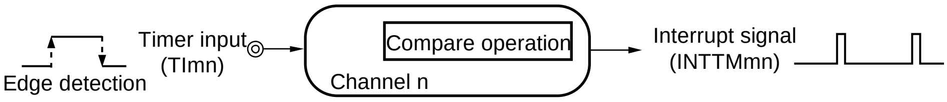
(2) Square wave output

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).



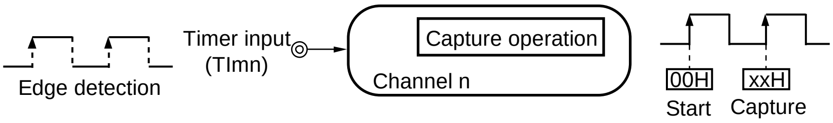
(3) External event counter

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) Input pulse interval measurement

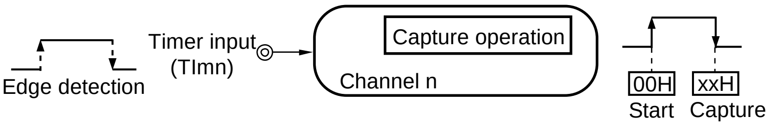
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(**Remark** is listed on the next page.)

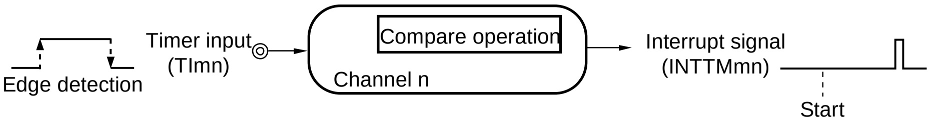
(5) Measurement of high-/low-level width of input signal

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.



(6) 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.



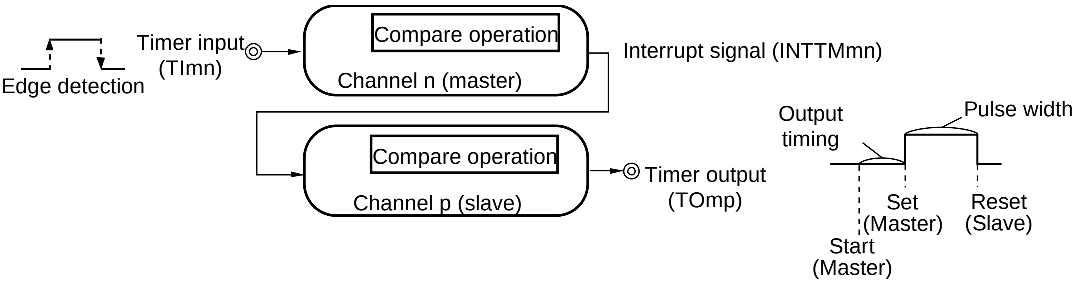
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.1.2 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.

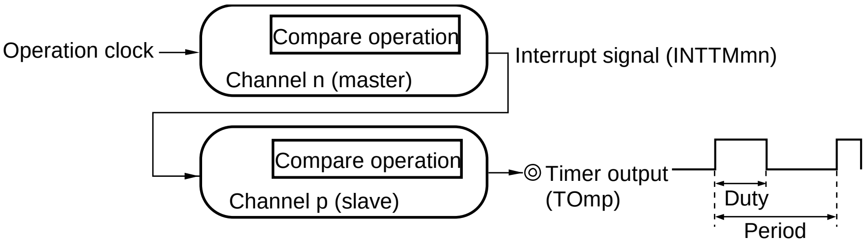
(1) One-shot pulse output

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

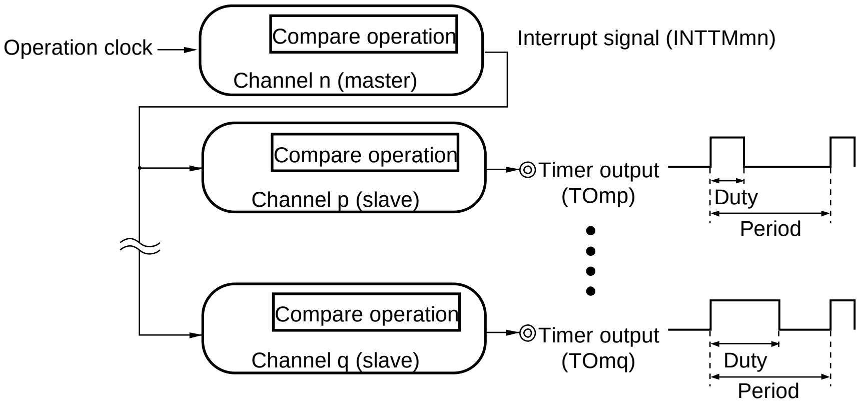
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(**Caution** and **Remark** are listed on the next page.)

(3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.



Caution For details about the rules of simultaneous channel operation function, see 8.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7),
p, q: Slave channel number (n < p < q ≤ 7)

8.1.3 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 8.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

8.1.4 LIN-bus supporting function (channel 7 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 (RxD0) of UART0 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 (RxD0) of UART0 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 (RxD0) of UART0 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 **8.3.14 Input switch control register (ISC)** and **8.8.4 Operation as input signal high-/low-level width measurement**.

8.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

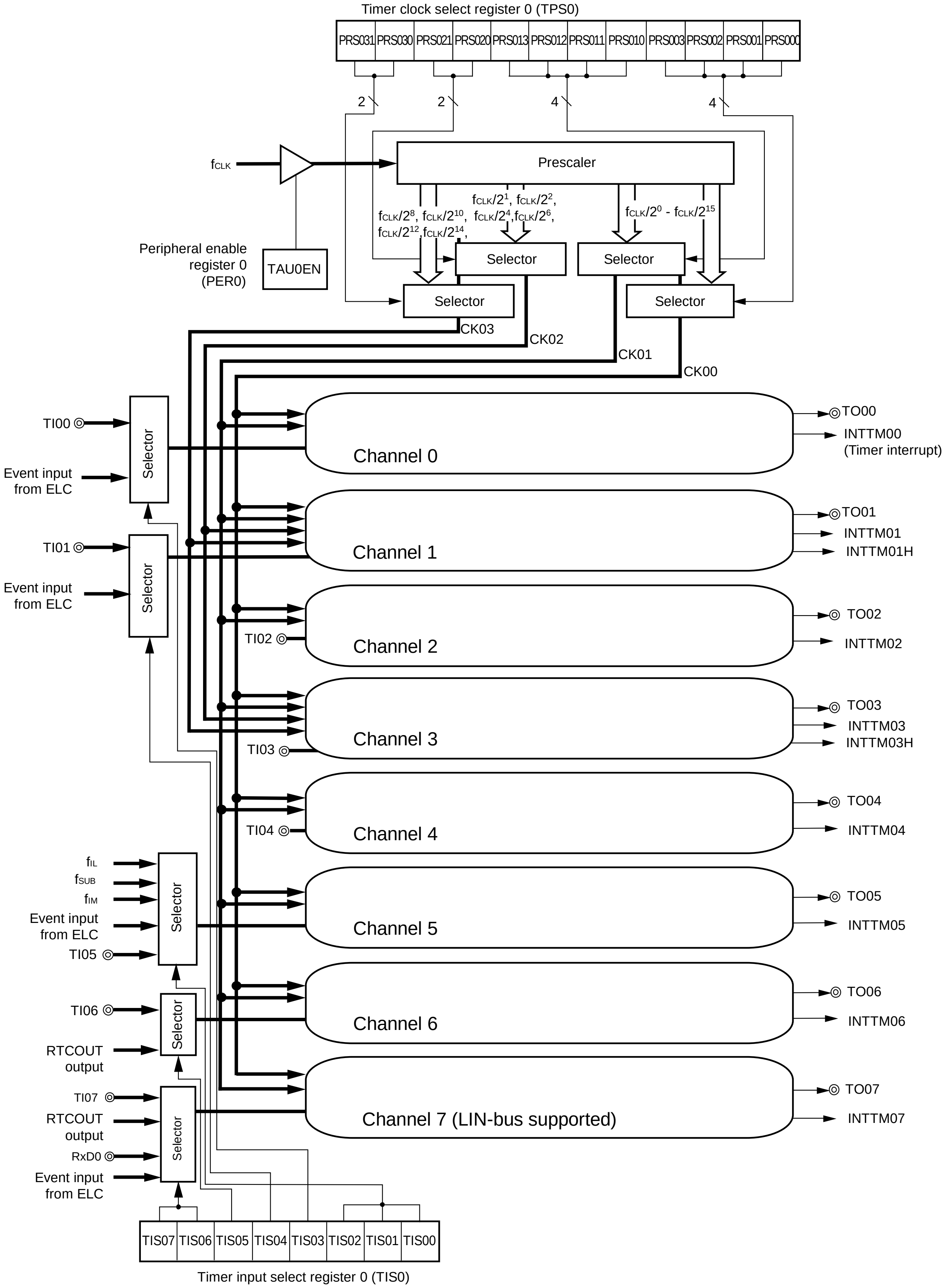
Table 8-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07, RxD0 pin (for LIN-bus)
Timer output	TO00 to TO07 pins, 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 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 (PM0, PM3, PM4, PM6, PM12) ● Port registers (P0, P3, P4, P6, P12)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-1 shows the block diagrams of the timer array unit.

Figure 8-1. Entire Configuration of Timer Array Unit


Timer clock select register 0 (TPS0)
PRS031 PRS030 PRS021 PRS020 PRS013 PRS012 PRS011 PRS010 PRS003 PRS002 PRS001 PRS000
2
2
4
4
f_{CLK}
Prescaler
$f_{CLK}/2^{1}$, $f_{CLK}/2^{2}$,
$f_{CLK}/2^{8}$, $f_{CLK}/2^{10}$, $f_{CLK}/2^{4}$,$f_{CLK}/2^{6}$,
$f_{CLK}/2^{12}$,$f_{CLK}/2^{14}$,
$f_{CLK}/2^{0}$ - $f_{CLK}/2^{15}$
Peripheral enable register 0 (PER0)
TAU0EN
Selector
Selector
Selector
Selector
CK03
CK02
CK01
CK00
TI00
Event input from ELC
Selector
TI01
Event input from ELC
Selector
Channel 0
TO00
INTTM00 (Timer interrupt)
Channel 1
TO01
INTTM01
INTTM01H
TI02
Channel 2
TO02
INTTM02
TI03
Channel 3
TO03
INTTM03
INTTM03H
TI04
Channel 4
TO04
INTTM04
f_{IL}
f_{SUB}
f_{IM}
Event input from ELC
TI05
Selector
Channel 5
TO05
INTTM05
TI06
RTCOUT output
Selector
Channel 6
TO06
INTTM06
TI07
RTCOUT output
RxD0
Event input from ELC
Selector
Channel 7 (LIN-bus supported)
TO07
INTTM07
TIS07 TIS06 TIS05 TIS04 TIS03 TIS02 TIS01 TIS00
Timer input select register 0 (TIS0)

Remark f_{SUB}: Subsystem clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

f_{IM}: Medium-speed on-chip oscillator clock frequency

Figure 8-2. Internal Block Diagram of Channel 0 of Timer Array Unit

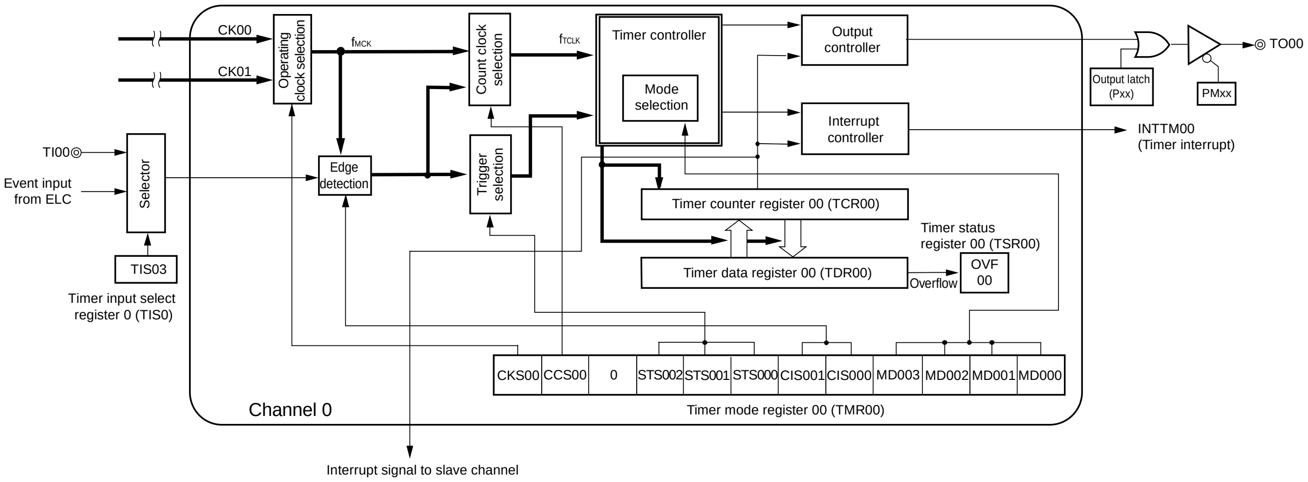
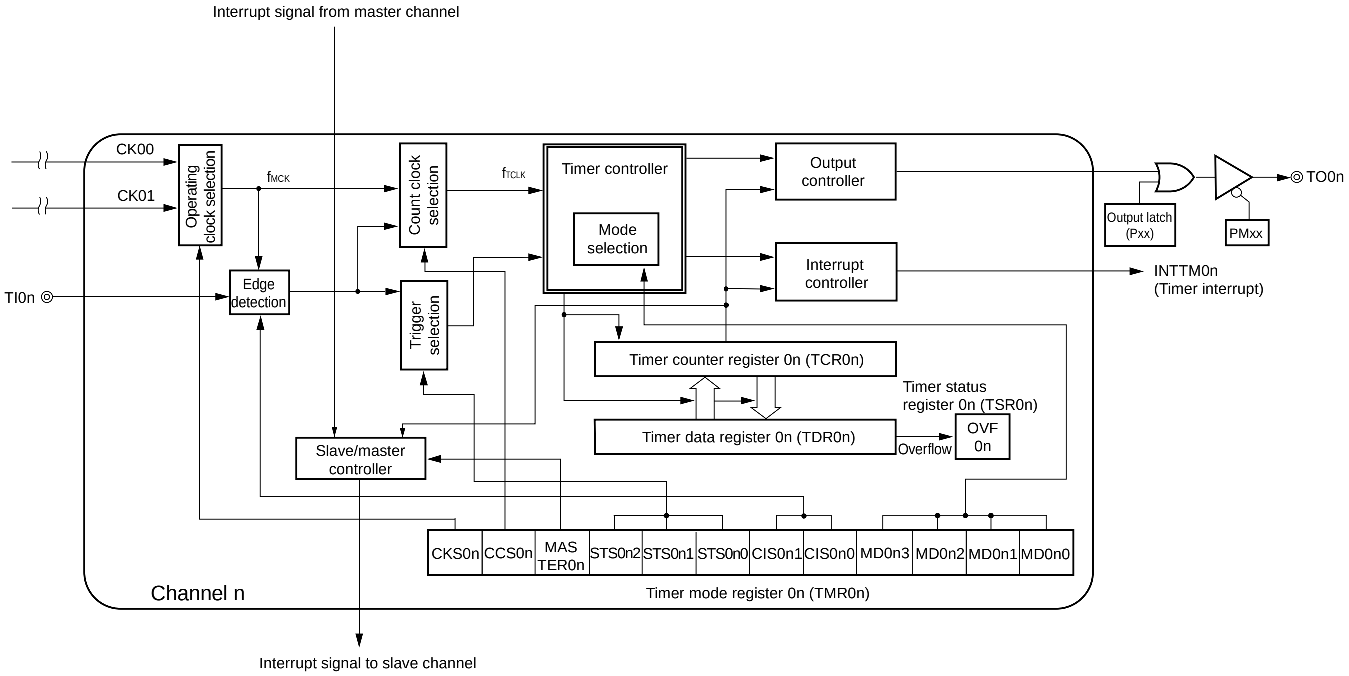


Figure 8-3. Internal Block Diagram of Channels 2, 4 of Timer Array Unit



Remark n = 2, 4

Figure 8-4. Internal Block Diagram of Channel 1 of Timer Array Unit

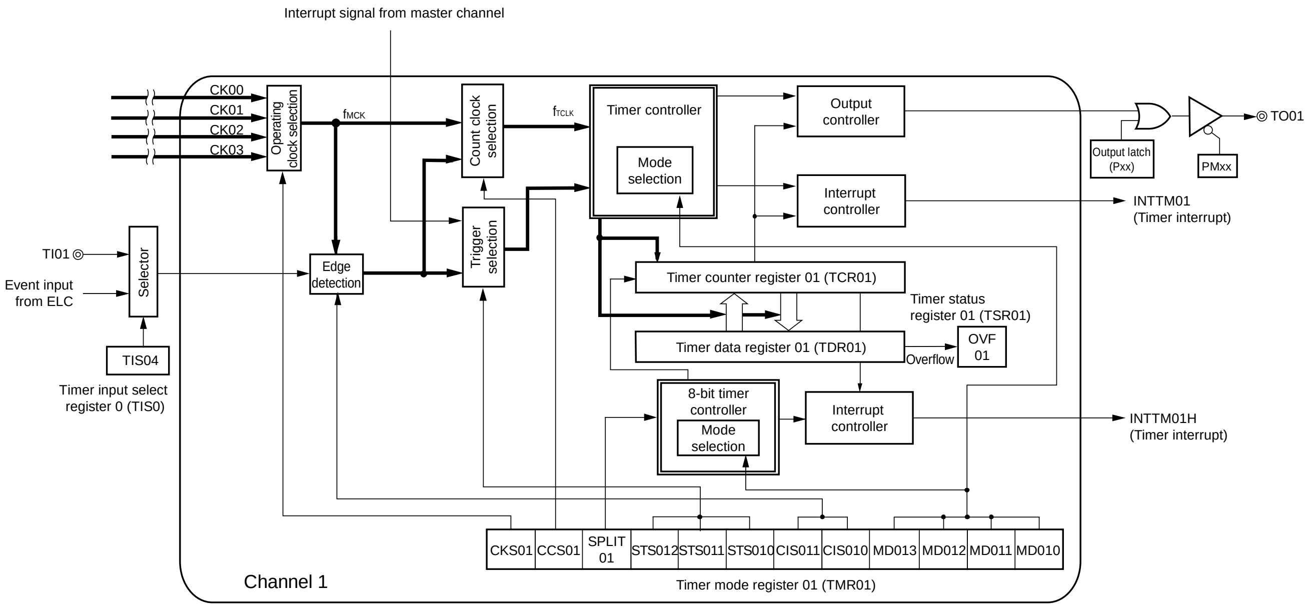


Figure 8-5. Internal Block Diagram of Channel 3 of Timer Array Unit

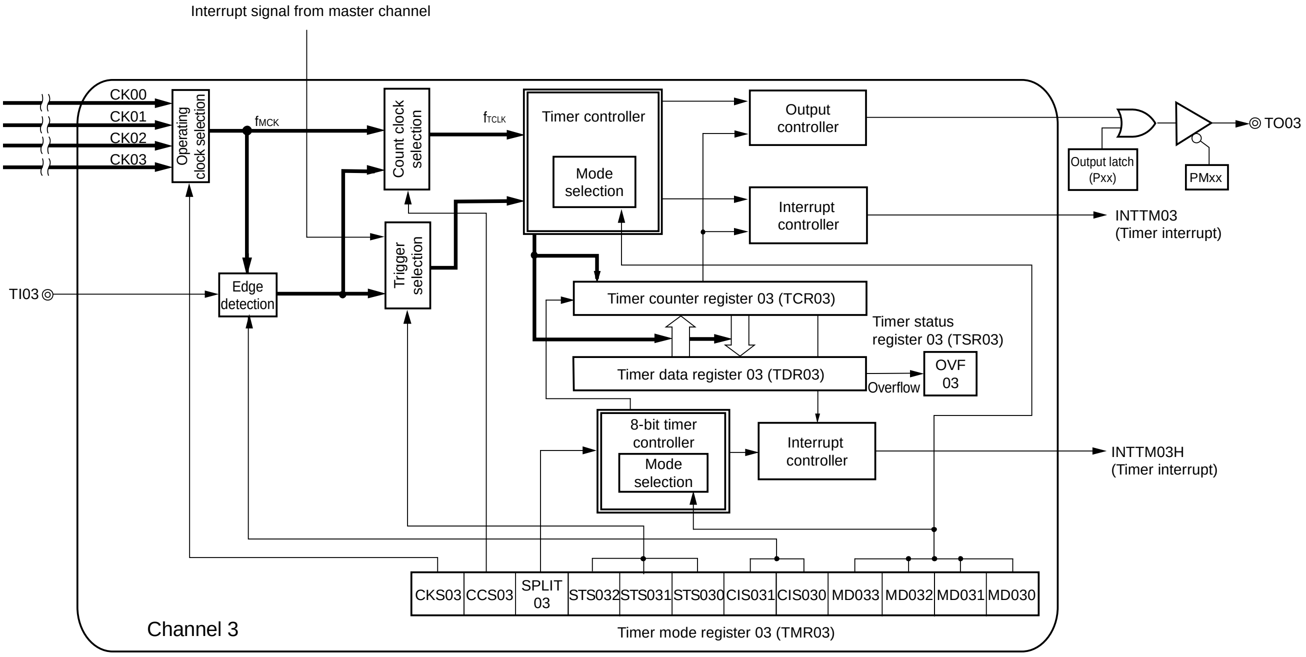


Figure 8-6. Internal Block Diagram of Channel 5 of Timer Array Unit

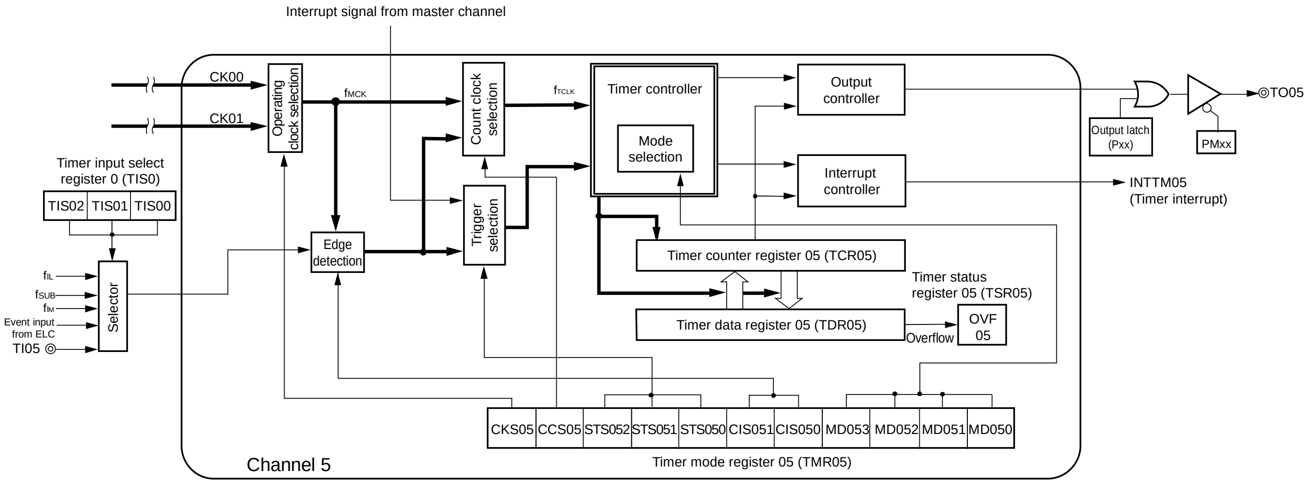


Figure 8-7. Internal Block Diagram of Channel 6 of Timer Array Unit

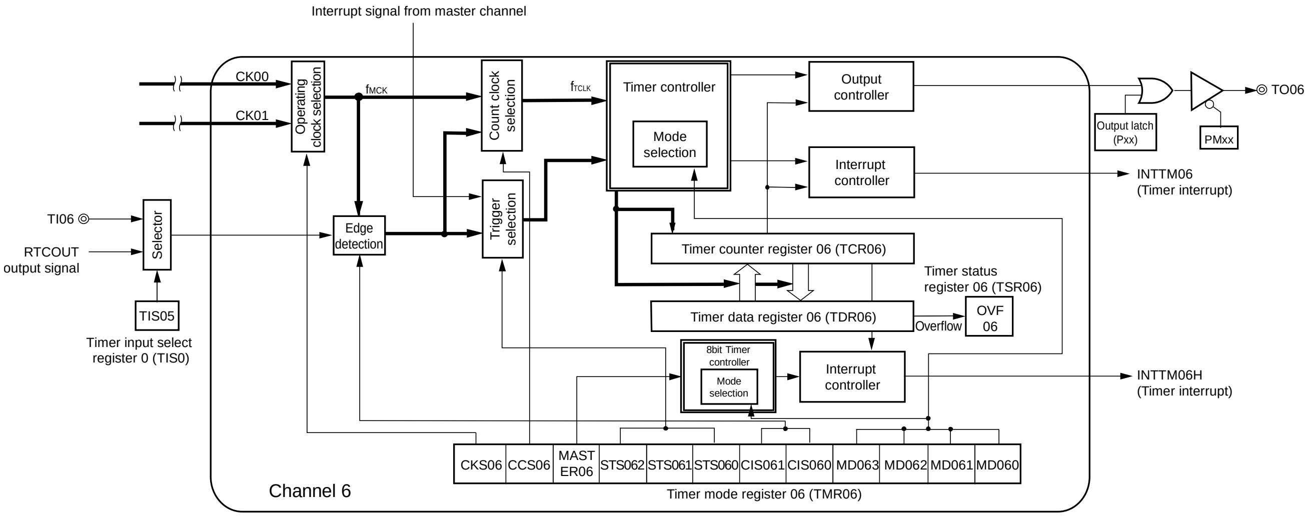
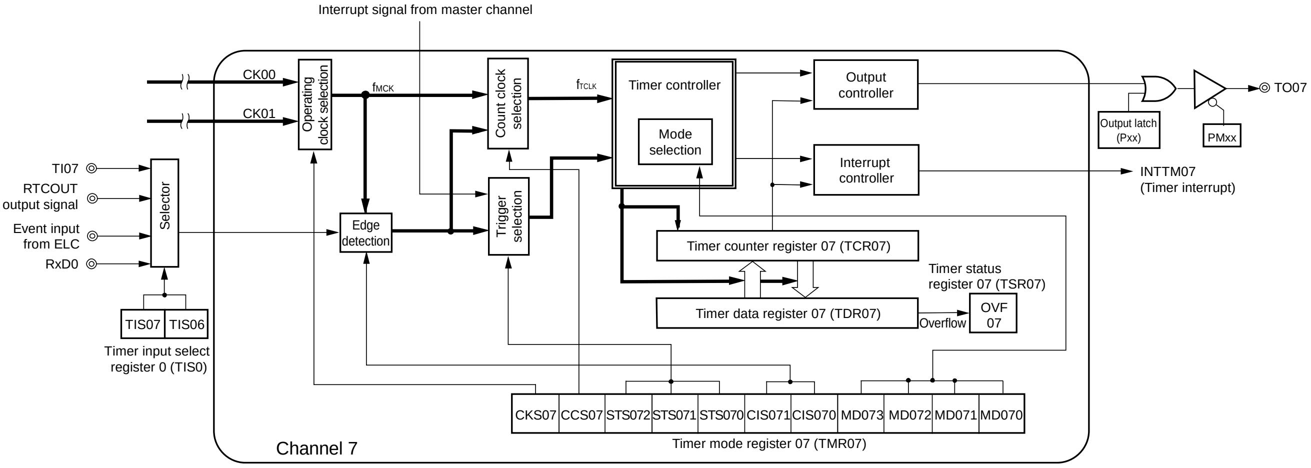


Figure 8-8. Internal Block Diagram of Channel 7 of Timer Array Unit



8.2.1 Timer count register mn (TCRmn)

The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (see **8.3.4 Timer mode register mn (TMRmn)**).

Figure 8-9. Format of Timer Count Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R

F0181H (TCR00) F0180H (TCR00)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The count value can be read by reading timer count register mn (TCRmn).
The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 8-2. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the count operation was paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture 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)

8.2.2 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 SPLIT bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to rewrite 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.

Reset signal generation clears this register to 0000H.

Figure 8-10. Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), After reset: 0000H R/W
FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07)

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

Figure 8-11. Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H R/W

	FFF1BH (TDR01H)								FFF1AH (TDR01L)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer count 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)

8.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- 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 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 (PM0, PM3, PM4, PM6, PM12)
- Port registers (P0, P3, P4, P6, P12)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the timer array unit 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.

Reset signal generation clears this register to 00H.

Figure 8-12. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. ● SFR used by the timer array unit 0 cannot be written. The read value is 00H. However, the SFR is not initialized. [Note]
1	Enables input clock supply. ● SFR used by the timer array unit 0 can be read and written.

Note To initialize the timer array unit 0 and the SFR used by the timer array unit 0, use bit 0 (TAU0RES) of PRR0.

Cautions 1. When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 0, writing to the registers which control the timer array unit is ignored. (except for the timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode registers 0, 3, 4, 6, 12 (PM0, PM3, PM4, PM6, PM12), and port registers 0, 3, 4, 6, 12 (P0, P3, P4, P6, P12)).

- **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)**

2. Be sure to clear bit 7 to “0”.

8.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the timer array unit in the reset state, be sure to set bit 0 (TAU0RES) of this register to 1.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-13. Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

TAU0RES	Control resetting of the timer array unit
0	The timer array unit is released from the reset state.
1	The timer array unit is in the reset state.

8.3.3 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation 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 operation 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 CKm2 is selected as the operation 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 CKm1 is selected as the operation 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 operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 8-14. Format of Timer Clock Select register m (TPSm) (1/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk)[Note] (k = 0, 1)						
					f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 12 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	4 MHz	8 MHz	12 MHz	20 MHz	24 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	2 MHz	4 MHz	6 MHz	10 MHz	12 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	1 MHz	2 MHz	3 MHz	5 MHz	6 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	500 kHz	1 MHz	1.5 MHz	2.5 MHz	3 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	250 kHz	500 kHz	750 kHz	1.25 MHz	1.5 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	125 kHz	250 kHz	375 kHz	625 kHz	750 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	62.5 kHz	125 kHz	188 kHz	313 kHz	375 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	31.3 kHz	62.5 kHz	93.8 kHz	156 kHz	188 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	15.6 kHz	31.3 kHz	46.9 kHz	78.1 kHz	93.8 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	7.81 kHz	15.6 kHz	23.4 kHz	39.1 kHz	46.9 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	11.7 kHz	19.5 kHz	23.4 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	1.95 kHz	3.91 kHz	5.86 kHz	9.76 kHz	11.7 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	976 Hz	1.95 kHz	2.93 kHz	4.88 kHz	5.86 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	488 Hz	976 Hz	1.46 kHz	2.44 kHz	2.93 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	244 Hz	488 Hz	732 Hz	1.22 kHz	1.46 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	122 Hz	244 Hz	366 Hz	610 Hz	732 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

Cautions 1. Be sure to clear bits 15, 14, 11, and 10 to “0”.

2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0, m = 0 to 7), interrupt requests output from timer array units cannot be used.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. The above $f_{CLK}/2^r$ is not a signal which is simply divided f_{CLK} by 2^r, but a signal which becomes high level for one period of f_{CLK} from its rising edge (r = 1 to 15). For details, see **8.5.1 Count clock (f_{TCLK})**.

Figure 8-14. Format of Timer Clock Select register m (TPSm) (2/2)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS m21	PRS m20	Selection of operation clock (CKm2)[Note]						
			f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 12 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2$	2 MHz	4 MHz	6 MHz	10 MHz	12 MHz	16 MHz
0	1	$f_{CLK}/2^2$	1 MHz	2 MHz	3 MHz	5 MHz	6 MHz	8 MHz
1	0	$f_{CLK}/2^4$	250 kHz	500 kHz	750 kHz	1.25 MHz	1.5 MHz	2 MHz
1	1	$f_{CLK}/2^6$	62.5 kHz	125 kHz	188 kHz	313 kHz	375 kHz	500 kHz

PRS m31	PRS m30	Selection of operation clock (CKm3)[Note]						
			f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 12 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2^8$	15.6 kHz	31.3 kHz	46.9 kHz	78.1 kHz	93.8 kHz	125 kHz
0	1	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	11.7 kHz	19.5 kHz	23.4 kHz	31.3 kHz
1	0	$f_{CLK}/2^{12}$	976 Hz	1.95 kHz	2.93 kHz	4.88 kHz	5.86 kHz	7.81 kHz
1	1	$f_{CLK}/2^{14}$	244 Hz	488 Hz	732 Hz	1.22 kHz	1.46 kHz	1.95 kHz

Note When changing the clock selected for f_{CLK} (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 (f_{MCK}) specified by using the CKSmn0, and CKSmn1 bits or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Caution Be sure to clear bits 15, 14, 11, 10 to “0”.

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in **Table 8-3** can be achieved by using the interval timer function.

Table 8-3. Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time[Note] (f_{CLK} = 32 MHz)			
		10 μs	100 μs	1 ms	10 ms
CKm2	$f_{CLK}/2$	√	–	–	–
	$f_{CLK}/2^2$	√	–	–	–
	$f_{CLK}/2^4$	√	√	–	–
	$f_{CLK}/2^6$	√	√	–	–
CKm3	$f_{CLK}/2^8$	–	√	√	–
	$f_{CLK}/2^{10}$	–	√	√	–
	$f_{CLK}/2^{12}$	–	–	√	√
	$f_{CLK}/2^{14}$	–	–	√	√

Note The margin is within 5%.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. For details of a signal of $f_{CLK}/2^j$ selected with the TPSm register, see **8.5.1 Count clock (f_{TCLK}).**

8.3.4 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **8.8 Independent Channel Operation Function of Timer Array Unit** and **8.9 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.
TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 8-15. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0[Note]	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCS mn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channels 0, 1, 5, 6, and 7, valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to “0”.

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} 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 (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-15. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0[Note]	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(Bit 11 of TMRmn (n = 2, 4, 6))

MAS TER mn	Selection of independent channel operation or simultaneous channel operation (slave or master) of channel n
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 of 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

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-15. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0[Note]	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

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.		

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-15. Format of Timer Mode Register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 5, 7)	CKS mn1	CKS mn0	0	CCS mn	0Note 1	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

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/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 of each mode varies depending on MDmn0 bit (see the table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see 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 modeNote 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 operationNote 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.

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

Notes 1. Bit 11 is fixed at 0 of read only, write is ignored.

2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

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).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.5 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status 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 8-4** 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 lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 8-16. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status 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 8-4. 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.

8.3.6 Timer channel enable status register m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 8-17. Format of Timer Channel Enable Status register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

TEH03	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.

TEH01	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.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 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)

8.3.7 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count 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.

Reset signal generation clears this register to 0000H.

Figure 8-18. Format of Timer Channel Start register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	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 8-5** in **8.5.2 Start timing of counter**).

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 8-5** in **8.5.2 Start timing of counter**).

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 8-5** in **8.5.2 Start timing of counter**). 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.

Cautions 1. Be sure to clear bits 15 to 12, 10, 8 to “0”

2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.

When the TImn pin noise filter is enabled (TNFENnm = 1): Four cycles of the operation clock (f_{MCK})

When the TImn pin noise filter is disabled (TNFENnm = 0): Two cycles of the operation clock (f_{MCK})

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.8 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, TTHm1,

TTHm3 = 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.

Reset signal generation clears this register to 0000H.

Figure 8-19. Format of Timer Channel Stop register m (TTm)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	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. The TTm1 and TTm3 bits respectively trigger stoppage of operation of the lower 8-bit timers in channels 1 and 3 when these channels are in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 of the TTm register to "0".

Remarks 1. When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0),n: Channel number (n = 0 to 7)

8.3.9 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 0, 1, 5, 6, 7 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-20. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TIS07	TIS06	Selection of timer input used with channel 7
0	0	Input signal of timer input pin (TI07)
0	1	RTCOUT output signal[Note]
1	0	RXDo input pin
1	1	Event input signal from ELC

TIS05	Selection of timer input used with channel 6
0	Input signal of timer input pin (TI06)
0	RTCOUT output signal[Note]

TIS04	Selection of timer input used with channel 1
0	Input signal of timer input pin (TI01)
1	Event input signal from ELC

TIS03	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI05)
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 above			Setting prohibited

Note When the RTCOUT output signal is selected, be sure to select it as an input source for both of channel 6 and 7.

Caution High-level width, low-level width of timer input is selected, will require more than 1/f_{MCK} +10 ns. Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKC register = 1), can not TIS02 bit set to 1.

8.3.10 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.

Reset signal generation clears this register to 0000H.

Figure 8-21. Format of Timer Output Enable register m (TOEm)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	TOE m7	TOE m6	TOE m5	TOE m4	TOE m3	TOE m2	TOE m1	TOE m0

TOE mn	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TOmn bit timer operation, to fixed the output. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Enable output of timer. Reflected in the TOmn bit timer operation, to generate the output waveform. Writing to the TOmn bit is disabled (writing 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)

8.3.11 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 oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P43/TI00/TO00, P42/TI01/TO01, P07/TI02/TO02, P06/TI03/TO03, P05/TI04/TO04, P04/TI05/TO05, P03/TI06/TO06, or P02/TI07/TO07 pin as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 8-22. Format of Timer Output register m (TOm)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	TOm 7	TOm 6	TOm 5	TOm 4	TOm 3	TOm 2	TOm 1	TOm 0

TOm n	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.12 Timer output level register m (TOLm)

The TOLm register is a register that 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.

Reset signal generation clears this register to 0000H.

Figure 8-23. Format of Timer Output Level register m (TOLm)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	TOL m7	TOL m6	TOL m5	TOL m4	TOL m3	TOL m2	TOL m1	0

TOL mn	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”.

Remarks 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.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.3.13 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.

Reset signal generation clears this register to 0000H.

Figure 8-24. Format of Timer Output Mode register m (TOMm)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	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 toggle 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, see **8.4.1 Basic rules of simultaneous channel operation function**.)

8.3.14 Input switch control register (ISC)

The ISC register is used to select the input signal for an external interrupt (INTP0).

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

Reset signal generation clears this register to 00H.

Figure 8-25. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	0	ISC0

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 RxD0 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 1 to “0”.

8.3.15 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel[Note].

The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note For details, see **8.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **8.5.2 Start timing of counter**, and **8.7 Timer Input (TImn) Control**.

Figure 8-26. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin or RxD0 pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	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 RxD0 pin can be selected.

8.3.16 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx) and port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

The port mode register (PMxx) and port register (Pxx) to be set depend on the product. For details, see **4.5 Register Settings When Using Alternate Function**.

When using the ports (such as P43/TI00/TO00) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P43/ TO00 for timer output
Set the PM43 bit of port mode register 4 to 0.
Set the P43 bit of port register 4 to 0.

When using the ports (such as P43/TI00) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P43/TO00 for timer input
Set the PM43 bit of port mode register 4 to 1.
Set the P43 bit of port register 4 to 0 or 1.

Remarks 1. In the case of the 80-pin product, in order to use a port pin that is multiplexed with a segment output for the timer I/O function, be sure to set the corresponding bit from among PFSEG27 of LCD port function register 4 (PFSEG4) and PFSEG32 to PFSEG37 of LCD port function register 5 (PFSEG5) to 0.

2. When using the P125/(TI05)/(TO05)/V_{L3} pin for timer I/O, be sure to clear the ISCVL3 bit of the LCD Input switch control register (ISCLCD) to "0".

3. When using the P126/(TI04)/(TO04)/CAPL and P127/(TI03)/(TO03)/CAPH pins for timer I/O, be sure to clear the ISCCAP bit of the LCD Input switch control register (ISCLCD) to "1".

8.4 Basic Rules of Timer Array Unit

8.4.1 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 (channel 0, 2, 4, etc.) can be set as a master channel.

(2) Any channel, except channel 0, can be set as a slave channel.

(3) The slave channel must be lower than the master channel.

Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.

(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.

Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.

(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.

(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.

(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.

(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.

(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.

(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.

(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.

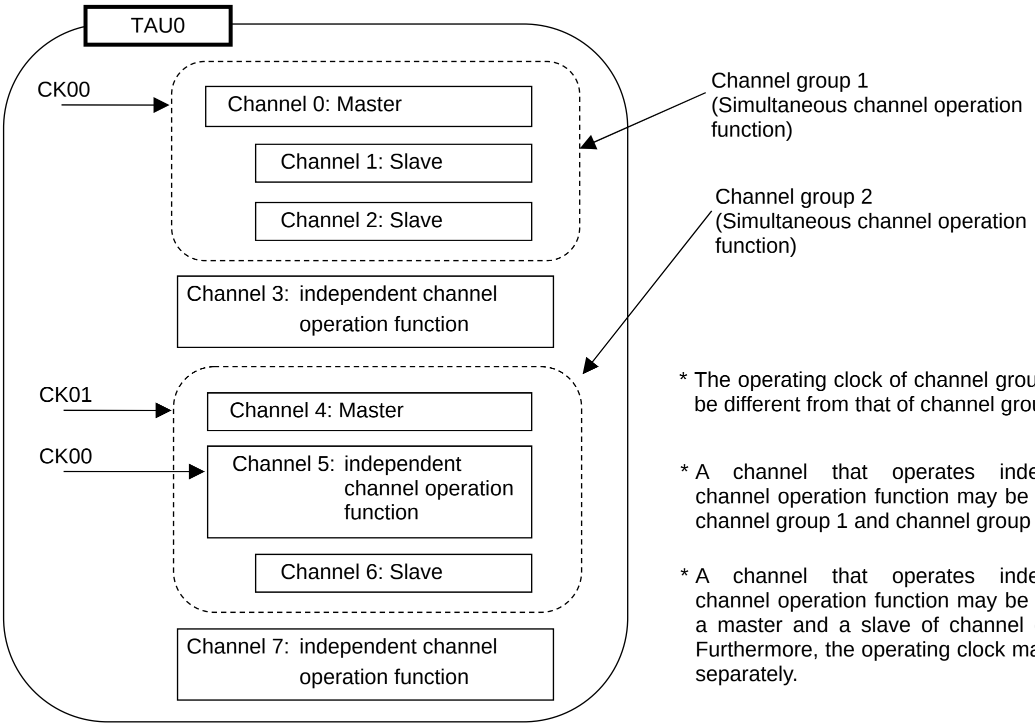
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

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 **8.4.1 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


TAU0
CK00
Channel 0: Master
Channel 1: Slave
Channel 2: Slave
Channel 3: independent channel operation function
CK01
Channel 4: Master
CK00
Channel 5: independent channel operation function
Channel 6: Slave
Channel 7: independent channel operation function
Channel group 1 (Simultaneous channel operation function)
Channel group 2 (Simultaneous channel operation function)

* The operating clock of channel group 1 may be different from that of channel group 2.

* A channel that operates independent channel operation function may be between channel group 1 and channel group 2.

* A channel that operates independent channel operation function may be between a master and a slave of channel group 2. Furthermore, the operating clock may be set separately.

8.4.2 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 SPLIT 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 operation 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 status 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
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status 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)

8.5 Operation of Counter

8.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

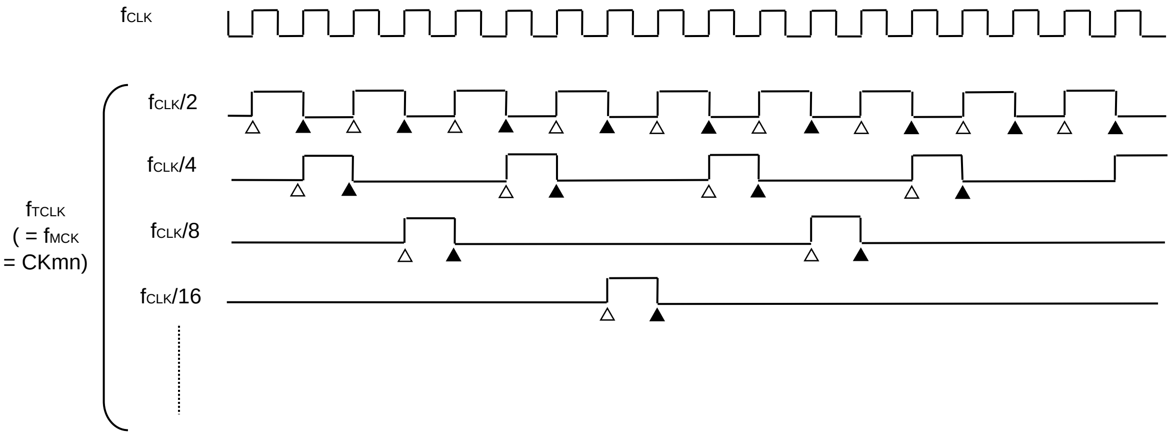
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (f_{TCLK}) is between f_{CLK} to f_{CLK} /2^{15} 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 count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 8-27. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)



Remarks 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

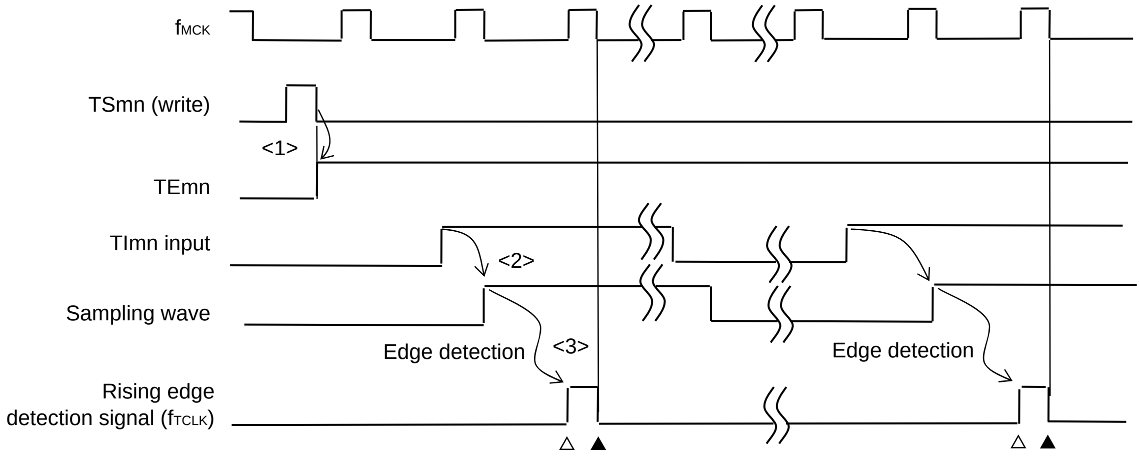
2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 8-28. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.
<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.
<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

3. The waveform of the TImn pin input signal, which is used for input pulse interval measurement, input signal of high/low width measurement, the delay counter, and one-shot pulse output, is the same as that shown in above figure.

8.5.2 Start timing of counter

Timer count 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 count Register mn (TCRmn) count start is shown in **Table 8-5**.

Table 8-5. Operations from Count Operation Enabled State to Timer Count Register mn (TCRmn) Count Start

Timer Operation Mode	Operation When TSmn = 1 Is Set
● Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **8.5.3 (1) Operation of interval timer mode**).
● Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation (see **8.5.3 (2) Operation of event counter mode**).
● Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **8.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
● One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **8.5.3 (4) Operation of one-count mode**).
● Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **8.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

8.5.3 Operation of counter

The counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

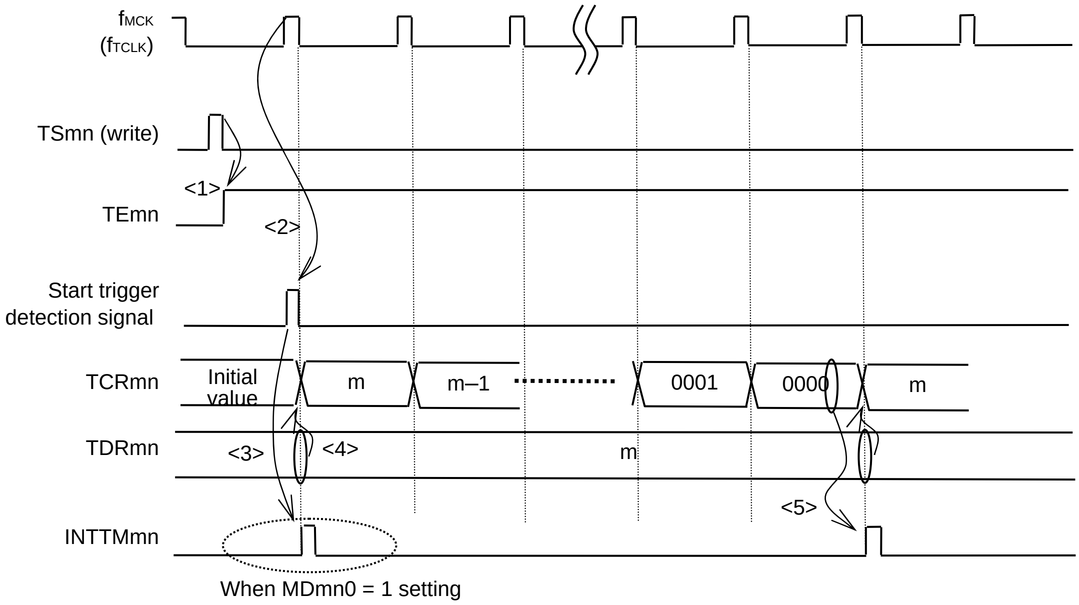
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 8-29. Operation Timing (In Interval Timer Mode)



Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark f_{MCK}, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with f_{CLK}.

(2) Operation of event counter mode

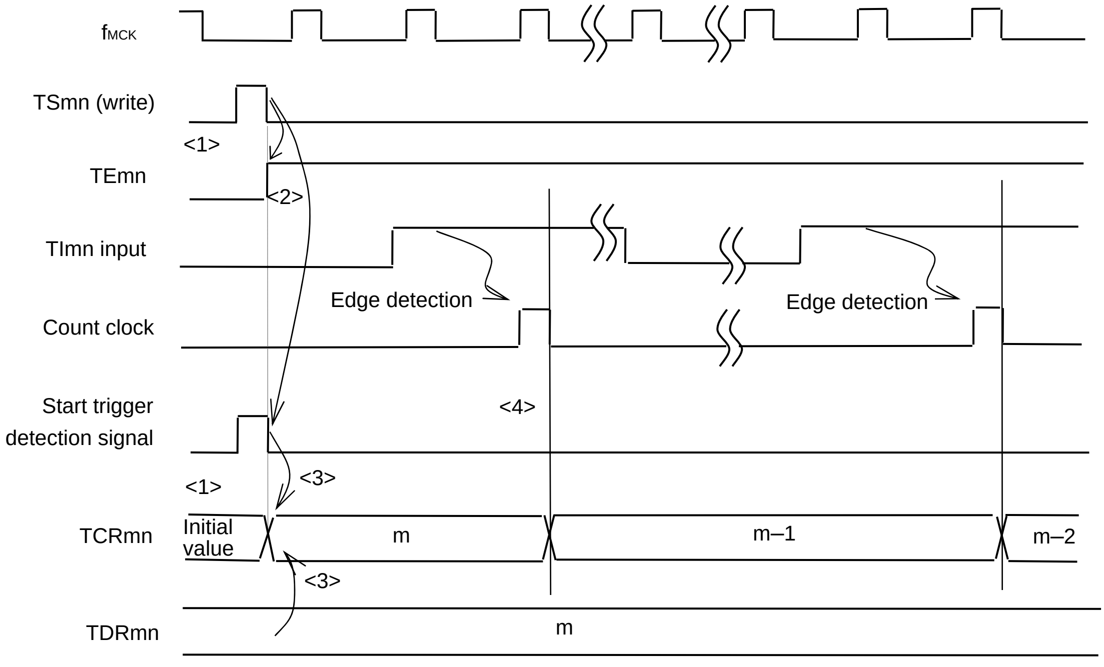
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<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 count clock of the valid edge of the TImn input .

Figure 8-30. Operation Timing (In Event Counter Mode)



Remark The timing is shown in above figure indicates while 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 count clock (fMCK).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

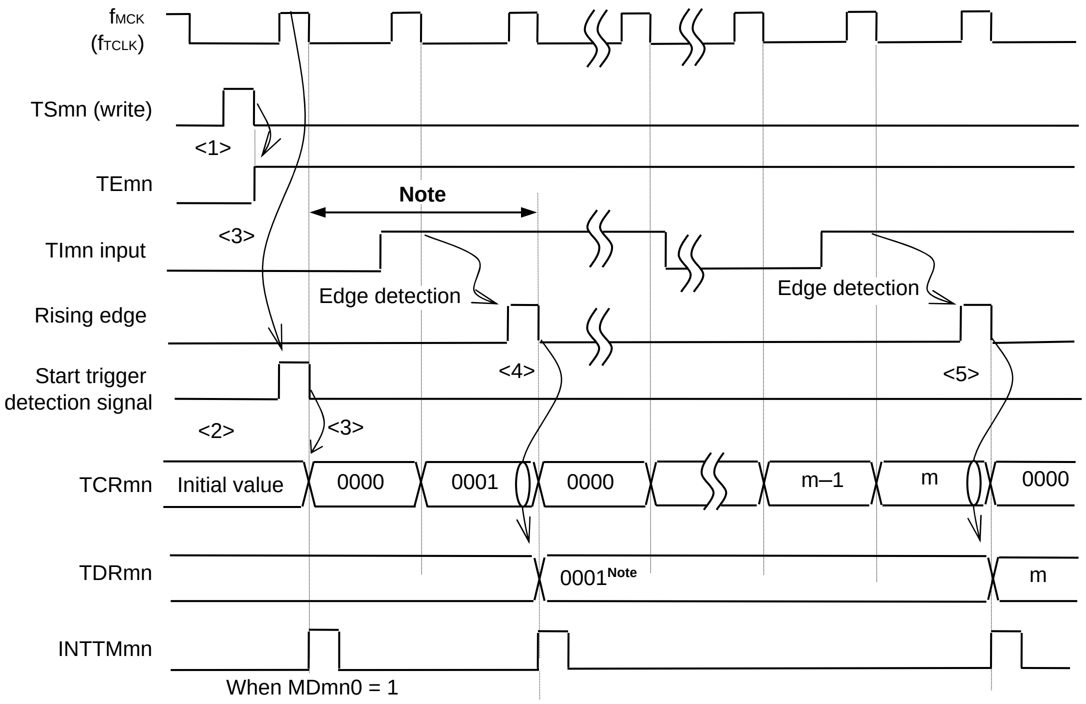
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)

<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is nomeaning. The TCRmn register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 8-31. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

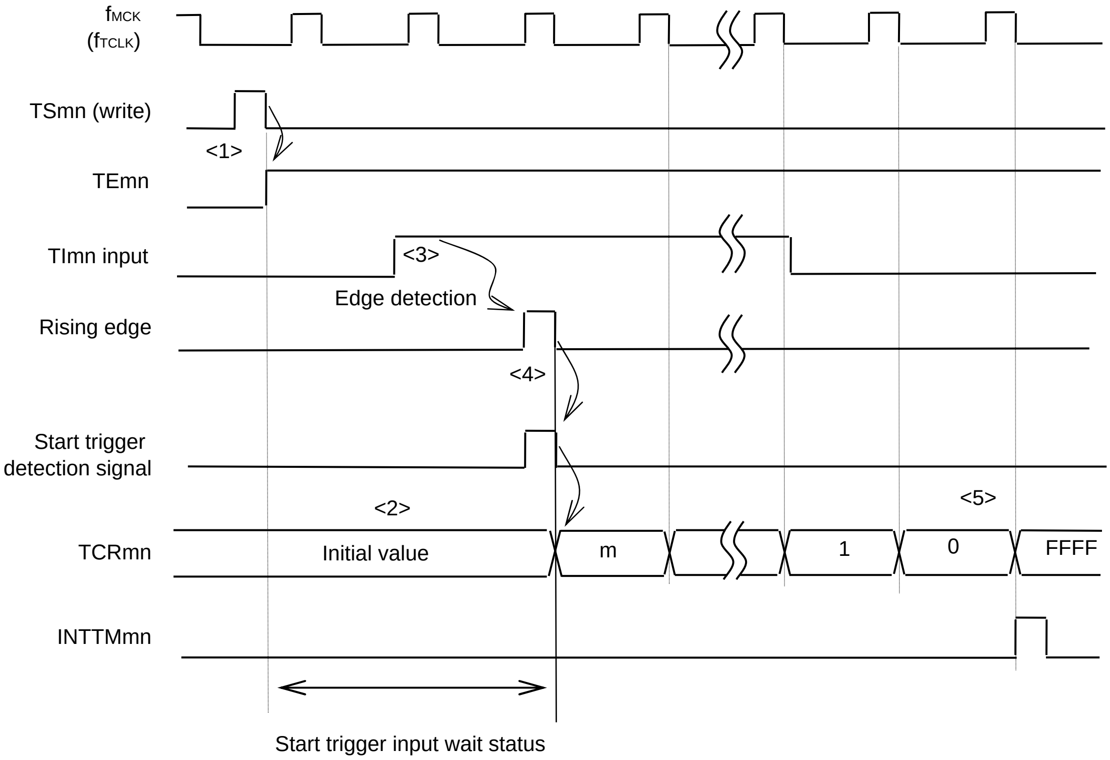
Remark The timing is shown in above figure indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one cycle occurs because the TImn input is not synchronous with the count clock (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.
<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops

.

Figure 8-32. Operation Timing (In One-count Mode)

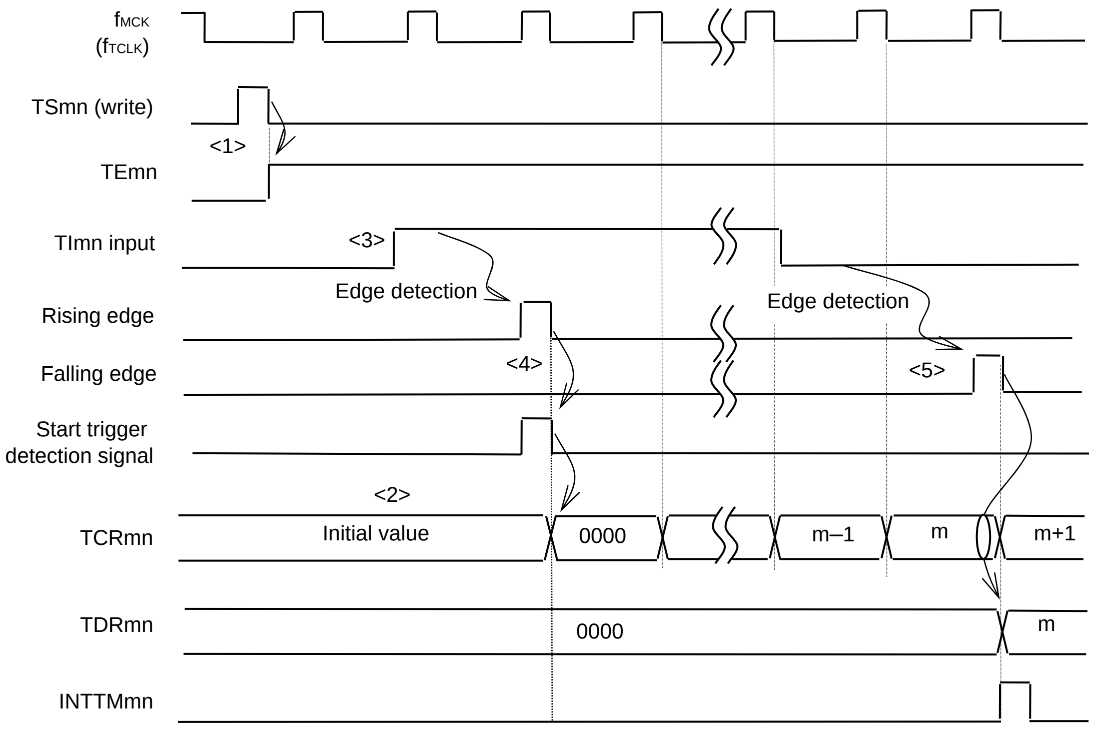


Remark The timing is shown in above figure indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.
<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 8-33. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

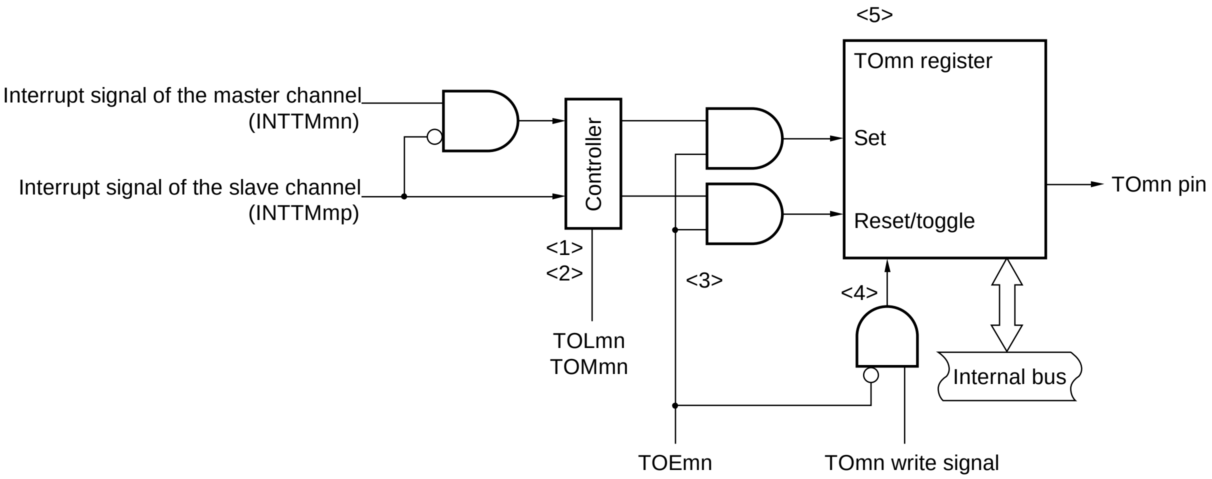


Remark The timing is shown in above figure indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

8.6 Channel Output (TOmn Pin) Control

8.6.1 TOmn pin output circuit configuration

Figure 8-34. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

<5> The TOm register can always be read, and the TOmn pin output level can be checked.

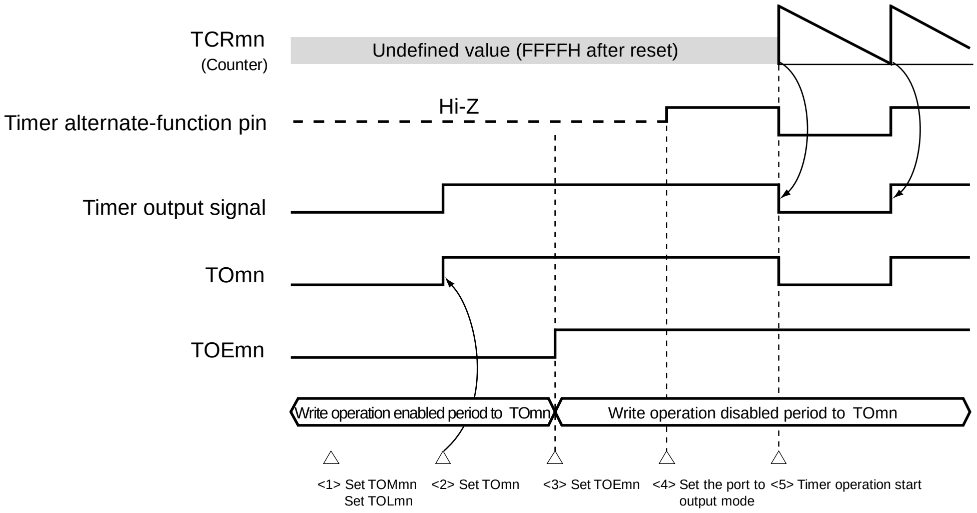
Caution Since outputs are N-ch open-drain outputs, an external pull-up resistor is required to use P60, P61, and P62 as channel output.

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

8.6.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 8-35. Status Transition from Timer Output Setting to Operation Start


TCRmn
(Counter)
Undefined value (FFFFH after reset)
Timer alternate-function pin
Hi-Z
Timer output signal
TOmn
TOEmn
Write operation enabled period to TOmn
Write operation disabled period to TOmn
<1> Set TOMmn Set TOLmn
<2> Set TOmn
<3> Set TOEmn
<4> Set the port to output mode
<5> Timer operation start

<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 status 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 I/O setting is set to output (see **8.3.16 Registers controlling port functions of pins to be used for timer I/O**).
<5> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.6.3 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 count 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 by 8.8 and 8.9.

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)

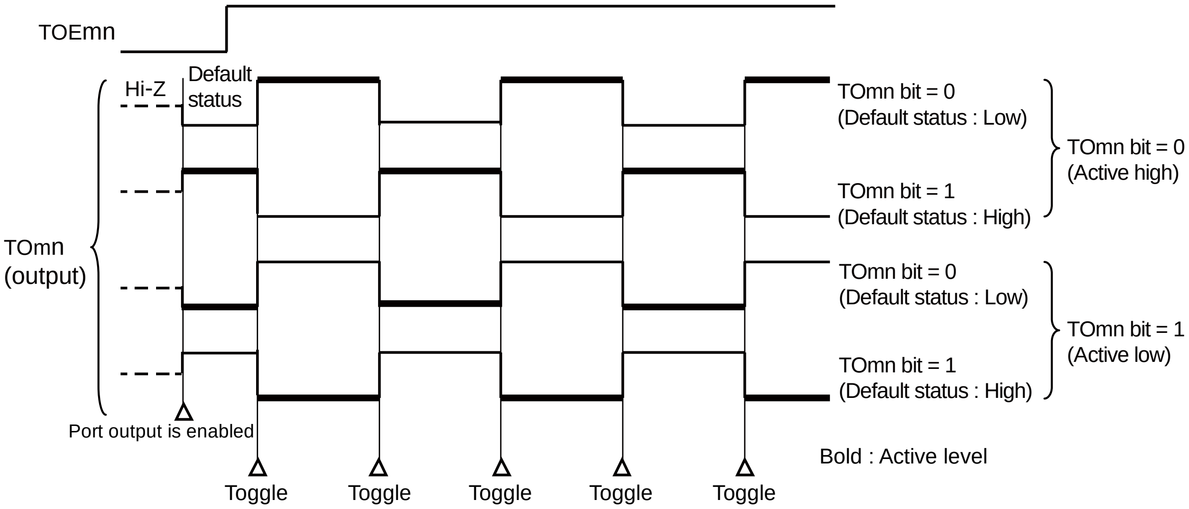
(2) Default level of TOmn pin and output level after timer operation start

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 reversed.

Figure 8-36. TOmn Pin Output Status at Toggle Output (TOMmn = 0)


TOEmn
Hi-Z
Default status
TOmn (output)
TOmn bit = 0 (Default status : Low)
TOmn bit = 1 (Default status : High)
TOmn bit = 0 (Active high)
TOmn bit = 0 (Default status : Low)
TOmn bit = 1 (Default status : High)
TOmn bit = 1 (Active low)
Port output is enabled
Bold : Active level
Toggle
Toggle
Toggle
Toggle
Toggle

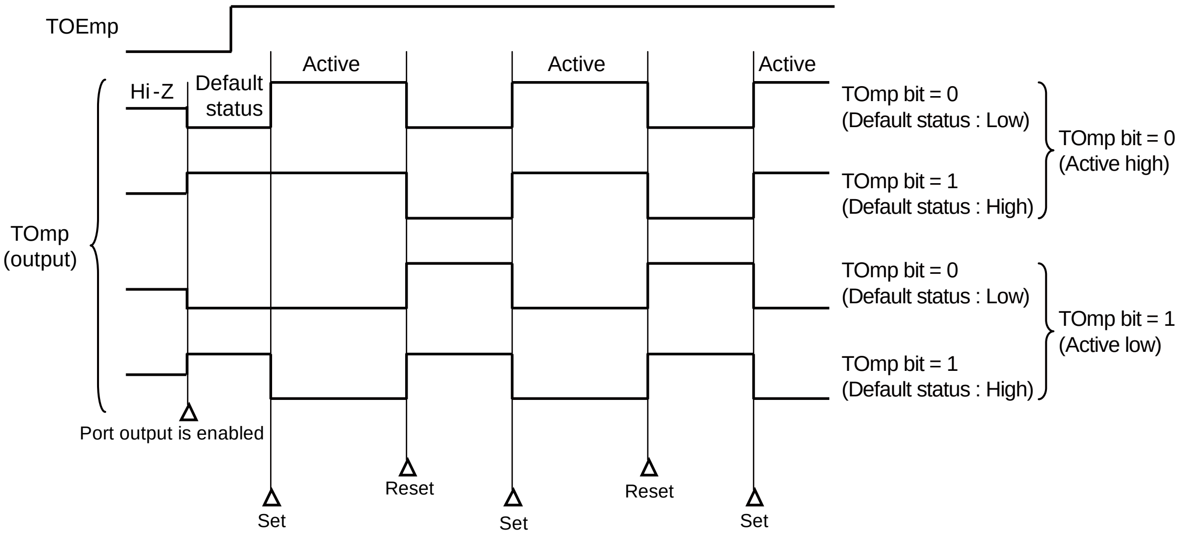
Remarks 1. Toggle: Reverse TOmn pin output status

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 8-37. TOmp Pin Output Status at PWM Output (TOMmp = 1)


TOEmp
Hi-Z
Default status
Active
Active
Active
TOmp (output)
TOmp bit = 0 (Default status : Low)
TOmp bit = 1 (Default status : High)
TOmp bit = 0 (Active high)
TOmp bit = 0 (Default status : Low)
TOmp bit = 1 (Default status : High)
TOmp bit = 1 (Active low)
Port output is enabled
Set
Reset
Set
Reset
Set

Remarks 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.

2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

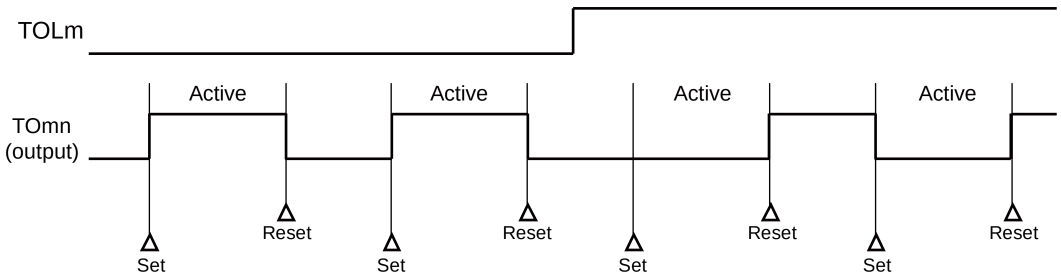
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been 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 8-38. Operation When TOLm Register Has Been Changed Contents During Timer Operation



Remarks 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.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

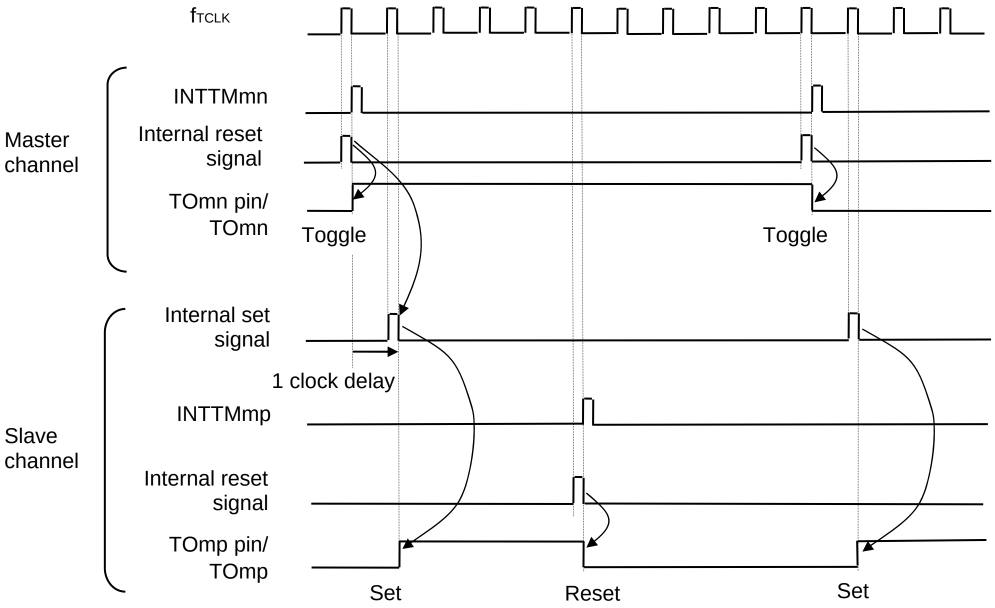
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 8-39 shows the set/reset operating statuses where the master/slave channels are set as follows.

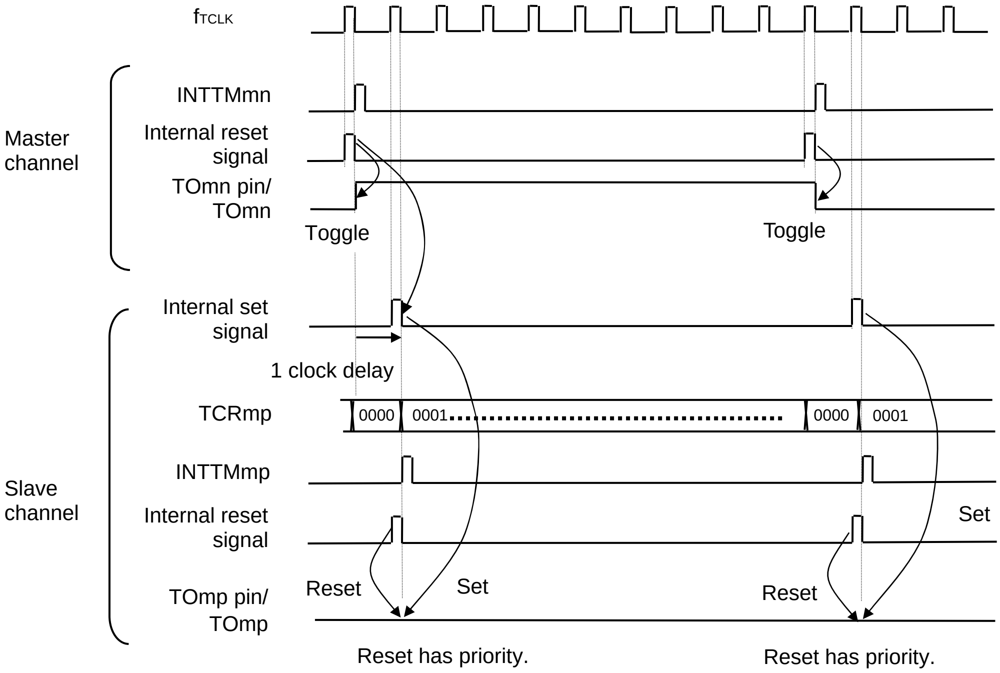
Master channel:	TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel:	TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 8-39. Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remarks 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

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

8.6.4 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 8-40 Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	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	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	1	1
								○	○	×	○	×	×	×	×

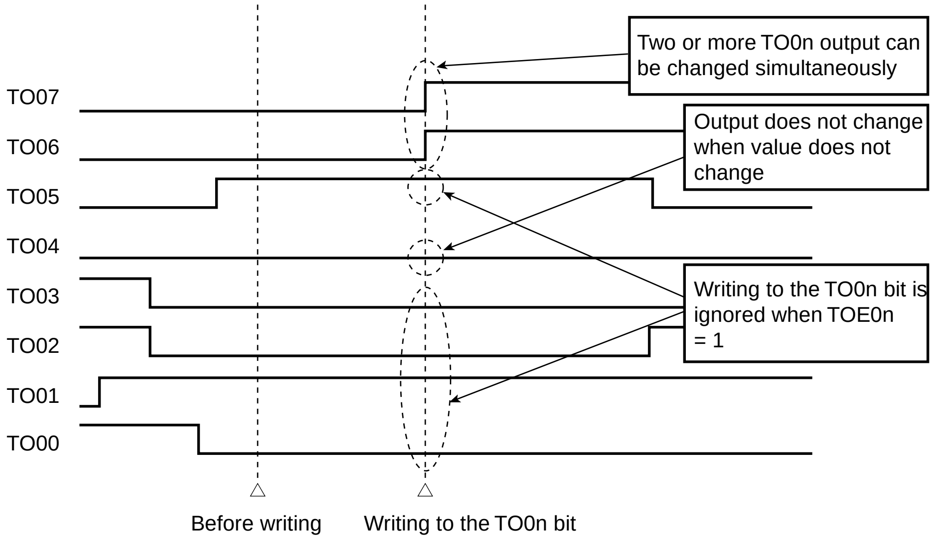
After writing

TO0	0	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 8-41. TO0n Pin Statuses by Collective Manipulation of TO0n Bit


Two or more TO0n output can be changed simultaneously
Output does not change when value does not change
Writing to the TO0n bit is ignored when TOE0n = 1
TO07
TO06
TO05
TO04
TO03
TO02
TO01
TO00
Before writing
Writing to the TO0n bit

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.6.5 Timer Interrupt and TOmn pin output at operation start

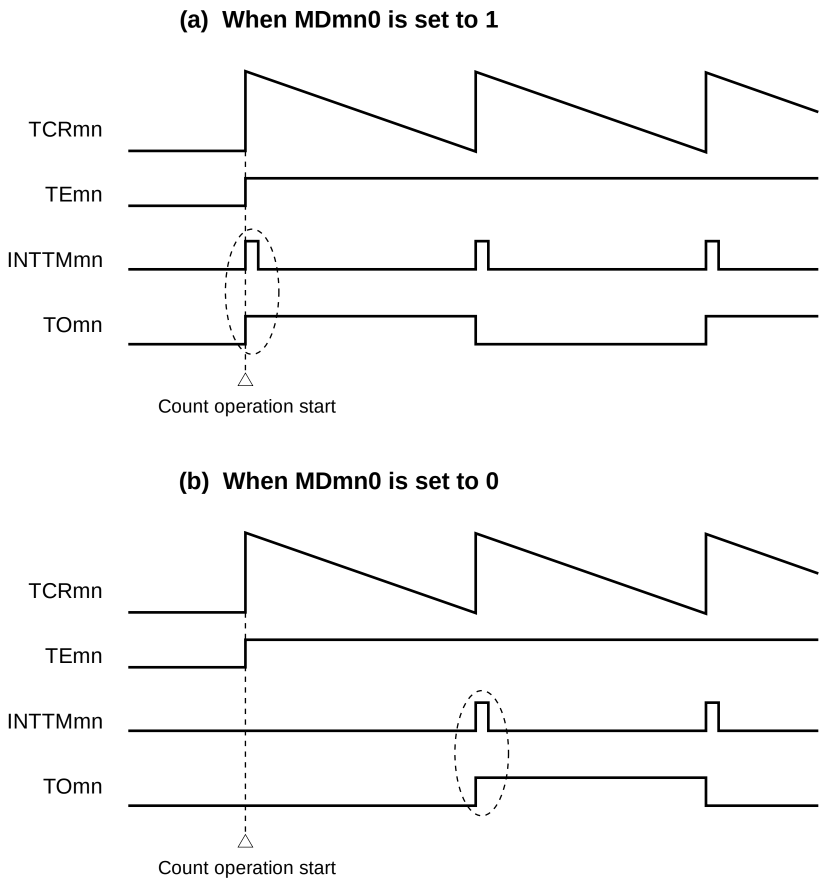
In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

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 8-42 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 8-42. Operation Examples of Timer Interrupt at Count Operation Start and TOmn Output



(a) When MDmn0 is set to 1

(b) When MDmn0 is set to 0

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.

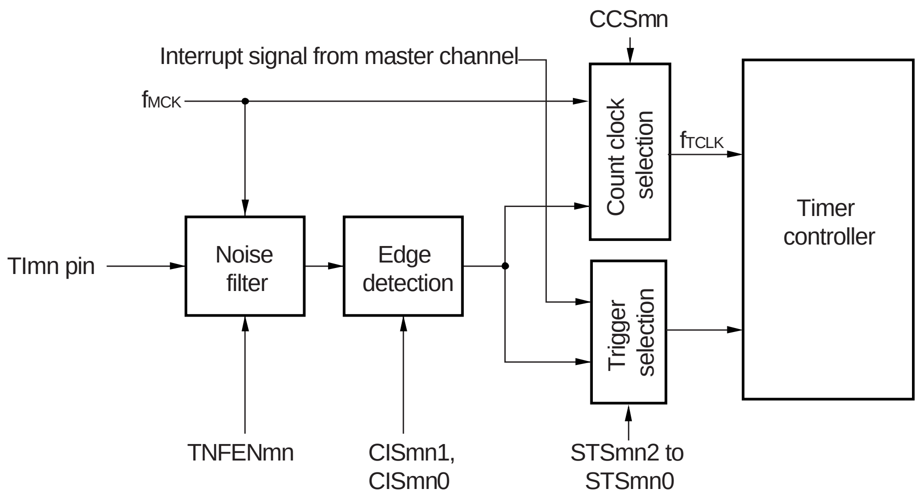
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

8.7 Timer Input (TImn) Control

8.7.1 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.

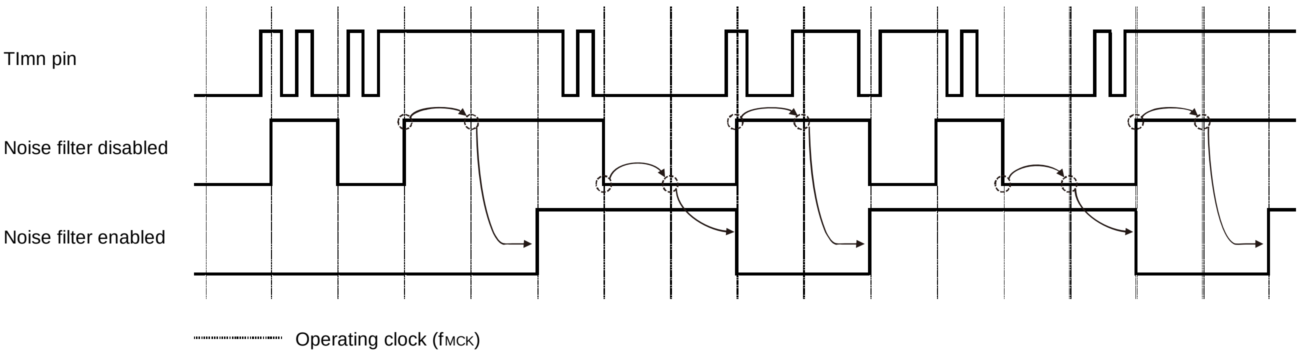
Figure 8-43. Input Circuit Configuration



8.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) 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 8-44. Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution The TImn pin input waveform is shown to explain the noise filter ON/OFF operation. For actual operation, refer to the high-level width/low-level width in 43.4 AC Characteristics.

8.7.3 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 (f_{MCK}), 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 (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

8.8 Independent Channel Operation Function of Timer Array Unit

8.8.1 Operation as interval timer/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.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

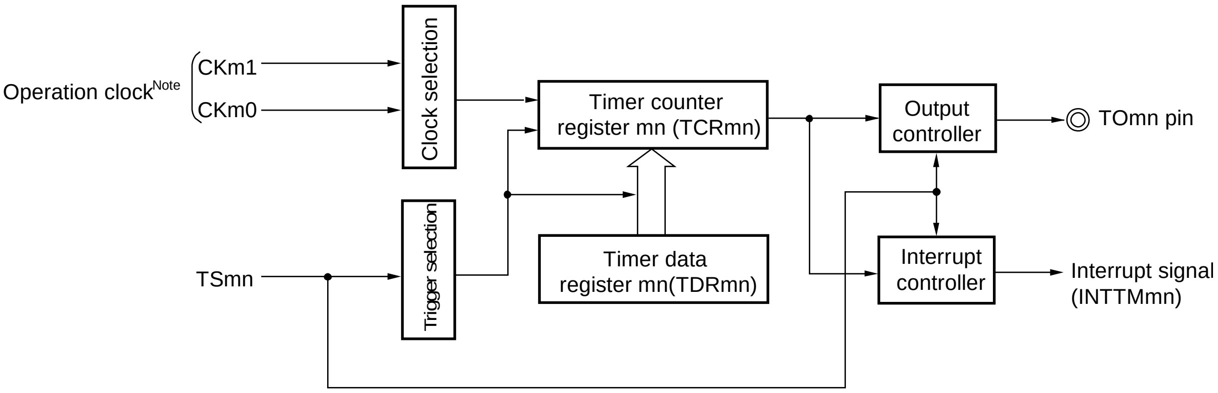
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

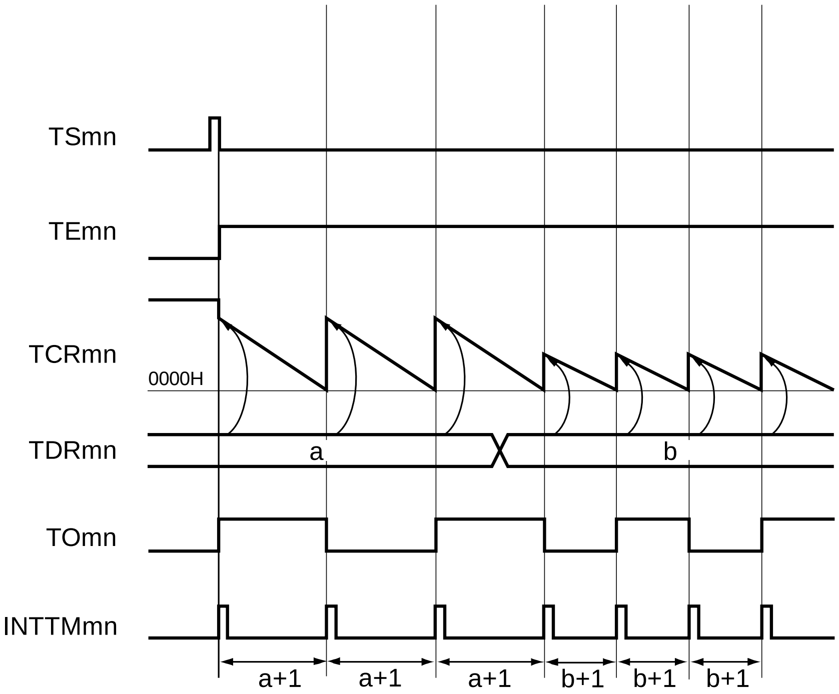
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-45. Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 8-46. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remarks **1.** m: Unit number (m = 0), n: Channel number (n = 0 to 7)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

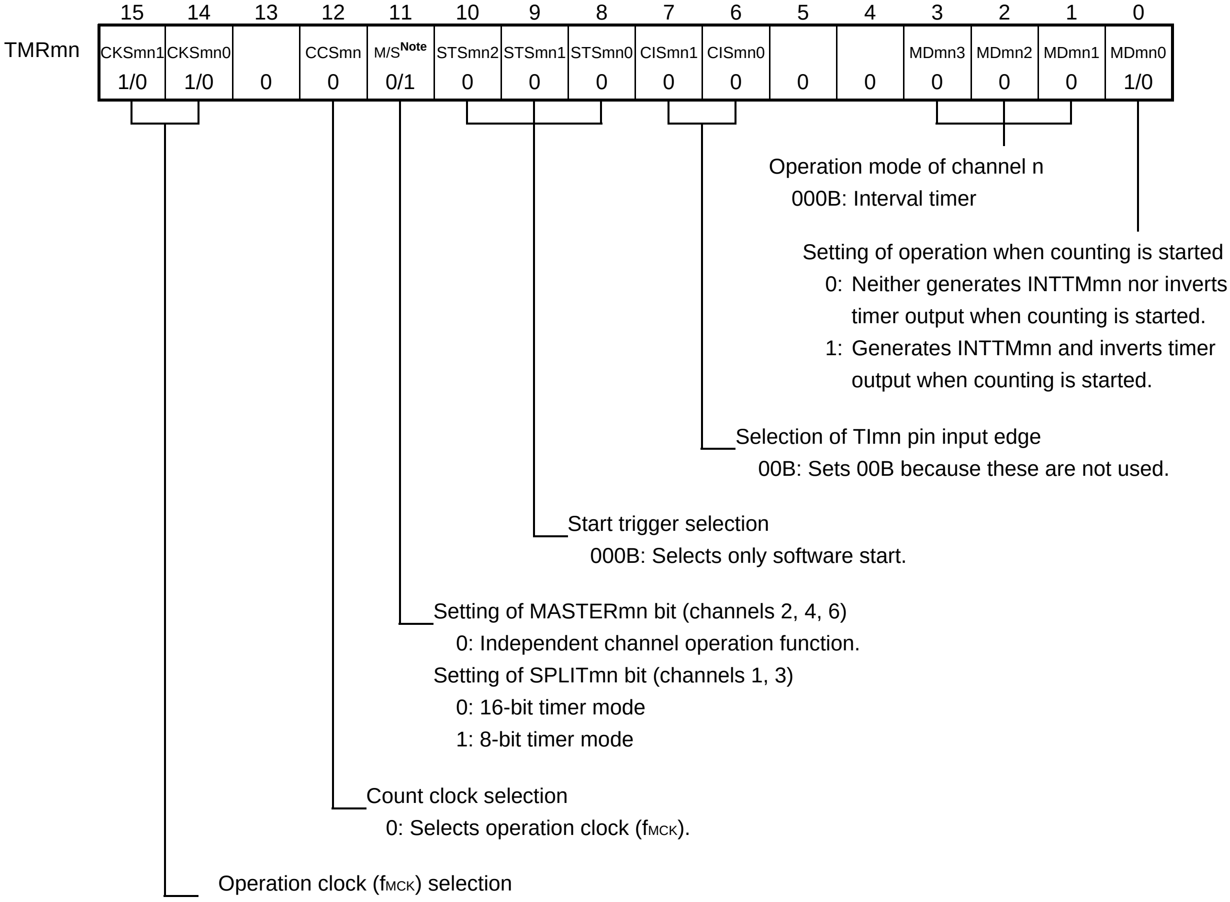
TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

TOmn: TOmn pin output signal

Figure 8-47. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 1/0	 0	CCSmn 0	M/S[Note] 0/1	STSmn2 0	STSmn1 0	STSmn0 0	CISmn1 0	CISmn0 0	 0	 0	MDmn3 0	MDmn2 0	MDmn1 0	MDmn0 1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

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 8-47. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-48. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. 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. Clears the port register and port mode register to 0.	TOmn does not change because channel stops operating. 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 count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
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 TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot 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), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit.	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

(**Remark** is listed on the next page.)

Figure 8-48. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
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.	Power-off status 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)

8.8.2 Operation as 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.

Specified number of counts = Set value of TDRmn + 1

Timer count register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.

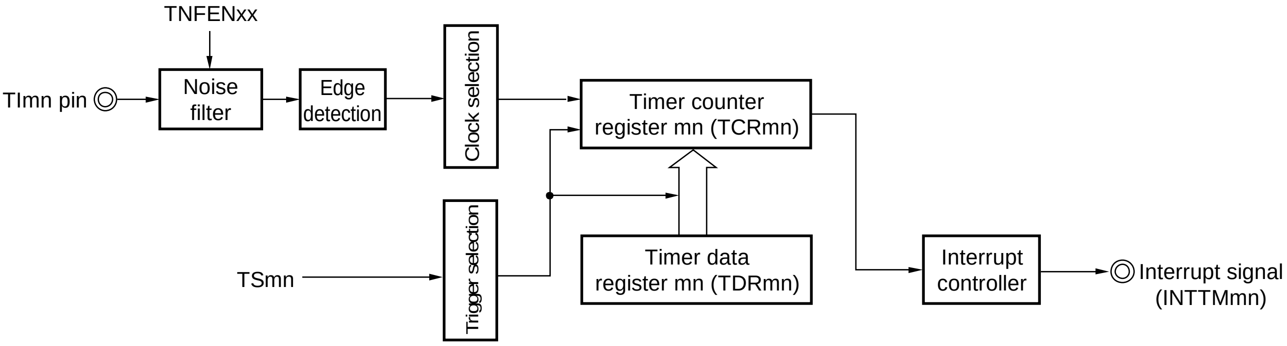
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

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.

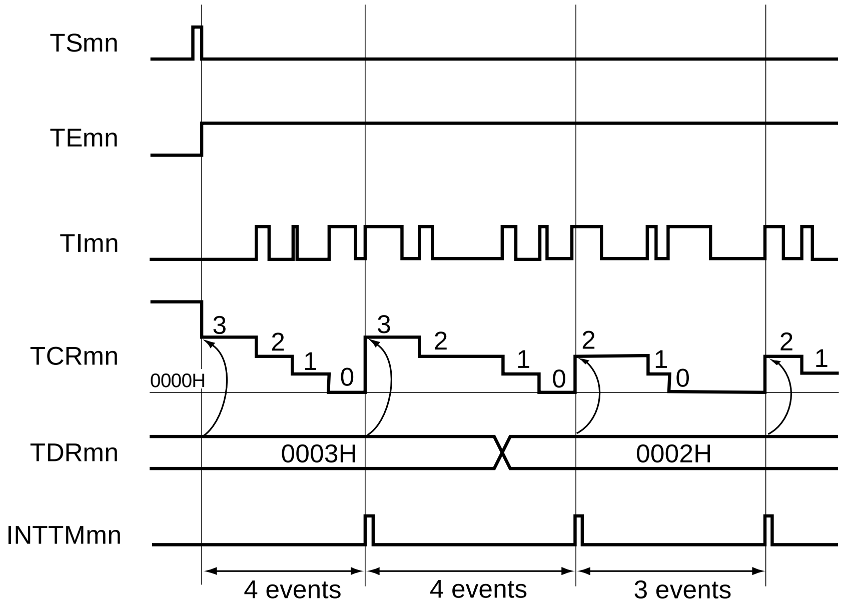
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 8-49. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-50. Example of Basic Timing of Operation as External Event Counter



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

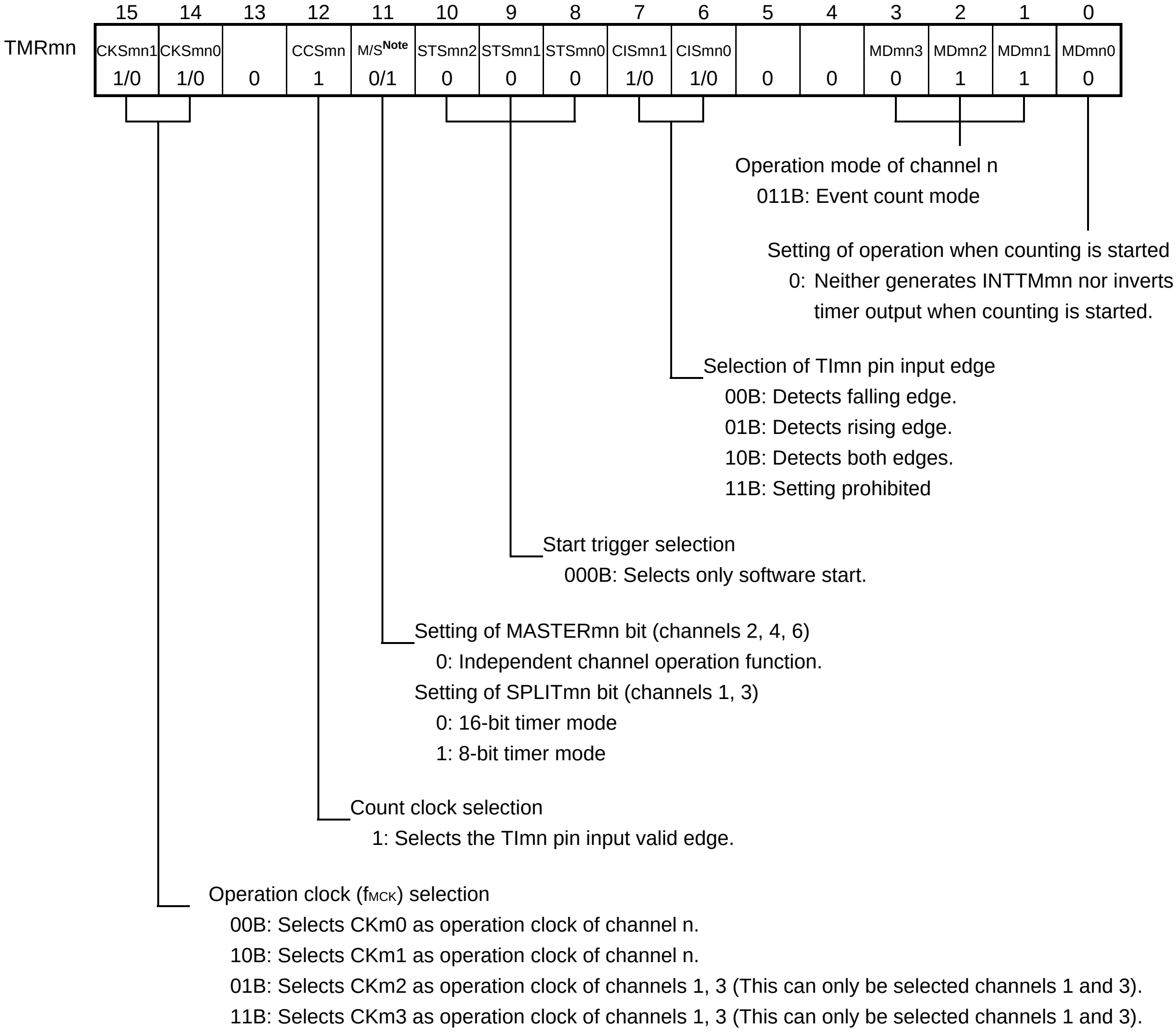
TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 8-51. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

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 8-51. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-52. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation 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 count 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. →	Power-off status 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)

8.8.3 Operation as 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.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

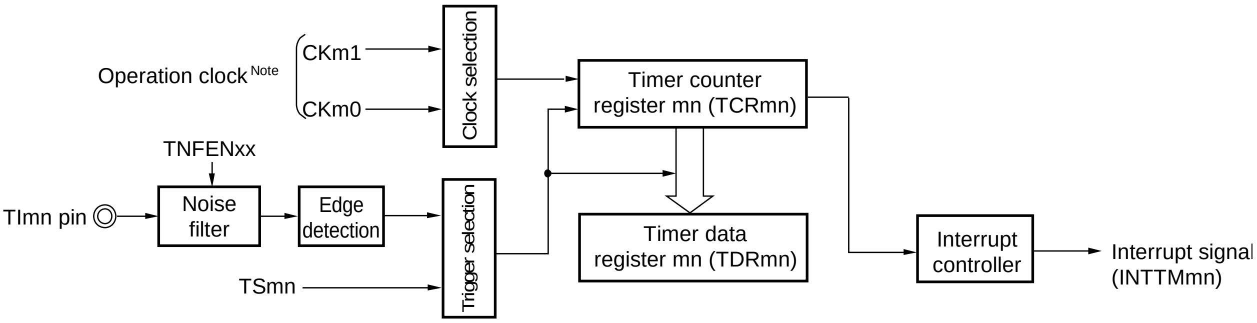
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status 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.

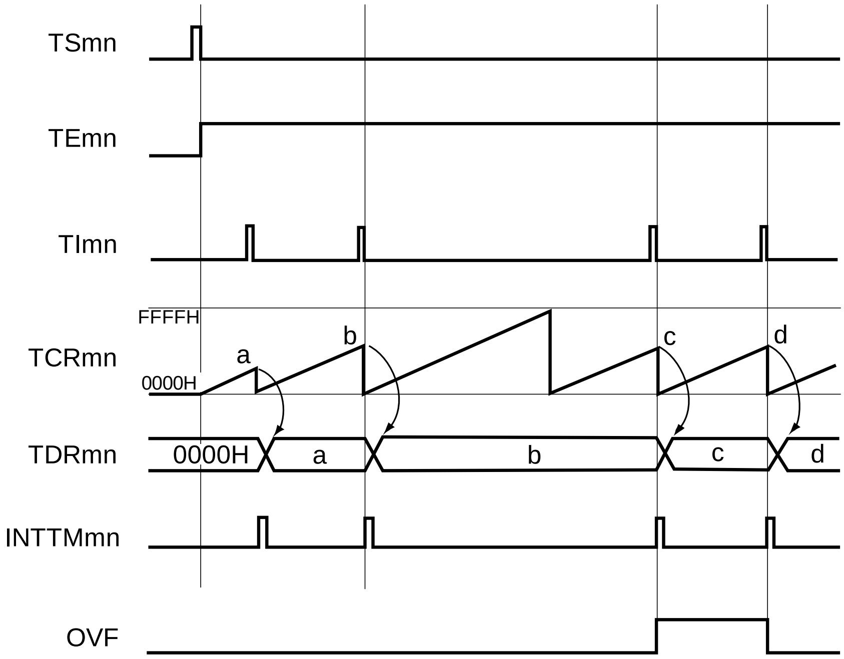
Figure 8-53. Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-54. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 8-55. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	 0	CCSmn 0	M/S[Note] 0	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	 0	 0	MDmn3 0	MDmn2 1	MDmn1 0	MDmn0 1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-56. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count 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 TImn pin input valid edge 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.	Power-off status 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)

8.8.4 Operation as 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 RxD0.

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.

Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture 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 operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status 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 status of the captured value can be checked.

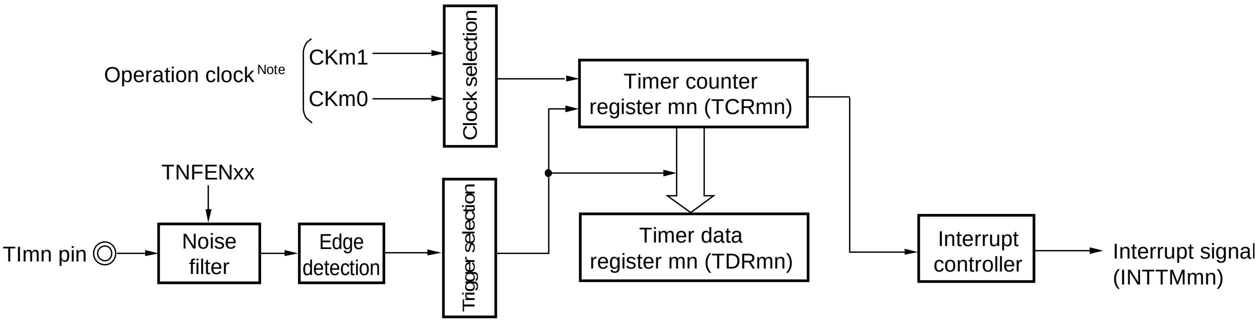
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

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.

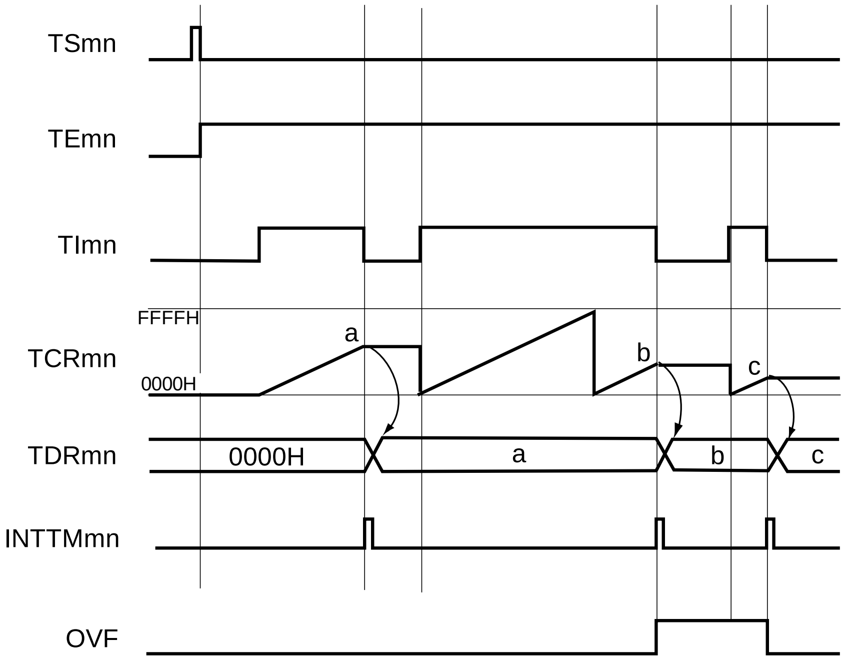
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 8-57. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 8-58. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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

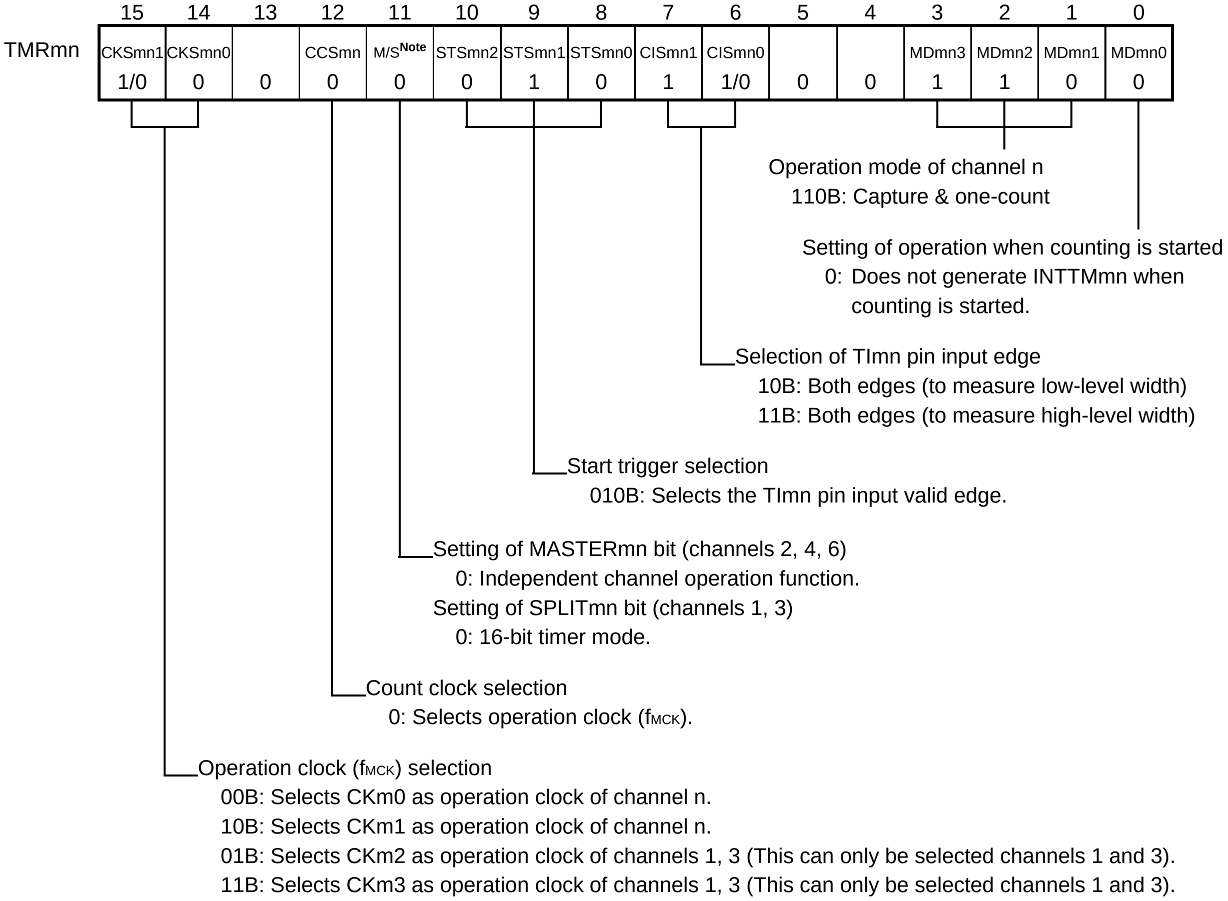
TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 8-59. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-60. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge.	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
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.	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.	Power-off status 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)

8.8.5 Operation as 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 can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

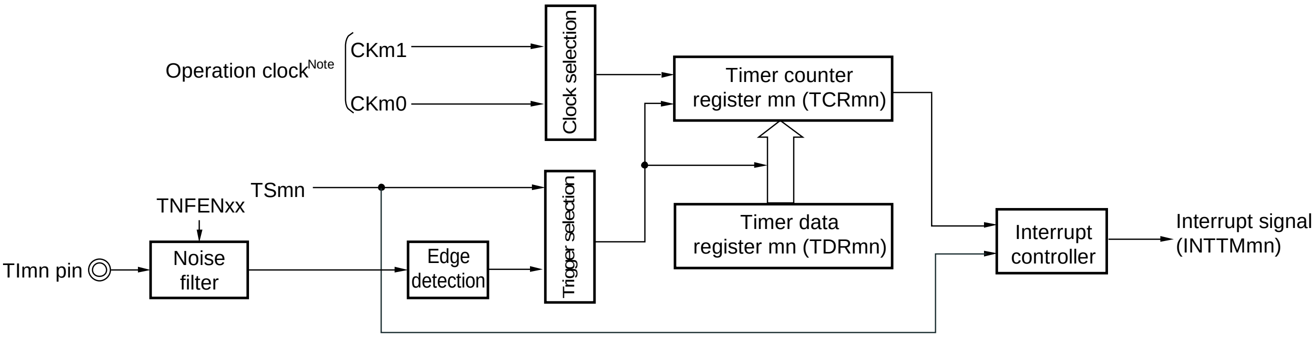
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

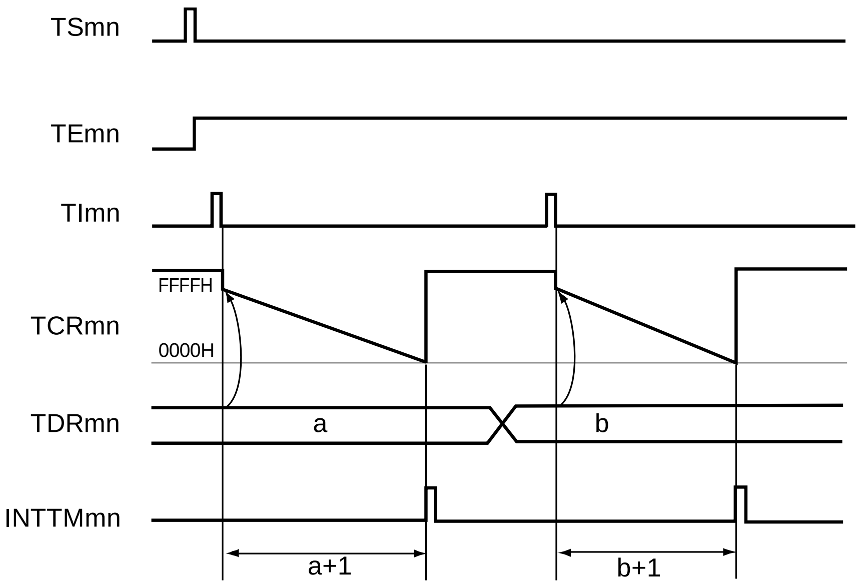
Figure 8-61. Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-62. Example of Basic Timing of Operation as Delay Counter



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

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 count register mn (TCRmn)
TDRmn:	Timer data register mn (TDRmn)

Figure 8-63. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.
1: 8-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

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 8-63. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 8-64. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status 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 count 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. →	Power-off status 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)

8.9 Simultaneous Channel Operation Function of Timer Array Unit

8.9.1 Operation as 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.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

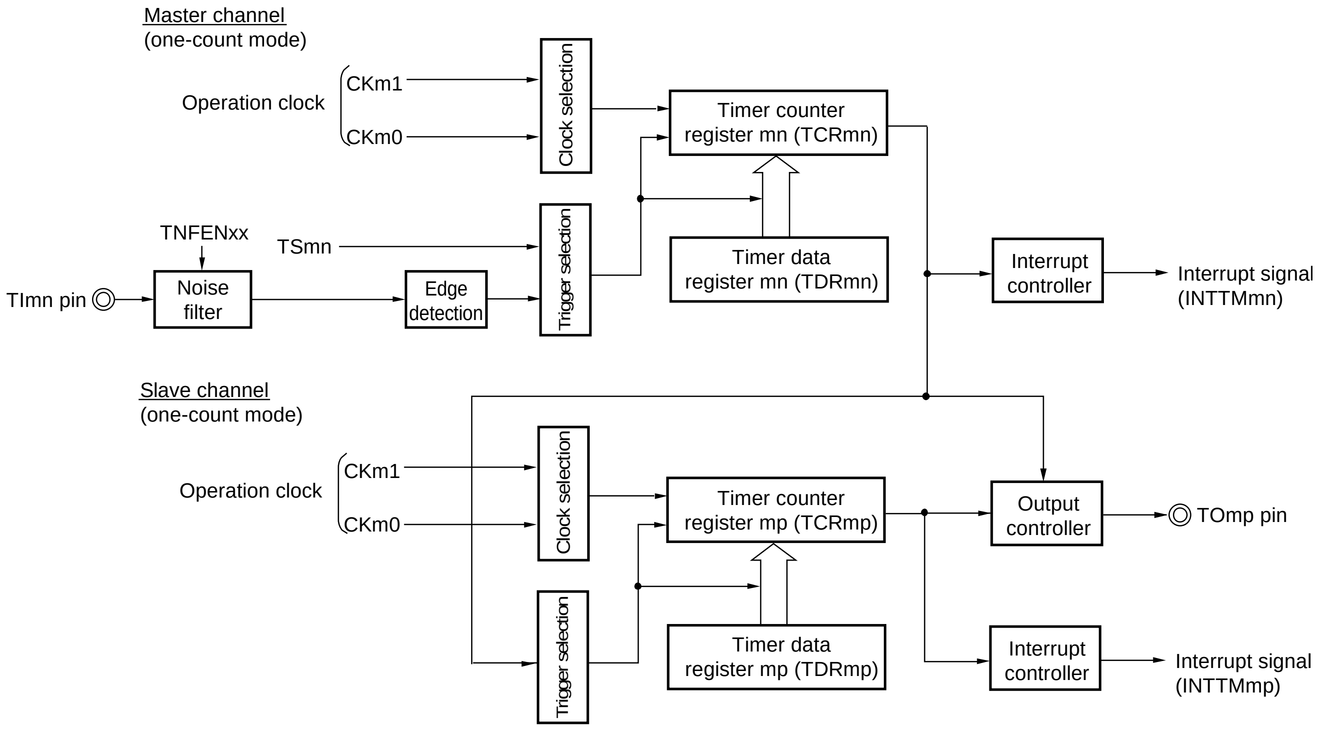
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during count operation, therefore, an illegal waveform may be output by conflicting with load timing. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.

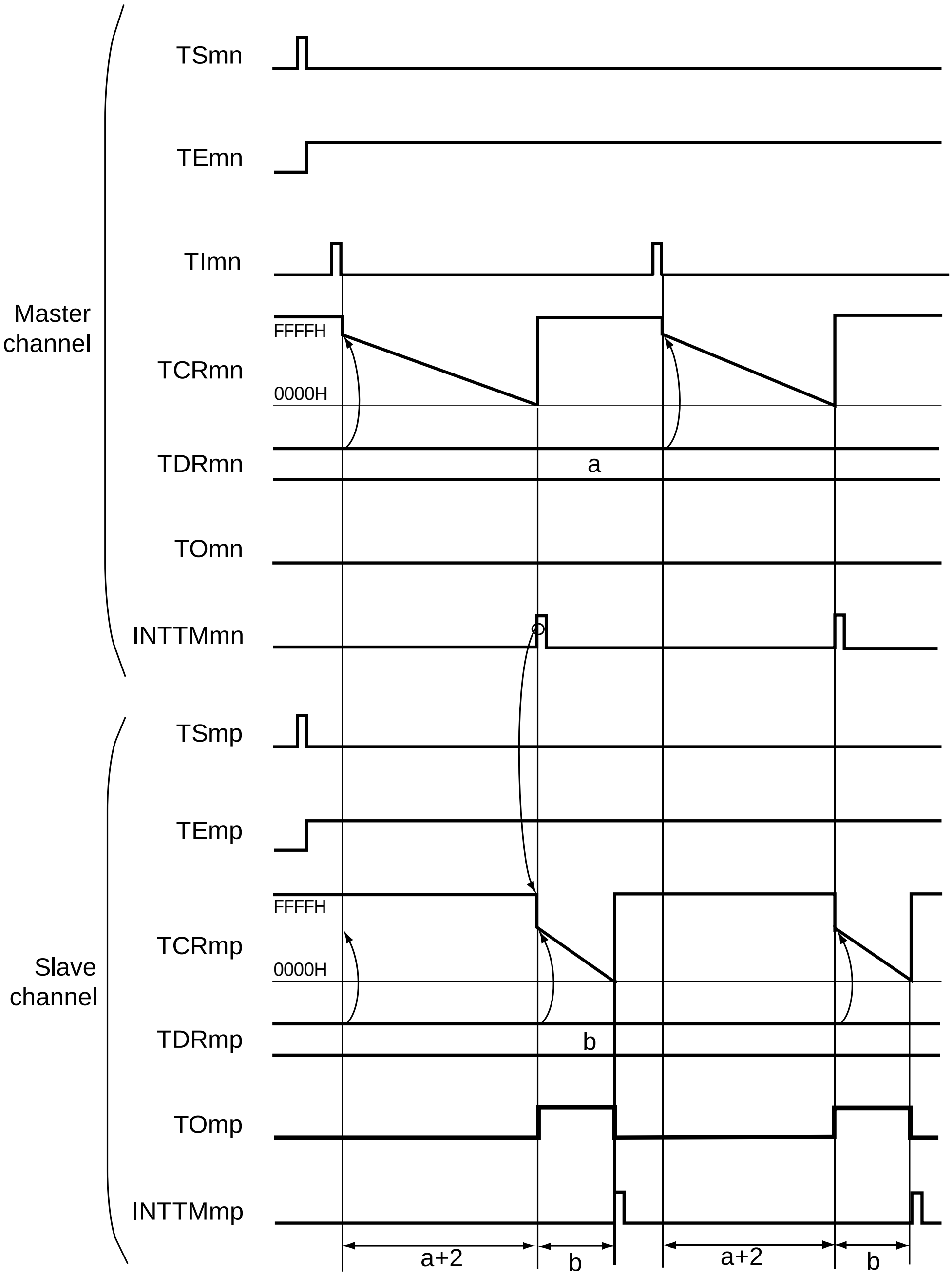
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 8-65. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 8-66. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

p: Slave channel number (n < p ≤ 7)

2.

TSmn, TSmp:	Bit n, p of timer channel start register m (TSm)
TEmn, TEmp:	Bit n, p of timer channel enable status register m (TEm)
TImn, TImp:	TImn and TImp pins input signal
TCRmn, TCRmp:	Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp:	Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp:	TOmn and TOmp pins output signal

Figure 8-67. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 4, 6)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

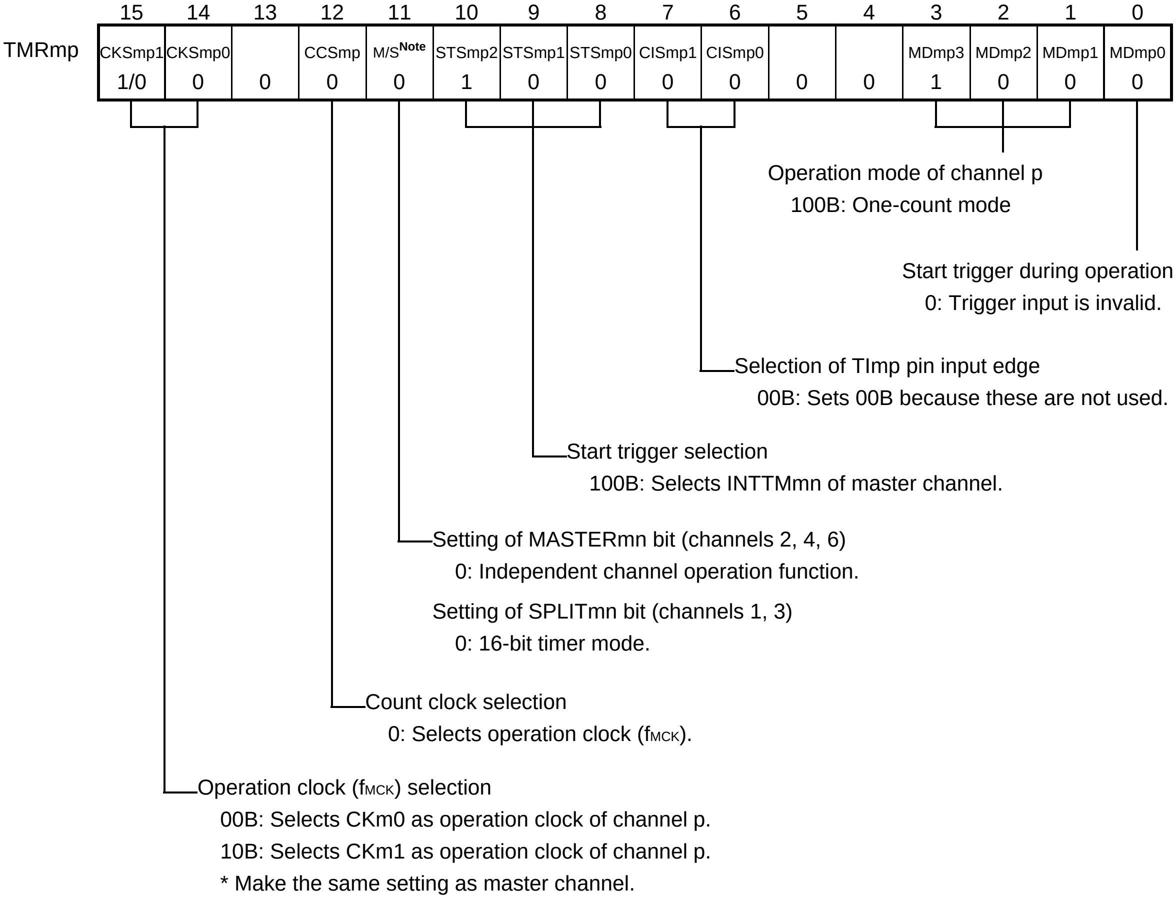
0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

Figure 8-68. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1 1/0	CKSmp0 0	 0	CCSmp 0	M/S[Note] 0	STSmp2 1	STSmp1 0	STSmp0 0	CISmp1 0	CISmp0 0	 0	 0	MDmp3 1	MDmp2 0	MDmp1 0	MDmp0 0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp 1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp 1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp 1

1: Sets the slave channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number ($n < p \leq 7$)

Figure 8-69. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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 status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 1 (on). 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.	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.
	Sets the TOEmp bit to 1 and enables operation of TOmp. 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.

(**Remark** is listed on the next page.)

Figure 8-69. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
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 TEmp 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 status. 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 software[Note]. **Note** Do not set the TSmn bit of the slave channel to 1.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	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.	Power-off status 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.)

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

8.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

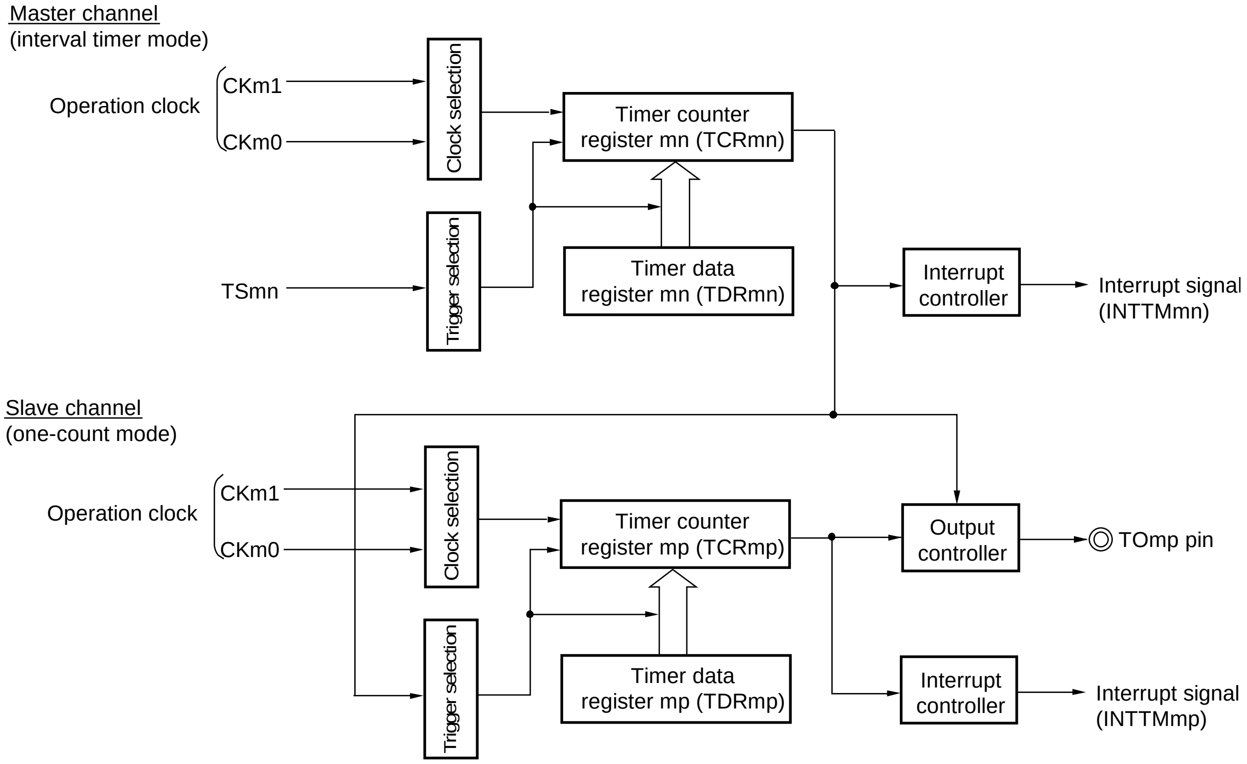
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

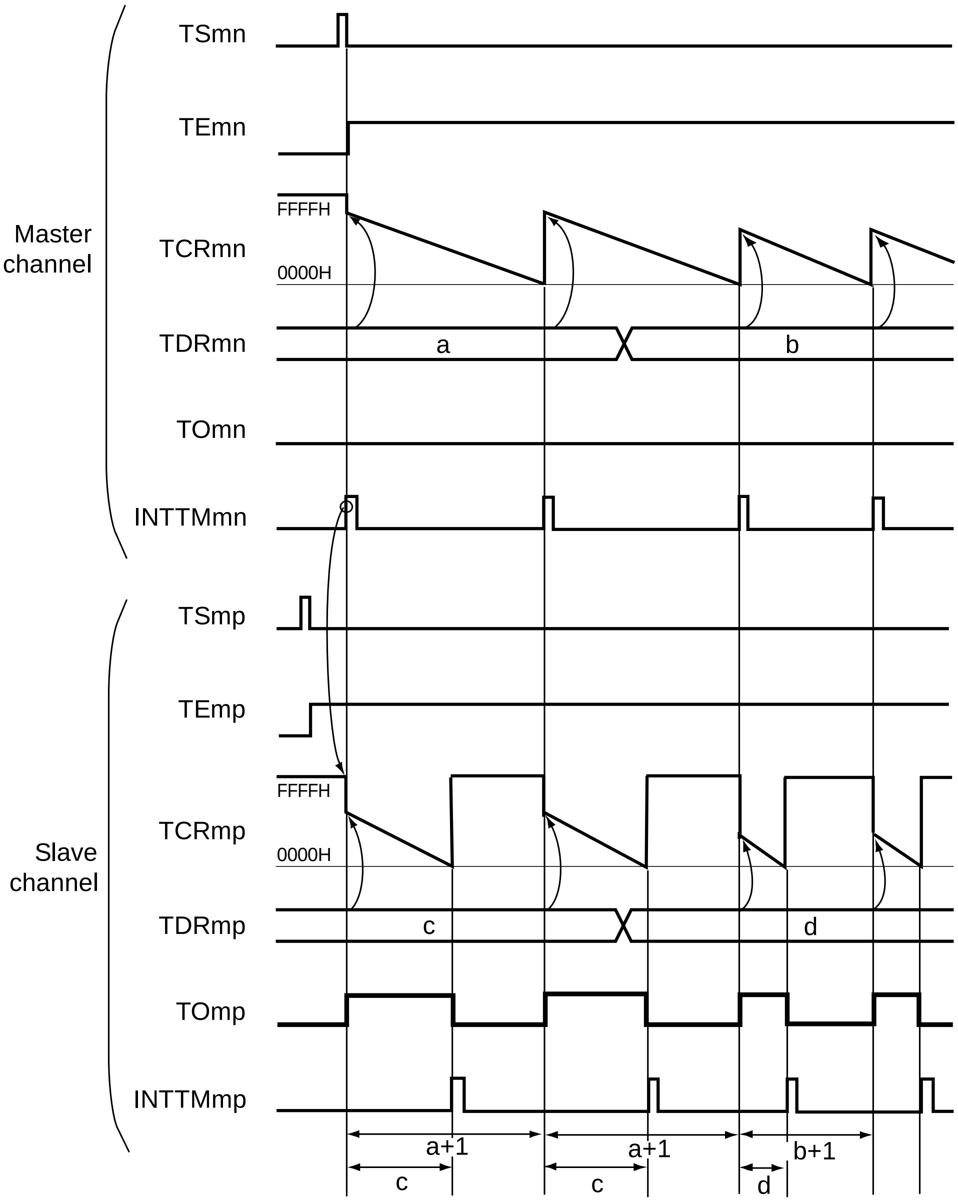
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 8-70. Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 8-71. Example of Basic Timing of Operation as PWM Function

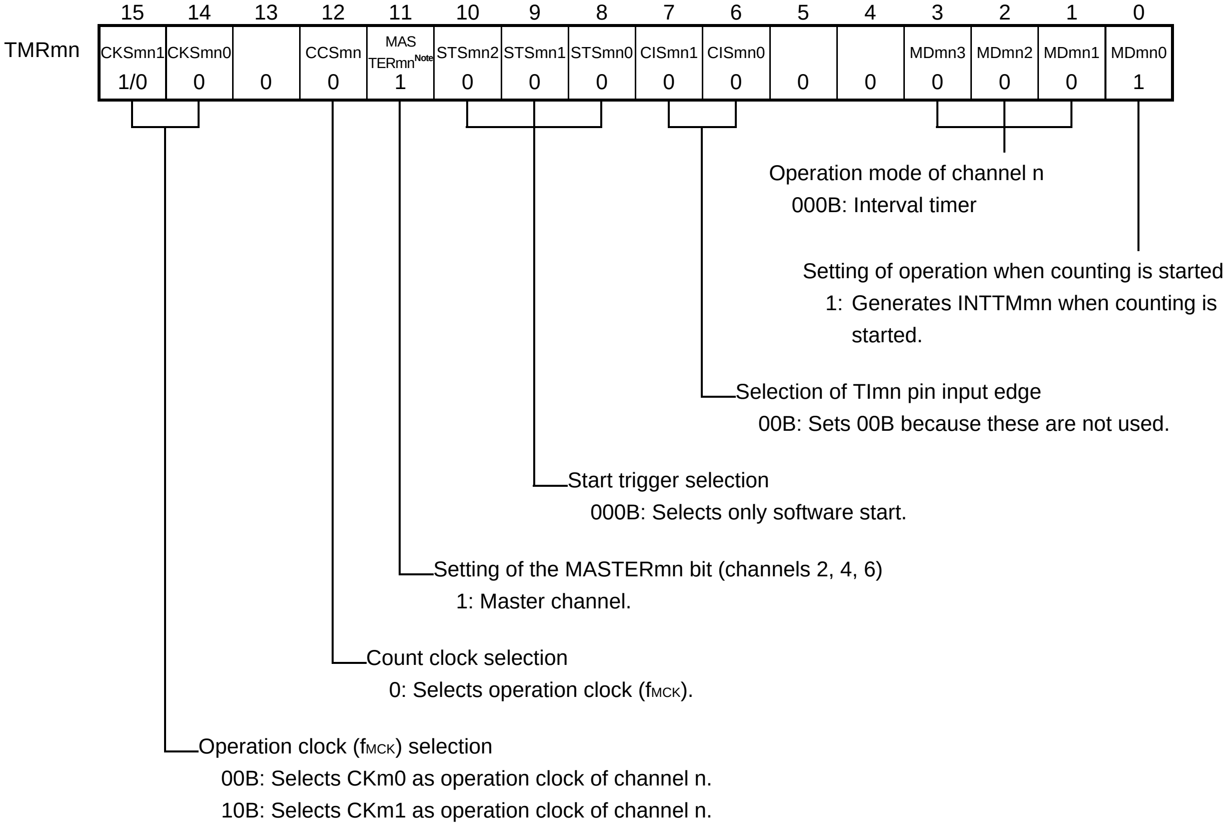


Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 8-72. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of the MASTERmn bit (channels 2, 4, 6)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

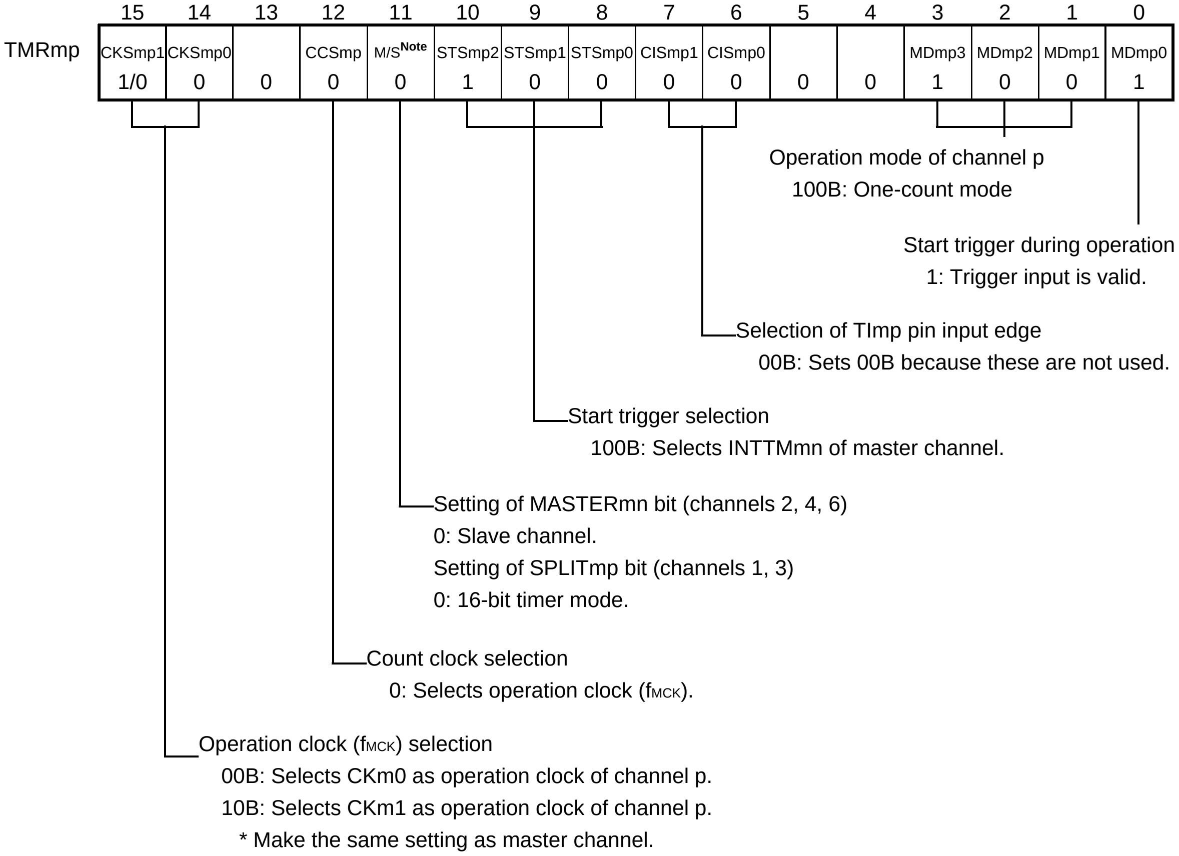
0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)

Figure 8-73. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S[Note]	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Slave channel.
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the slave channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 8-74. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Stops supply of timer array unit m input clock. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Supplies timer array unit m input clock. 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.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output.	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp.	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOmp pin outputs the TOmp set level.

(**Remark** is listed on the next page.)

Figure 8-74. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
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 count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	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.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

8.9.3 Operation as 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.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

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 count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

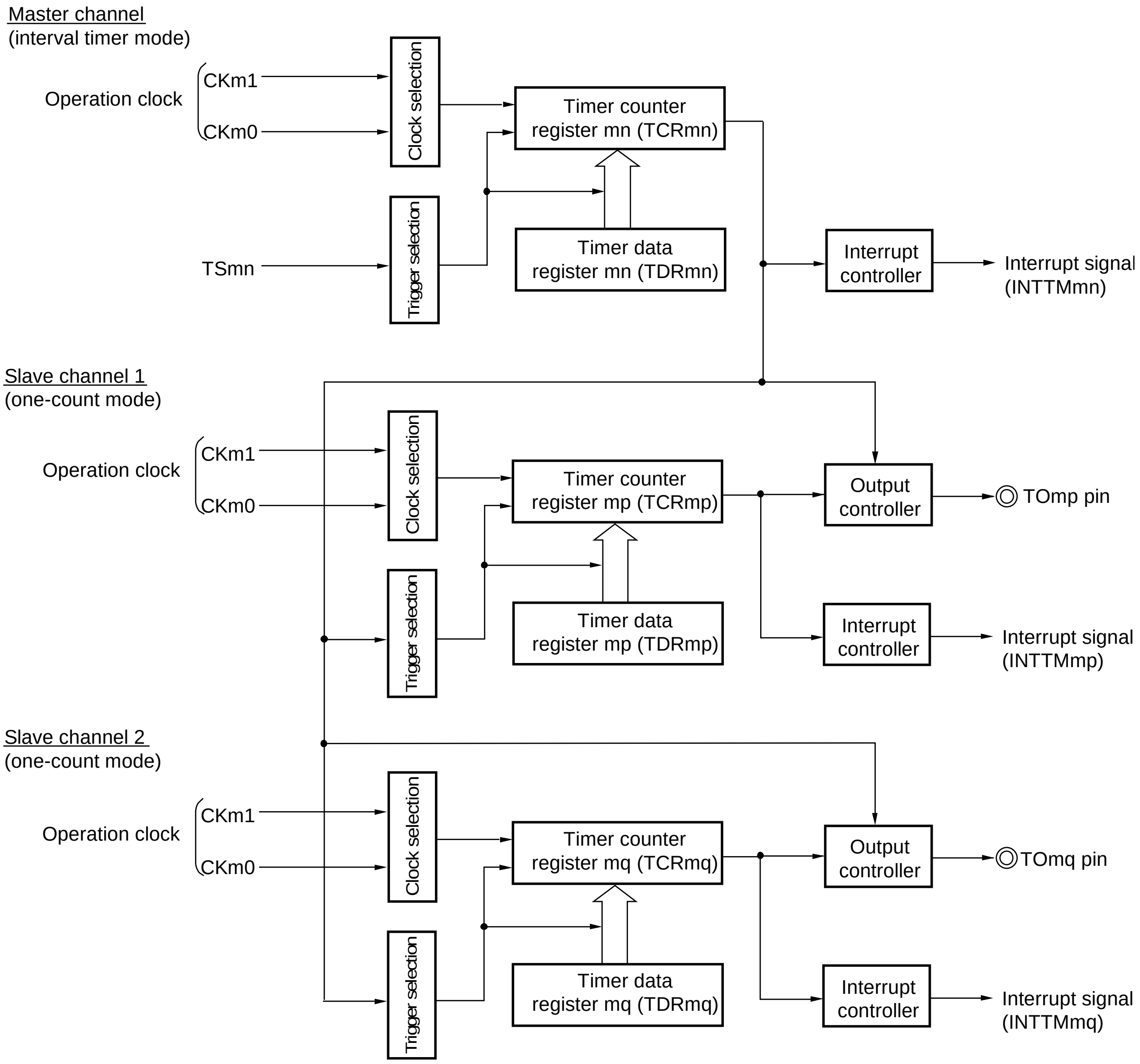
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).

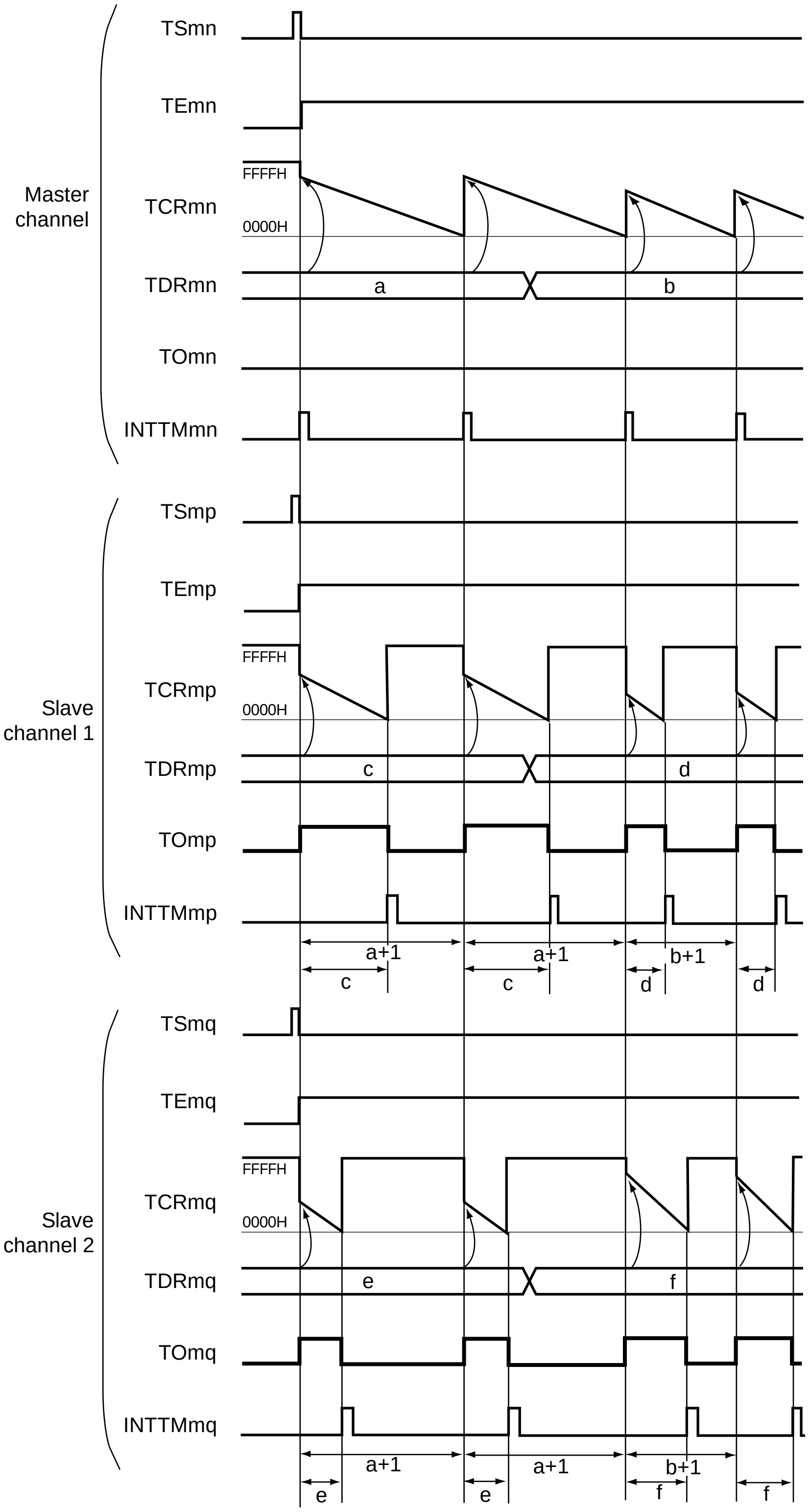
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 8-75. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 8-76. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



(**Remark** is listed on the next page.)

Remarks 1. m: Unit number ($m = 0$), n: Channel number ($n = 0, 2, 4$)
p: Slave channel number, q: Slave channel number
$n < p < q \le 7$ (Where p and q are integers greater than n)

2.

TSmn, TSmp, TSmq:	Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq:	Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq:	Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq:	Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq:	TOmn, TOmp, and TOmq pins output signal

Figure 8-77. Example of Set Contents of Registers
When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of the MASTERmn bit (channels 2, 4, 6)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)

Figure 8-78. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S[Note]	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1	CKSmq0		CCSmq	M/S[Note]	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp and MASTERmq bits (channels 2, 4, 6)
0: Slave channel.

Setting of SPLITmp and SPLITmq bits (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	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	TOEmp
	1/0	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	TOLmp
	1/0	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	TOMmp
	1	1

1: Sets the slave channel output mode.

Note		
	TMRm2, TMRm4, TMRm6:	MASTERmp, MASTERmq bit
	TMRm1, TMRm3:	SPLITmp, SPLIT0q bit
	TMRm5, TMRm7:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 8-79. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Stops supply of timer array unit m input clock. (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Supplies timer array unit m input clock. 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, mq (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.	The TOmp and TOmq pins go into Hi-Z output state.
	Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs.	The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq.	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0.	The TOmp and TOmq pins output the TOmp and TOmq set levels.

(**Remark** is listed on the next page.)

Figure 8-79. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
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, TEmp, 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 count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	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.	Power-off status 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: Channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

8.10 Cautions When Using Timer Array Unit

8.10.1 Cautions when using timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

For details, see **4.5 Register Settings When Using Alternate Function**.

(a) Using TO01 to TO07 outputs (80-pin product only)

In addition to clearing the port mode register (the PMxx bit) and the port register (the Pxx bit) to 0, be sure to set the corresponding bit from among PFSEG27 of LCD port function register 4 (PFSEG4) and PFSEG32 to PFSEG37 of LCD port function register 5 (PFSEG5) to 0.

(b) Using TO00 and TO01 outputs assigned to the P43 and P42

So that the alternated PCLBUZ1 and PCLBUZ0 outputs become 0, not only set the port mode register (the PM43 and PM42 bits) and the port register (the P43 and P42 bits) to 0, but also use the bit 7 of the clock output select register n (CKSn) with the same setting as the initial status.

(c) Using TO02 to TO07 outputs assigned to the P07 to P02

So that the alternated P07/SO00/TxD0, P06/SDA00, P05/SCK00/SCL00, P04/TxD1, P03/SDA10 and P02/SCL10 outputs become 1, not only set the port mode register (the PM07 to PM02 bits) and the port register (the P07 to P02 bits) to 0, but also use the serial channel enable status register 0 (SE0), serial output register 0 (SO0), and serial output enable register 0 (SOE0) with the same setting as the initial status.

(d) Using TO06 outputs assigned to the P31

So that the alternated P06/TxD2/IrTxD outputs become 1, not only set the port mode register (the PM06 bit) and the port register (the P06 bit) to 0, but also use the serial channel enable status register 0 (SE0), serial output register 0 (SO0), and serial output enable register 0 (SOE0) with the same setting as the initial status.

CHAPTER 9 REALTIME CLOCK WITH INDEPENDENT POWER SUPPLY

9.1 Overview

The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.

For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.

For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.

The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the basic clock. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

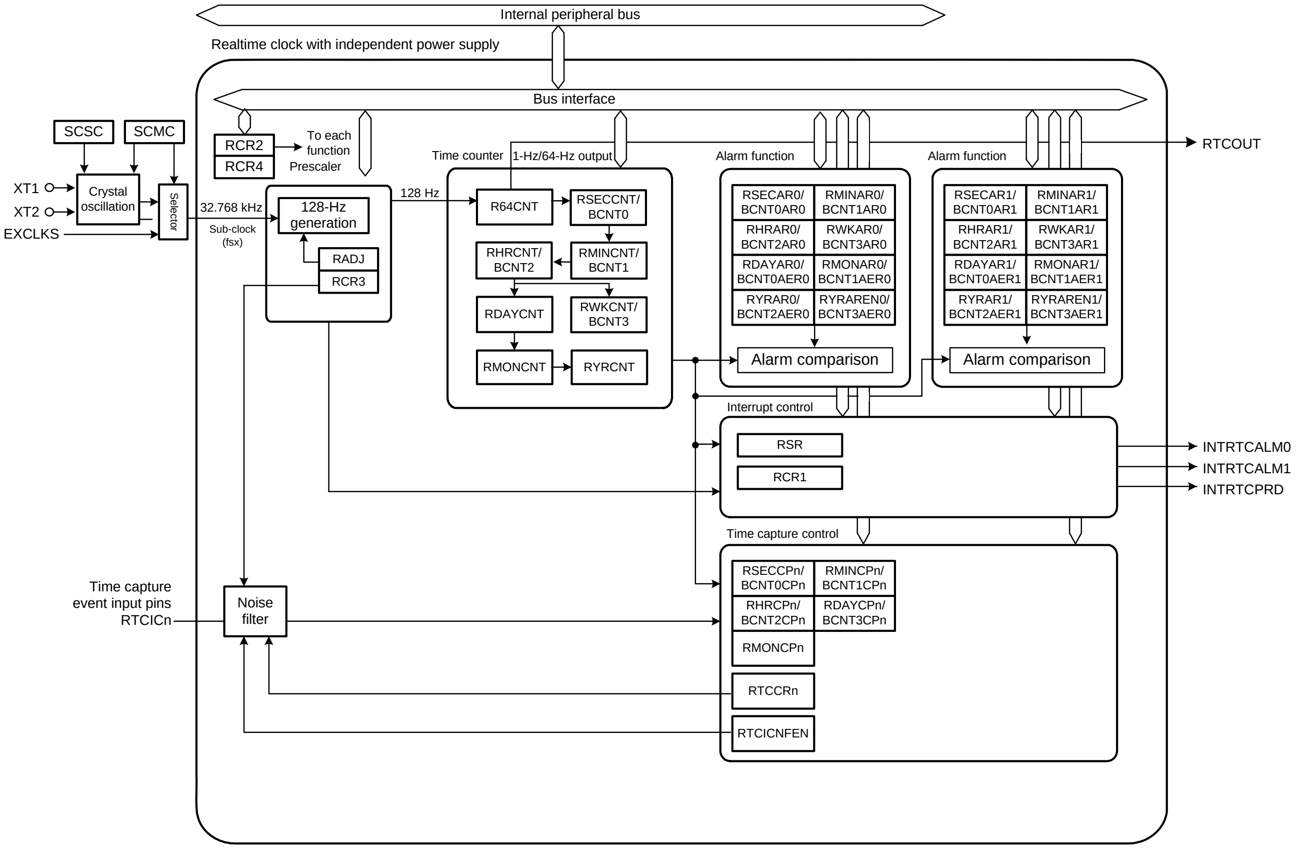
Table 9-1 lists the specifications of the RTC, shows a block diagram of the RTC, and **Table 9-2** shows the pin configuration of the RTC.

Table 9-1. RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Count source	Sub-clock (f_{SX}) **Note**
Clock and calendar functions	● Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30-second adjustment function: If the value of the lowest-level counter is equivalent to less than 30 seconds, it is rounded down to 00 seconds, and if it is equivalent to 30 seconds or more, it is rounded up to one minute. 0.5-second adjustment function: If the value of the lowest-level counter is equivalent to less than 0.5 of a second, it is rounded down to 00 seconds, and if it is equivalent to 0.5 of a second or more, it is rounded up to one second. Automatic adjustment function for leap years ● Binary count mode Count seconds in 32 bits, binary display ● Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1-Hz/64-Hz) output
Interrupts	● Alarm interrupt (ALM) – As an alarm interrupt condition, selectable which of the below is compared with: Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected Binary count mode: Each bit of the 32-bit binary counter – Two alarm times can be set. ● Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, or 1/256 second can be selected as an interrupt period. ● Recovery from standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	● Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output

Note The XT1 clock oscillator runs on the VRTC power-supply. It can operate after release from the RTC power-on reset following the power supply to the VRTC pin being turned on. If the VRTC power-supply stops, an RTC power-on reset is generated and the XT1 clock oscillator stops.

Figure 9-1. Block Diagram of RTC



R64CNT:	64-Hz counter
RSECCNT/BCNT0:	Second counter/Binary counter 0
RMINCNT/BCNT1:	Minute counter/Binary counter 1
RHRCNT/BCNT2:	Hour counter/Binary counter 2
RWKCNT/BCNT3:	Day-of-week counter/Binary counter 3
RDAYCNT:	Date counter
RMONCNT:	Month counter
RYRCNT:	Year counter
RSR:	Status register
RCR1:	RTC control register 1
RCR2:	RTC control register 2
RCR3:	RTC control register 3
RCR4:	RTC control register 4
RADJ:	Time error adjustment register
RTCCRn:	Time capture control register n
RSECCPn/BCNT0CPn:	Second capture register n/BCNT0 capture register n
RMINCPn/BCNT1CPn:	Minute capture register n/BCNT1 capture register n
RHRCPn/BCNT2CPn:	Hour capture register n/BCNT2 capture register n
RDAYCPn/BCNT3CPn:	Date capture register n/BCNT3 capture register n
RMONCPn:	Month capture register n

RSECAR0/BCNT0AR0:	Second alarm register 0/Binary counter 0 alarm register 0
RMINAR0/BCNT1AR0:	Minute alarm register 0/Binary counter 1 alarm register 0
RHRAR0/BCNT2AR0:	Hour alarm register 0/Binary counter 2 alarm register 0
RWKAR0/BCNT3AR0:	Day-of-week alarm register 0/Binary counter 3 alarm register 0
RDAYAR0/BCNT0AER0:	Date alarm register 0/Binary counter 0 alarm enable register 0
RMONAR0/BCNT1AER0:	Month alarm register 0/Binary counter 1 alarm enable register 0
RYRAR0/BCNT2AER0:	Year alarm register 0/Binary counter 2 alarm enable register 0
RYRAREN0/BCNT3AER0:	Year alarm enable register 0/Binary counter 3 alarm enable register 0
RSECAR1/BCNT0AR1:	Second alarm register 1/Binary counter 0 alarm register 1
RMINAR1/BCNT1AR1:	Minute alarm register 1/Binary counter 1 alarm register 1
RHRAR1/BCNT2AR1:	Hour alarm register 1/Binary counter 2 alarm register 1
RWKAR1/BCNT3AR1:	Day-of-week alarm register 1/Binary counter 3 alarm register 1
RDAYAR1/BCNT0AER1:	Date alarm register 1/Binary counter 0 alarm enable register 1
RMONAR1/BCNT1AER1:	Month alarm register 1/Binary counter 1 alarm enable register 1
RYRAR1/BCNT2AER1:	Year alarm register 1/Binary counter 2 alarm enable register 1
RYRAREN1/BCNT3AER1:	Year alarm enable register 1/Binary counter 3 alarm enable register 1

Remark n = 0 to 2

Table 9-2. Pin Configuration of RTC

Pin Name	I/O	Function
XT1	Input	Connecting a 32.768-kHz crystal vibrator.
XT2	Input	
EXCLKS	Input	Connecting a 32.768-kHz external clock input.
RTCOUT	Output	1 or 64-Hz waveform output pin.
RTCIC0	Input	Time capture event input pins
RTCIC1	Input	
RTCIC2	Input	

9.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with **9.6.4 Notes when writing to and reading from registers**.

If the value in an RTC register after a reset is given as undefined in the list, it is not initialized by a reset. When RTC enters the reset state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. Note that a reset generated during writing to or updating of a register might destroy the register value.

Table 9-3 shows the power domains, the values after different types of reset, and the R/W properties of the registers that control the RTC.

Table 9-3. Registers that Control the RTC (1/3)

Special Function Register (SFR) Name	Bit Name	Power Domain	RTC Power-on Reset	After MCU Reset		RTC Software Reset	R/W Property	
				Power-on Reset	Reset other than RTCPOR [Note 1]		VRTCEN = 0	VRTCEN = 1
PER2	VRTCEN	V_{DD}	–	0	0	–	Readable and writable	Readable and writable [Note 4]
R64CNT	–	VRTC	Undefined	–	–	00H	Not readable	Readable
RSECCNT/BCNT0	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RMINCNT/BCNT1	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RHRCNT/BCNT2	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RWKCNT/BCNT3	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RDAYCNT	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RMONCNT	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RYRCNT	–	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable [Notes 5, 7]
RSECARm/BCNT0ARm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]
RMINARm/BCNT1ARm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]
RHRARm/BCNT2ARm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]
RWKARm/BCNT3ARm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]
RDAYARm/BCNT0AERm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]
RMONARm/BCNT1AERm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable [Note 5]

(**Notes** are listed on the page after the next page.)

Table 9-3. Registers that Control the RTC (2/3)

Special Function Register (SFR) Name	Bit Name	Power Domain	RTC Power-on Reset	After MCU Reset		RTC Software Reset	R/W Property	
				Power-on Reset	Reset other than RTCPOR [Note 1]		VRTCEN = 0	VRTCEN = 1
RYRARm/BCNT2AERm (m = 0, 1)	–	VRTC	Undefined	–	–	0000H	Not readable, not writable	Readable and writable[Note 5]
RYRARENm/BCNT3AERm (m = 0, 1)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable[Note 5]
RCR1	AIE	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 3, 6, 8]
	PIE	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 3, 6, 8]
	RTCOS	V_{DD}	–	0	–	–	Not readable, not writable	Readable and writable[Notes 4, 7]
	PES	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 6, 8]
RCR2	START	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 3, 6, 8]
	RESET	V_{DD}	–	0	–	–	Not readable, not writable	Readable and writable[Notes 4, 6, 8]
	ADJ30	V_{DD}	–	0	–	0	Not readable, not writable	Readable and writable[Notes 4, 6, 8]
	RTCOE	V_{DD}	–	0	–	–	Not readable, not writable	Readable and writable[Notes 4, 7]
	AADJE	VRTC	Undefined	–	–	0	Not readable, not writable	Readable and writable[Note 5]
	AADJP	VRTC	Undefined	–	–	0	Not readable, not writable	Readable and writable[Note 5]
	HR24	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 5, 7]
	CNTMD	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Notes 6, 8]
RCR3	RTCICEN	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Note 5]
RCR4	ADJ500M	VRTC	Undefined	–	–	–	Not readable, not writable	Readable and writable[Note 5]
RADJ	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable[Notes 6, 8]
RTCCRy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable, not writable	Readable and writable[Notes 6, 8]
RSECCPy/BCNT0CPy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable	Readable
RMINCPy/BCNT1CPy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable	Readable

(**Notes** are listed on the next page.)

Table 9-3. Registers that Control the RTC (3/3)

Special Function Register (SFR) Name	Bit Name	Power Domain	RTC Power-on Reset	After MCU Reset		RTC Software Reset	R/W Property	
				Power-on Reset	Reset other than RTCPOR [Note 1]		VRTCEN = 0	VRTCEN = 1
RHRCPy/BCNT2CPy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable	Readable
RDAYCPy/BCNT3CPy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable	Readable
RMONCPy (y = 0 to 2)	–	VRTC	Undefined	–	–	00H	Not readable	Readable
RSR	CF	V_{DD}	–	0[Note 2]	–	0	Not readable, not writable	Readable and writable[Note 4]
SCMC	–	VRTC	00H	–	–	–	Not readable, not writable	Readable and writable[Note 4]
SCSC	–	VRTC	40H	–	–	–	Not readable, not writable	Readable and writable[Note 4]
RTCPORSR	_	VRTC	00H	–	–	–	Not readable, not writable	Readable and writable[Note 4]
RTCICNFEN	–	VRTC	00H	–	–	–	Not readable, not writable	Readable and writable[Note 4]

Notes **1.** Refer to (1) to (8) in **CHAPTER 30 RESET FUNCTION** according to the reset factors.

2. The value read after release from the reset state may be undefined.

3. Wait for 2 cycles of the CPU clock (f_{CLK}) before reading this register after changing the setting of the VRTCEN bit from 0 to 1.

4. Values written can be read immediately after writing to the register.

5. Values written will be read correctly from the 4th cycle of the CPU clock (f_{CLK}) after writing.

6. The value in the register is updated in synchronization with the source for counting. If the value in the register is overwritten, confirm that the value has actually been updated before proceeding with further processing.

7. Do not write to the register during counting (RCR2.START = 1). Stop the counter before writing.

8. After a reset is generated, only write to this RTC register after 6 clock cycles of the source for counting have elapsed. Setting the registers that control the RTC while the power supply from the VRTC pin is stopped is prohibited.

9.2.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

To manipulate of the registers of the realtime clock with independent power supply, be sure to set bit 0 (VRTCEN) to 1.

The PER2 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

VRTCEN	Control of independent power supply RTC input clock supply
0	Stops input clock supply. ● SFR used by the independent power supply RTC cannot be written. The read value is 00H. The independent power supply RTC is able to operate with the sub clock (f_{SX}) as the source for counting.
1	Enables input clock supply. ● SFR used by the independent power supply RTC can be read and written.

Cautions 1. Leak current may be generated via the VRTC pin when the V_{DD} pin power supply voltage is less than 1.8 V. Therefore, set the VRTCEN bit to 0 except during reading or writing of the SFRs of the independent power supply RTC.

2. When the power of the VRTC pin is not supplied, set the VRTCEN bit to 0.

3. Be sure to clear the following bits to 0.
Bits 1 and 3

9.2.2 64-Hz counter (R64CNT)

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock.

The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 00h by an RTC software reset or executing 30-second or 0.5-second adjustment.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-3. Format of 64-Hz Counter (R64CNT)

Address: F0581H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
R64CNT	0	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ

F1HZ	1 Hz
Indicate the state for 1 Hz of the sub-second digit.	

F2HZ	2 Hz
Indicate the state for 2 Hz of the sub-second digit.	

F4HZ	4 Hz
Indicate the state for 4 Hz of the sub-second digit.	

F8HZ	8 Hz
Indicate the state for 8 Hz of the sub-second digit.	

F16HZ	16 Hz
Indicate the state for 16 Hz of the sub-second digit.	

F32HZ	32 Hz
Indicate the state for 32 Hz of the sub-second digit.	

F64HZ	64 Hz
Indicate the state for 64 Hz of the sub-second digit.	

9.2.3 Second counter (RSECCNT)/binary counter 0 (BCNT0)

(1) In calendar count mode:

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.

A BCD value from 00 through 59 can be specified; if a value outside this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-4. Format of Second Counter (RSECCNT)

Address: F0583H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RSECCNT	0	SEC10			SEC1			

SEC10	10-Second Count
Counts from 0 to 5 for 60-second counting.	

SEC1	1-Second Count
Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	

(2) In binary count mode:

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-5. Format of Binary Counter 0 (BCNT0)

Address: F0583H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT0	BCNT[7:0]							

9.2.4 Minute counter (RMINCNT)/binary counter 1 (BCNT1)

(1) In calendar count mode:

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.

A BCD value from 00 through 59 can be specified; if a value outside this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-6. Format of Minute Counter (RMINCNT)

Address: F0585H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RMINCNT	0	MIN10			MIN1			

MIN10	10-Minute Count
Counts from 0 to 5 for 60-minute counting.	

MIN1	1-Minute Count
Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	

(2) In binary count mode:

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-7. Format of Binary Counter 1 (BCNT1)

Address: F0585H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT1	BCNT[15:8]							

9.2.5 Hour counter (RHRCNT)/binary counter 2 (BCNT2)

(1) In calendar count mode:

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.

The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).

When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)

When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)

If a value outside this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-8. Format of Hour Counter (RHRCNT)

Address: F0587H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RHRCNT	0	PM	HR10		HR1			

PM	PM
0	a.m.
1	p.m.
Time Counter Setting for a.m./p.m.	

HR10	10-Hour Count
Counts from 0 to 2 once per carry from the ones place.	

HR1	1-Hour Count
Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	

(2) In binary count mode:

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.

Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-9. Format of Binary Counter 2 (BCNT2)

Address: F0587H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT2	BCNT[23:16]							

9.2.6 Day-of-week counter (RWKCNT)/binary counter 3 (BCNT3)

(1) In calendar count mode:

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.

A BCD value from 0 through 6 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-10. Format of Day-of-Week Counter (RWKCNT)

Address: F0589H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RWKCNT	0	0	0	0	0	DAYW		

DAYW2	DAYW1	DAYW0	Day-of-Week Counting
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday
1	1	1	Setting Prohibited

(2) In binary count mode:

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.

The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.

Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-11. Format of Binary Counter 3 (BCNT3)

Address: F0589H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT3	BCNT[31:24]							

9.2.7 Date counter (RDAYCNT)

The RDAYCNT counter is used in calendar count mode.

RDAYCNT is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.

Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A BCD value from 01 through 31 can be specified; if a value outside this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-12. Format of Date Counter (RDAYCNT)

Address: F058BH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RDAYCNT	0	0	DATE10		DATE1			

DATE10	10-Day Count
Counts from 0 to 3 once per carry from the ones place.	

DATE1	1-Day Count
Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	

9.2.8 Month counter (RMONCNT)

The RMONCNT counter is used in calendar count mode.

RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.

A BCD value from 01 through 12 can be specified; if a value outside this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

Figure 9-13. Format of Month Counter (RMONCNT)

Address: F058DH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RMONCNT	0	0	0	MON10	MON1			

MON10	10-Month Count
Counts from 0 to 1 once per carry from the ones place.	

MON1	1-Month Count
Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	

9.2.9 Year counter (RYRCNT)

The RYRCNT counter is used in calendar count mode.

RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

A BCD value from 00 through 99 can be specified; if a value outside this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.

RYRCNT can only be accessed with a 16-bit memory manipulation instruction.

Figure 9-14. Format of Year Counter (RYRCNT)

Address: F058EH After reset: Undefined R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RYRCNT	0	0	0	0	0	0	0	0	YR10				YR1			

YR10	10-Year Count
Counts from 0 to 9 once per carry from ones place.	

YR1	1-Year Count
Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	

9.2.10 Second alarm register m (RSECARm)/binary counter 0 alarm register m (BCNT0ARm) (m = 0, 1)

(1) In calendar count mode:

RSECARm is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECARm value is compared with the RSECCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

A BCD value from 00 through 59 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

Figure 9-15. Format of Second Alarm Register m (RSECARm)

Address: F0591H (RSECAR0), F05B1H (RSECAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RSECARm	ENB	SEC10			SEC1			

ENB	ENB
0	The register value is not compared with the RSECCNT counter value.
1	The register value is compared with the RSECCNT counter value.

SEC10	10 Seconds
Setting value for the tens place of seconds	

SEC1	1 Seconds
Setting value for the ones place of seconds	

(2) In binary count mode:

The BCNT0ARm counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.

This register is set to 00h by an RTC software reset.

Figure 9-16. Format of Binary Counter 0 Alarm Register m (BCNT0ARm)

Address: F0591H (BCNT0AR0), F05B1H (BCNT0AR1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT0ARm	BCNTAR[7:0]							

9.2.11 Minute alarm register m (RMINARm)/binary counter 1 alarm register m (BCNT1ARm)

(1) In calendar count mode:

RMINARm is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINARm value is compared with the RMINCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

A BCD value from 00 through 59 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

Figure 9-17. Format of Minute Alarm Register m (RMINARm)

Address: F0593H (RMINAR0), F05B3H (RMINAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RMINARm	ENB	MIN10			MIN1			

ENB	ENB
0	The register value is not compared with the RMINCNT counter value.
1	The register value is compared with the RMINCNT counter value.

MIN10	10 Minutes
Setting value for the tens place of minutes	

MIN1	1 Minute
Setting value for the ones place of minutes	

(2) In binary count mode:

The BCNT1ARm counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.

This register is set to 00h by an RTC software reset.

Figure 9-18. Format of Binary Counter 1 Alarm Register m (BCNT1ARm)

Address: F0593H (BCNT1AR0), F05B3H (BCNT1AR1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT1ARm	BCNTAR[15:8]							

9.2.12 Hour alarm register m (RHRARm)/binary counter 2 alarm register m (BCNT2ARm) (m = 0, 1)

(1) In calendar count mode:

RHRARm is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRARm value is compared with the RHRCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
This register is set to 00h by an RTC software reset.

Figure 9-19. Format of Hour Alarm Register m (RHRARm)

Address: F0595H (RHRAR0), F05B5H (RHRAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RHRARm	ENB	PM	HR10		HR1			

ENB	ENB
0	The register value is not compared with the RHRCNT counter value.
1	The register value is compared with the RHRCNT counter value.

PM	PM
0	a.m.
1	p.m.
Time Alarm Setting for a.m. or p.m..	

HR10	10 Hours
Setting value for the tens place of hours	

HR1	1 Hour
Setting value for the ones place of hours	

(2) In binary count mode:

The BCNT2ARm counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is set to 00h by an RTC software reset.

Figure 9-20. Format of Binary Counter 2 Alarm Register m (BCNT2ARm)

Address: F0595H (BCNT2AR0), F05B5H (BCNT2AR1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT2ARm	BCNTAR[23:16]							

9.2.13 Day-of-week alarm register m (RWKARm)/binary counter 3 alarm register m (BCNT3ARm) (m = 0, 1)

(1) In calendar count mode:

RWKARm is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKARm value is compared with the RWKCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

A BCD value from 0 through 6 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

Figure 9-21. Format of Day-of-Week Alarm Register m (RWKARm)

Address: F0597H (RWKAR0), F05B7H (RWKAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RWKARm	ENB	0	0	0	0	DAYW		

ENB	ENB
0	The register value is not compared with the RWKCNT counter value.
1	The register value is compared with the RWKCNT counter value.

DAYW2	DAYW1	DAYW0	Day-of-Week Counting
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday
1	1	1	Setting Prohibited

(2) In binary count mode:

The BCNT3ARm counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is set to 00h by an RTC software reset.

Figure 9-22. Format of Binary Counter 3 Alarm Register m (BCNT3ARm)

Address: F0597H (BCNT3AR0), F05B7H (BCNT3AR1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT3ARm	BCNTAR[31:24]							

9.2.14 Date alarm register m (RDAYARm)/binary counter 0 alarm enable register m (BCNT0AERm) (m = 0, 1)

(1) In calendar count mode:

RDAYARm is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYARm value is compared with the RDAYCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

A BCD value from 01 through 31 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

Figure 9-23. Format of Date Alarm Register m (RDAYARm)

Address: F0599H (RDAYAR0), F05B9H (RDAYAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RDAYARm	ENB	0	DATE10			DATE1		

ENB	ENB
0	The register value is not compared with the RDAYCNT counter value.
1	The register value is compared with the RDAYCNT counter value.

DATE10	10 Days
Setting value for the tens place of days	

DATE1	1 Days
Setting value for the ones place of days	

(2) In binary count mode:

The BCNT0AERm register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

This register is set to 00h by an RTC software reset.

Figure 9-24. Format of Binary Counter 0 Alarm Enable Register m (BCNT0AERm)

Address: F0599H (BCNT0AER0), F05B9H (BCNT0AER1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT0AERm	ENB[7:0]							

9.2.15 Month alarm register m (RMONARm)/binary counter 1 alarm enable register m (BCNT1AERm) (m = 0, 1)

(1) In calendar count mode:

RMONARm is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONARm value is compared with the RMONCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

A BCD value from 01 through 12 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 00h by an RTC software reset.

Figure 9-25. Format of Month Alarm Register m (RMONARm)

Address: F059BH (RMONAR0), F05BBH (RMONAR1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RMONARm	ENB	0	0	MON10	MON1			

ENB	ENB
0	The register value is not compared with the RMONCNT counter value.
1	The register value is compared with the RMONCNT counter value.

MON10	10 Months
Setting value for the ones place of days	

MON1	1 Month
Setting value for the ones place of days	

(2) In binary count mode:

The BCNT1AERm register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

This register is set to 00h by an RTC software reset.

Figure 9-26. Format of Binary Counter 1 Alarm Enable Register m (BCNT1AERm)

Address: F059BH (BCNT1AER0), F05BBH (BCNT1AER1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT1AERm	ENB[15:8]							

9.2.16 Year alarm register m (RYRARm)/binary counter 2 alarm enable register m (BCNT2AERm) (m = 0, 1)

(1) In calendar count mode:

RYRARm is an alarm register corresponding to the BCD-coded year counter RYRCNT.

A BCD value from 00 through 99 can be specified; if a value outside this range is specified, the RTC does not operate correctly.

This register is set to 0000h by an RTC software reset.

RYRARm can only be accessed with a 16-bit memory manipulation instruction.

Figure 9-27. Format of Year Alarm Register m (RYRARm)

Address: F059CH (RYRAR0), F05BCH (RYRAR1) After reset: Undefined R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RYRARm	0	0	0	0	0	0	0	0	YR10				YR1			

YR10	10 Years
Setting value for the tens place of years	

YR1	1 Year
Setting value for the ones place of years	

(2) In binary count mode:

The BCNT2AERm register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

This register is set to 0000h by an RTC software reset.

This register can only be accessed with a 16-bit memory manipulation instruction.

Figure 9-28. Format of Binary Counter 2 Alarm Enable Register m (BCNT2AERm)

Address: F059CH (BCNT2AER0), F05BCH (BCNT2AER1) After reset: Undefined

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT2AERm	0	0	0	0	0	0	0	0	ENB[23:16]							

9.2.17 Year alarm enable register m (RYRARENm)/binary counter 3 alarm enable register m (BCNT3AERm) (m = 0, 1)

(1) In calendar count mode:

When the ENB bit in RYRARENm is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECARm, RMINARm, RHRARm, RWKARm, RDAYARm, RMONARm, and RYRARENm), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

This register is set to 00h by an RTC software reset.

Figure 9-29. Format of Year Alarm Enable Register m (RYRARENm)

Address: F059FH (RYRAREN0), F05BFH (RYRAREN1) After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RYRARENm	ENB	0	0	0	0	0	0	0

ENB	ENB
0	The register value is not compared with the RYRCNT counter value.
1	The register value is compared with the RYRCNT counter value.

(2) In binary count mode:

The BCNT3AERm register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the interrupt request flag RTCAIF0 (for m = 0) or RTCAIF1 (for m = 1) is set to 1.

This register is set to 00h by an RTC software reset.

Figure 9-30. Format of Binary Counter 3 Alarm Enable Register m (BCNT3AERm)

Address: F059FH (BCNT3AER0), F05BFH (BCNT3AER1) After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT3AERm	ENB[31:24]							

9.2.18 RTC control register 1 (RCR1)

The RCR1 register is used in both calendar count mode and in binary count mode.

Bits AIE, PIE, and PES are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

Figure 9-31. Format of RTC Control Register 1 (RCR1)

Address: F05A3H After reset: Undefined[Note] R/W

Symbol	7	6	5	4	3	2	1	0
RCR1	PES				RTCOS	PIE	0	AIE

PES	Periodic Interrupt Select
0110	A periodic interrupt is generated every 1/256 second.
0111	A periodic interrupt is generated every 1/128 second.
1000	A periodic interrupt is generated every 1/64 second.
1001	A periodic interrupt is generated every 1/32 second.
1010	A periodic interrupt is generated every 1/16 second.
1011	A periodic interrupt is generated every 1/8 second.
1100	A periodic interrupt is generated every 1/4 second.
1101	A periodic interrupt is generated every 1/2 second.
1110	A periodic interrupt is generated every 1 second.
1111	A periodic interrupt is generated every 2 seconds.
Other than above	No periodic interrupts are generated.
These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.	

RTCOS	RTCOUT Output Select
0	RTCOUT outputs 1 Hz.
1	RTCOUT outputs 64 Hz.
This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. About I/O ports, refer to **CHAPTER 4 PORT FUNCTIONS**.	

PIE	Periodic Interrupt Control
0	A periodic interrupt request is disabled.
1	A periodic interrupt request is enabled.
This bit enables or disabled a periodic interrupt.	

AIE	Alarm Interrupt Control
0	An alarm interrupt request is disabled.
1	An alarm interrupt request is enabled.
This bit enables or disables alarm interrupt requests.	

Note The setting of the RTCOS bit following the generation of a power-on reset signal is 0.

Caution Be sure to set bit 1 to 0.
This register can only be accessed with an 8-bit memory manipulation instruction.

9.2.19 RTC control register 2 (RCR2)

(1) In calendar count mode:

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

Figure 9-32. Format of RTC Control Register 2 (RCR2) (In Calendar Count Mode) (1/3)

Address: F05A5H After reset: Undefined[Note] R/W

Symbol	7	6	5	4	3	2	1	0
RCR2	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START

CNTMD	Count Mode Select
0	The calendar count mode.
1	The binary count mode.
This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode. When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed. For details on initial settings, refer to **9.3.1 Outline of initial settings of registers after power on**.	

HR24	Hours Mode
0	The RTC operates in 12-hour mode.
1	The RTC operates in 24-hour mode.
This bit specifies whether the RTC will operate in 12- or 24-hour mode. Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.	

AADJP	Automatic Adjustment Period Select
0	The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute.
1	The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.
This bit selects the automatic-adjustment period. Set the plus–minus bits (RADJ.PMADJ) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.	

AADJE	Automatic Adjustment Control
0	Automatic adjustment is disabled.
1	Automatic adjustment is enabled.
This bit controls (enables or disables) automatic adjustment. Set the plus–minus bits (RADJ.PMADJ) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.	

Note The setting of the RESET, ADJ30, and RTCOE bits following the generation of a power-on reset signal is 0.

Figure 9-32. Format of RTC Control Register 2 (RCR2) (In Calendar Count Mode) (2/3)

Address: F05A5H After reset: Undefined[Note 1] R/W

Symbol	7	6	5	4	3	2	1	0
RCR2	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START

RTCOE	RTCOUT Output Control
0	RTCOUT output disabled.
1	RTCOUT output enabled.
This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin. Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.	

ADJ30	30-Second Adjustment Control
In writing	
0	Writing is invalid.
1	30-second adjustment is executed.
In reading	
0	The time counters are operating normally or 30-second adjustment has been completed.
1	30-second adjustment is in progress.
This bit is for 30-second adjustment. When 1 is written to the ADJ30 bit, an RSECCNT value equivalent to less than 30 seconds is rounded down to 00 seconds and a value equivalent to 30 seconds or more is rounded up to 1 minute. The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set to 0, and then make next settings. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the 30-second adjustment is completed. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is set to 0 by an RTC software reset.	

RESET	RTC Software Reset Control
In writing	
0	Writing is invalid.
1	The prescaler and the target registers for RTC software reset[Note 2] are initialized
In reading	
0	The time counters are operating normally or an RTC software reset has been completed.
1	An RTC software reset is in progress.
This bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the initialization is completed.	

Notes 1. The setting of the RESET, ADJ30, and RTCOE bits following the generation of a power-on reset signal is 0.

2. R64CNT, RSECARm/BCNT0ARm, RMINARm/BCNT1ARm, RHRARm/BCNT2ARm, RWKARm/BCNT3ARm, RDAYARm/BCNT0AERm, RMONARm/BCNT1AERm, RYRARm/BCNT2AERm, RYRARENm/BCNT3AERm, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP, RSR

Figure 9-32. Format of RTC Control Register 2 (RCR2) (In Calendar Count Mode) (3/3)

Address: F05A5H After reset: Undefined[Note] R/W

Symbol	7	6	5	4	3	2	1	0
RCR2	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START

START	Prescaler and time counter operation control
0	Prescaler and time counter are stopped.
1	Prescaler and time counter operate normally.

This bit stops or restarts the prescaler or time counter operation.
The START bit is updated in synchronization with the next count source. When the START bit is modified, check that the bit has been updated before proceeding to the next processing. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the bit is updated.

Note The setting of the RESET, ADJ30, and RTCOE bits following the generation of a power-on reset signal is 0.

Cautions 1. This register can only be accessed with an 8-bit memory manipulation instruction.

2. 30-second adjustment and 0.5-second adjustment cannot be used simultaneously. Only set either ADJ30 or ADJ500M to 1.

(2) In binary count mode:

Figure 9-33. Format of RTC Control Register 2 (RCR2) (In Binary Count Mode) (1/2)

Address: F05A5H After reset: Undefined[Note] R/W

Symbol	7	6	5	4	3	2	1	0
RCR2	CNTMD	0	AADJP	AADJE	RTCOE	0	RESET	START

CNTMD	Count Mode Select
0	The calendar count mode.
1	The binary count mode.
This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode. When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed. For details on initial settings, refer to **9.3.1 Outline of initial settings of registers after power on**.	

AADJP	Automatic Adjustment Period Select
0	Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds
1	Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 8 seconds
This bit selects the automatic-adjustment period. Correction period can be selected from 32 second units or 8 second units in binary count mode. Set the plus–minus bits (RADJ.PMADJ) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.	

AADJE	Automatic Adjustment Control
0	Automatic adjustment is disabled.
1	Automatic adjustment is enabled.
This bit controls (enables or disables) automatic adjustment. Set the plus–minus bits (RADJ.PMADJ) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.	

RTCOE	RTCOUT Output Control
0	RTCOUT output disabled.
1	RTCOUT output enabled.
This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin. Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control as well as setting this bit.	

Note The setting of the RESET and RTCOE bits following the generation of a power-on reset signal is 0.

Figure 9-33. Format of RTC Control Register 2 (RCR2) (In Binary Count Mode) (2/2)

Address: F05A5H After reset: Undefined[Note 1] R/W

Symbol	7	6	5	4	3	2	1	0
RCR2	CNTMD	0	AADJP	AADJE	RTCOE	0	RESET	START

RESET	RTC Software Reset Control
In writing	
0	Writing is invalid.
1	The prescaler and the target registers for RTC software reset[Note 2] are initialized
In reading	
0	The time counters are operating normally or an RTC software reset has been completed.
1	An RTC software reset is in progress.
This bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the initialization is completed.	

START	32-bit binary counter, 64-Hz counter, and prescaler operation control
0	The 32-bit binary counter, 64-Hz counter, and prescaler are stopped.
1	The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation.
This bit stops or restarts the prescaler or time counter operation. The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated, and then make next settings. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the bit is updated.	

Notes 1. The setting of the RESET and RTCOE bits following the generation of a power-on reset signal is 0.

2. R64CNT, RSECARm/BCNT0ARm, RMINARm/BCNT1ARm, RHRARm/BCNT2ARm, RWKARm/BCNT3ARm, RDAYARm/BCNT0AERm, RMONARm/BCNT1AERm, RYRARm/BCNT2AERm, RYRARENm/BCNT3AERm, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP, RSR

Caution This register can only be accessed with an 8-bit memory manipulation instruction.

9.2.20 RTC control register 3 (RCR3)

The RCR3 register is used to select the time capture event input (RTCICn) pins enable/disable.

Figure 9-34. Format of RTC Control Register 3 (RCR3)

Address: F05A7H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RCR3	0	0	0	0	0	0	0	RTCICEN

RCR3 register	RTCCRn register	Time capture event input (RTCICn) enable/disable
RTCICEN	TCEN	
0	0	RTCICn input is disabled.
0	1	Setting prohibitedNote
1	0	RTCICn input is disabled.
1	1	RTCICn input is enabled.

RTCICEN bit (Time capture event input (RTCICn) control bit)
This bit is used to select the time capture event input (RTCICn) pins enable/disable.
When using the time capture event input function, be sure to set the TCEN bit to 1 after setting the RTCICEN bit to 1.
When not using the time capture event input function, set the RTCICEN bit to 0. However, the external interrupt function of the RTCICn pin can be used, even when setting the RTCICEN bit to 0.

Note Setting the TCEN bit to 1 is prohibited when setting the RTCICEN to 0. When using the RTCICn pin, be sure to set the TCEN bit to 1 after setting the RTCICEN bit to 1.

Caution Be sure to set bits 7 to 1 to “0”. This register can only be accessed with an 8-bit memory manipulation instruction.

Remark n = 0 to 2

9.2.21 RTC control register 4 (RCR4)

The RCR4 register controls 0.5-second adjustment in the calendar count mode.

Figure 9-35. Format of RTC Control Register 4 (RCR4)

Address: F05A9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RCR4	0	0	0	0	0	0	0	ADJ500M

ADJ500M	0.5-Second Adjustment Control
In writing	
0	Writing is invalid.
1	0.5-second adjustment is executed.
In reading	
0	The time counters are operating normally or 0.5-second adjustment has been completed.
1	0.5-second adjustment is in progress.
This bit is for 0.5-second adjustment. When 1 is written to the ADJ500M bit, an R64CNT value less than 40h (0.5 of a second) is rounded down to 00h (00 seconds) and a value of or greater than 40h (0.5 of a second) is rounded up to 1 second. The 0.5-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ500M bit is automatically set to 0 after the 0.5-second adjustment is completed. In case when 1 is written to the ADJ500M bit, check that the bit is set to 0, and then make next settings. Additionally, be sure to maintain the value of the VRTCEN bit as 1 until the 0.5-second adjustment is completed. When the 0.5-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ500M bit is set to 0 by an RTC software reset.	

Cautions 1. This register can only be accessed with an 8-bit memory manipulation instruction.

2. 30-second adjustment and 0.5-second adjustment cannot be used simultaneously. Only set either ADJ30 or ADJ500M to 1.

9.2.22 Time error adjustment register (RADJ)

The RADJ register is used both in calendar count mode and in binary count mode.

Adjustment is performed by the addition to or subtraction from the prescaler.

In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.

In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated before continuing with further processing.

This register is set to 00h by an RTC software reset.

Figure 9-36. Format of Time Error Adjustment Register (RADJ)

Address: F05AFH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RADJ	PMADJ		ADJ					

PMADJ	Time Error Adjustment Operation Control
00	Adjustment is not performed.
01	Adjustment is performed by the addition to the prescaler.
10	Adjustment is performed by the subtraction from the prescaler.
11	Setting prohibited
These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ bits.	

ADJ	Time Error Adjustment Value Setting
These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.	

9.2.23 Time capture control register y (RTCCRy) (y = 0 to 2)

The RTCCRy register is used both in calendar count mode and in binary count mode.

RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRy is updated in synchronization with the count source. When RTCCRy is modified, check that all the bits except for the TCST bit have been updated before continuing with further processing.

This register is set to 00h by an RTC software reset.

Caution If the VRTC power is being supplied while the supply of power from the V_{DD} pin is stopped, the time capture function of the RTC is available.

Figure 9-37. Format of Time Capture Control Register y (RTCCRy) (y = 0 to 2) (1/2)

Address: RTCCR0 F05C1H, RTCCR1 F05C3H, RTCCR2 F05C5H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RTCCRy	TCEN	0	TCNF		0	TCST	TCCT	

TCEN	Time Capture Event Input Pin Control
0	The RTCICn pin is disabled as the time capture event input.
1	The RTCICn pin is enabled as the time capture event input.
This bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). When the functions of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are multiplexed, set the port control and enable this bit. In this case, the port control should be set first. If the TCEN bit is set to 0, set also the TCCT bits to 00b.	

TCNF	Time Capture Noise Filter Control
00	The noise filter is off.
01	Setting prohibited
10	The noise filter is on (count source).
11	The noise filter is on (count source by divided by 32).
These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). When the noise filter is on, the count source divided by 1 or 32 is selectable by the combination with the RTCICNFEN register. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined. Set the TCNF bits while the TCCT bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT bits. Set the TCNF[1:0] bits when the TCEN bit is 1.	

Figure 9-37. Format of Time Capture Control Register y (RTCCRy) (y = 0 to 2) (2/2)

Address: RTCCR0 F05C1H, RTCCR1 F05C3H, RTCCR2 F05C5H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
RTCCRy	TCEN	0	TCNF		0	TCST	TCCT	

TCST	Time Capture Status
0	No event is detected.
1	An event is detected.[Note]

This bit indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected.
When the TCST bit is 0, no event is detected.
When the TCST bit is 1, this bit indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.
If an event is detected while the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, set the TCST bit to 0 for deleting the captured value.
Writing 0 sets the TCST bit to 0. In addition, writing any other value except 0 has no effect.
Set the TCST bit while the TCCT bits are 00b (no event is detected).
The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit has been updated before continuing with further processing.

TCCT	Time Capture Control
00	No event is detected.
01	Rising edge is detected.
10	Falling edge is detected.
11	Both edges are detected.

These bits control the edge detection of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). The detection edge is selectable. The TCCT bits should be set while the TCEN bit is 1.

Note Indicates that an event has been detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

9.2.24 Second capture register y (RSECCPy) (y = 0 to 2)/BCNT0 capture register y (BCNT0CPy) (y = 0 to 2)

(1) In calendar count mode:

RSECCPy is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-38. Format of Second Capture Register y (RSECCPy) (y = 0 to 2)

Address: RSECCP0 F05D3H, RSECCP1 F05E3H, RSECCP2 F05F3H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RSECCPy	0	SEC10			SEC1			

SEC10	10-Second Capture
Capture value for the tens place of seconds	

SEC1	1-Second Capture
Capture value for the ones place of seconds	

(2) In binary count mode:

BCNT0CPy is a read-only register that captures the BCNT0 value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-39. Format of BCNT0 Capture Register y (BCNT0CPy) (y = 0 to 2)

Address: BCNT0CP0 F05D3H,BCNT0CP1 F05E3H, BCNT0CP2 F05F3H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT0CPy	BCNTCPy							

9.2.25 Minute capture register y (RMINCPy) (y = 0 to 2)/BCNT1 capture register y (BCNT1CPy) (y = 0 to 2)

(1) In calendar count mode:

RMINCPy is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-40. Format of Minute Capture Register y (RMINCPy) (y = 0 to 2)

Address: RMINCP0 F05D5H, RMINCP1 F05E5H, RMINCP2 F05F5H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RMINCPy	0	MIN10			MIN1			

MIN10	10-Minute Capture
Capture value for the tens place of minutes	

MIN1	1-Minute Capture
Capture value for the ones place of minutes	

(2) In binary count mode:

BCNT1CPy is a read-only register that captures the BCNT1 value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-41. Format of BCNT1 Capture Register y (BCNT1CPy) (y = 0 to 2)

Address: BCNT1CP0 F05D5H, BCNT1CP1 F05E5H, BCNT1CP2 F05F5H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT1CPy	BCNTCPy							

9.2.26 Hour capture register y (RHRCPy) (y = 0 to 2)/BCNT2 capture register y (BCNT2CPy) (y = 0 to 2)

(1) In calendar count mode:
RHRCPy is a read-only register that captures the RHRCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-42. Format of Hour Capture Register y (RHRCPy) (y = 0 to 2)

Address: RHRCP0 F05D7H, RHRCP1 F05E7H, RHRCP2 F05F7H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RHRCPy	0	PM	HR10		HR1			

PM	PM
0	a.m.
1	p.m.

HR10	10-Hour Capture
Capture value for the tens place of hours	

HR1	1-Hour Capture
Capture value for the ones place of hours	

(2) In binary count mode:
BCNT2CPy is a read-only register that captures the BCNT2 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-43. Format of BCNT2 Capture Register y (BCNT2CPy) (y = 0 to 2)

Address: BCNT2CP0 F05D7H, BCNT2CP1 F05E7H, BCNT2CP2 F05F7H After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT2CPy	BCNTCPy							

9.2.27 Date capture register y (RDAYCPy) (y = 0 to 2)/BCNT3 capture register y (BCNT3CPy) (y = 0 to 2)

(1) In calendar count mode:

RDAYCPy is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-44. Format of Date Capture Register y (RDAYCPy) (y = 0 to 2)

Address: RDAYCP0 F05DBH, RDAYCP1 F05EBH, RDAYCP2 F05FBH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RDAYCPy	0	0	DATE10		DATE1			

DATE10	10-Day Capture
Capture value for the tens place of days	

DATE1	1-Day Capture
Capture value for the ones place of days	

(2) In binary count mode:

BCNT3CPy is a read-only register that captures the BCNT3 value when a time capture event is detected.

The event detection times detected by the RTCTC0, RTCTC1, and RTCTC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-45. Format of BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2)

Address: BCNT3CP0 F05DBH, BCNT3CP1 F05EBH, BCNT3CP2 F05FBH After reset: Undefined

Symbol	7	6	5	4	3	2	1	0
BCNT3CPy	BCNTCPy							

9.2.28 Month capture register y (RMONCPy) (y = 0 to 2)

(1) In calendar count mode:

RMONCPy is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

Figure 9-46. Format of Month Capture Register y (RMONCPy) (y = 0 to 2)

Address: RMONCP0 F05DDH, RMONCP1 F05EDH, RMONCP2 F05FDH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RMONCPy	0	0	0	MON10	MON1			

MON10	10-Month Capture
Capture value for the tens place of months	

MON1	1-Month Capture
Capture value for the ones place of months	

9.2.29 RTC status register (RSR)

RSR is a carry flag register. This register is a common function with the calendar count mode and the binary count mode.

The CF flag is set to 1 when the prescaler or the time counter matches each interrupt setting condition. The prescaler, time counter, and the setting register of each interrupt are not reset, so the flag may be set before it is read.

This register is set to 00h by an RTC software reset.

Figure 9-47. Format of RTC Status Register (RSR)

Address:F05A1H After reset: 00H[Note] R/W

Symbol	7	6	5	4	3	2	1	0
RSR	0	0	0	0	0	0	CF	0

CF	Carry flag
0	No carry of second counter/binary counter 0, and no carry of the 64 Hz counter when the 64 Hz counter is reading
1	Carry of second counter/binary counter 0, or carry of the 64 Hz counter when the 64 Hz counter is reading
During CF = 1, be sure to read again because the value which is read from the count register is not guaranteed. <Clear conditions> ● 0 is written to the CF flag. <Set condition> ● Carry of second counter/binary counter 0, or carry of the 64 Hz counter when the 64 Hz counter is reading ● 1 is written to the CF flag.	

Note After reset is released, read value may be undefined.

Cautions 1. Use the RTCAIF0 flag of IF1L (for m = 0) or RTCAIF1 flag of IF1H (for m = 1) to confirm the generation of RTC alarm interrupt m.

2. Use the RTCRIF flag of the interrupt request flag register (IF1H) to generate the periodic interrupt.

Remark m = 0, 1

9.2.30 Sub clock operation mode control register (SCMC)

This register is used to set the operating mode of the XT1/P123 and XT2/EXCLKS/P124 pins, and to select the gain of the oscillator.

After release from an RTC power-on reset or a reset from any other source, the SCMC register can be written only once by an 8-bit memory manipulation instruction. This register can be read by an 8-bit memory manipulation instruction.

The SCMC register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal clears this register to 00H. This register is not reset by other reset sources (including the power-on reset of the V_{DD} power supply).

Figure 9-48. Format of Sub Clock Operation Mode Control Register (SCMC)

Address: F0384H After reset: 00H[Note] R/W

Symbol	7	6	5	4	3	2	1	0
SCMC	0	0	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	0

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

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low-power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

Note The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are only initialized by an RTC power-on reset; they retain their values following a reset due to another source (including the power-on reset of the V_{DD} power supply).

Cautions 1. After the CPU is released from the reset state, the SCMC register can be written only once by an 8-bit memory manipulation instruction. When using the SCMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop.

2. After the CPU is released from the reset state, set the SCMC register before XT1 oscillation is started as set by the sub clock operation status control register (SCSC).

3. Specify the settings for the AMPHS1 and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

4. Count the f_{XT} oscillation stabilization time by using software.

5. After the CPU is released from the reset state following writing to the SCMC register and then a reset other than an RTC power-on reset, set the same value as the value before the reset to prevent incorrect operation in the case of an endless loop or runaway execution.

Cautions 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 the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 6.7 Resonator and Oscillator Constants. Using ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is not recommended for applications (e.g. utility meters) which require securing wide margins for oscillation. In such cases, we recommend the use of normal oscillation (AMPHS1, AMPHS0 = 0, 1).**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) 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 potential 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 resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high 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.**

7. Be sure to clear bits 7, 6, 3, and 0 to 0.

9.2.31 Sub clock operation status control register (SCSC)

This register is used to control the operation of the sub clock.

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

The SCSC register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal sets this register to 40H.

Figure 9-49. Format of Sub Clock Operation Status Control Register (SCSC)

Address: F0386H After reset: 40H R/W

Symbol	7	<6>	5	4	3	2	1	0
SCSC	0	XTSTOP	0	0	0	0	0	0

XTSTOP[Note]	Control of XT1 oscillator operation
0	XT1 oscillation mode: XT1 oscillator operating External clock input mode: External clock from EXCLKS pin is valid. Input port mode: Input port
1	XT1 oscillation mode: XT1 oscillator stopped External clock input mode: External clock from EXCLKS pin is invalid. Input port mode: Input port

Note The XTSTOP bit is only initialized by an RTC power-on reset; it retains its value following a reset due to another source (including the power-on reset of the V_{DD} power supply).

Cautions 1. When starting XT1 oscillation by setting the XTSTOP bit, use software to wait for oscillation of the sub clock to become stable.

2. Be sure to clear the bits 7 and 5 to 0 to 0.

9.2.32 RTC power-on-reset status register (RTCPORSR)

The RTCPORSR register is used to check the occurrence of an RTC Power-on reset.

Writing 1 to bit 0 (RTCPORF) of the RTCPORSR register enables this function. Writing 0 disables this function.

Write 1 to the RTCPORF bit in advance to enable checking of the occurrence of an RTC power-on reset.

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

The RTCPORSR register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal clears this register to 00H.

Cautions 1. The RTCPORSR register is reset only by an RTC power-on reset; it retains the value when a reset caused by another source occurs.

2. The RTCPORSR register is readable and writable while the VRTCEN bit is "1".

Figure 9-50. Format of RTC Power-on-reset Status Register (RTCPORSR)

Address: F0380H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCPORSR	0	0	0	0	0	0	0	RTCPORF

RTCPORF	Checking occurrence of RTC Power-on reset
0	RTC power-on reset has occurred.
1	No RTC power-on reset has occurred.

9.2.33 Noise filter enable register for RTCICn pin (n = 0 to 2) (RTCICNFEN)

The RTCICNFEN register enables or disables the noise filters for the RTCICn pins (n = 0 to 2).

Each noise filter has two enable bits, and in combination they are used to select the frequency obtained by dividing the RTC counting source (f_{SX}) by 2^{12} or 2^{13}.

At this time, when the level on the RTCICn pin sampled at the selected interval matches three times, the level is judged to be the actual input level.

Caution The RTCICNFEN register is reset only by an RTC power-on reset; it retains the value when a reset caused by another source occurs.

Figure 9-51. Format of Noise Filter Enable Register for RTCICn Pin (n = 0 to 2) (RTCICNFEN)

Address: F0382H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCICNFEN	0	RTCIC2NF1	RTCIC1NF1	RTCIC0NF1	0	RTCIC2NF0	RTCIC1NF0	RTCIC0NF0

RTCIC2NF1	RTCIC2NF0	Enabling or disabling of the noise filter for the RTCIC2 pin
0	0	The noise filter is disabled.
0	1	
1	0	The noise filter is enabled and driven by the RTC counting source divided by 2^{12} = 250 ms.
1	1	The noise filter is enabled and driven by the RTC counting source divided by 2^{13} = 500 ms.

RTCIC1NF1	RTCIC1NF0	Enabling or disabling of the noise filter for the RTCIC1 pin
0	0	The noise filter is disabled.
0	1	
1	0	The noise filter is enabled and driven by the RTC counting source divided by 2^{12} = 250 ms.
1	1	The noise filter is enabled and driven by the RTC counting source divided by 2^{13} = 500 ms.

RTCIC0NF1	RTCIC0NF0	Enabling or disabling of the noise filter for the RTCIC0 pin
0	0	The noise filter is disabled.
0	1	
1	0	The noise filter is enabled and driven by the RTC counting source divided by 2^{12} = 250 ms.
1	1	The noise filter is enabled and driven by the RTC counting source divided by 2^{13} = 500 ms.

Cautions 1. Be sure to clear bits 7 and 3 to 0.

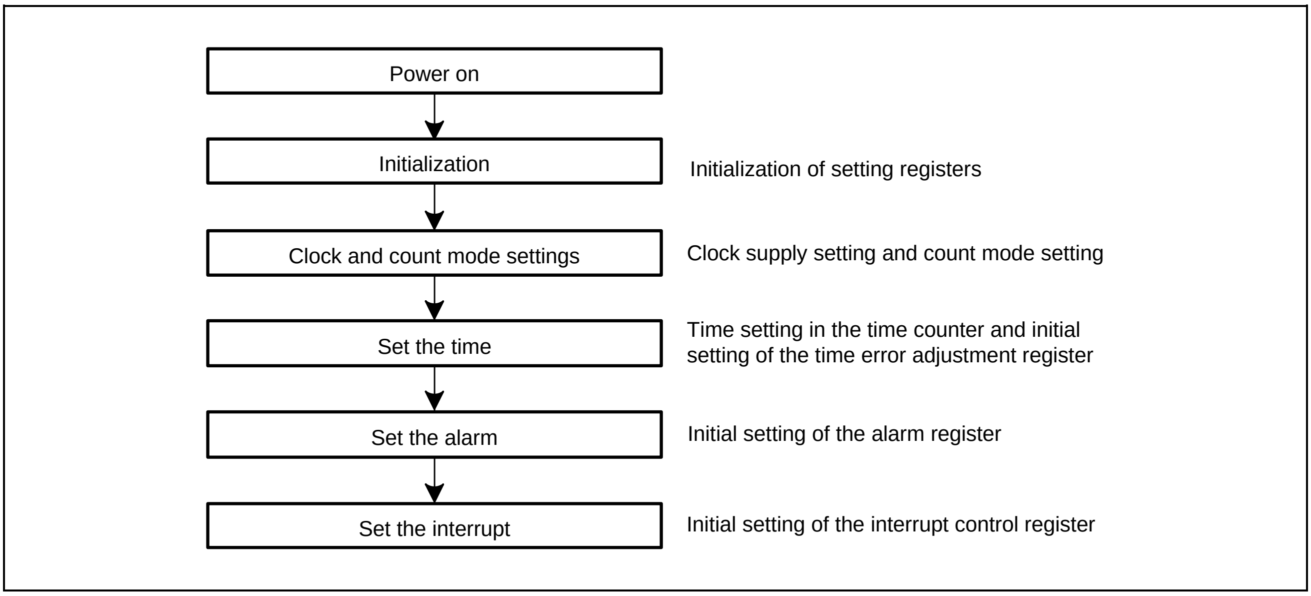
2. Set the RTCICNFEN register while the RTCCRy.TCCT bits are 00b (no event is detected). When the noise filter is used, set the RTCICNFEN register, wait for three cycles of the specified sampling period, and then set the RTCCRy.TCCT bits. Set the RTCICNFEN register when the TCEN bit is 1.

9.3 Operation

9.3.1 Outline of initial settings of registers after power on

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

Figure 9-52. Outline of Initial Settings after Power On

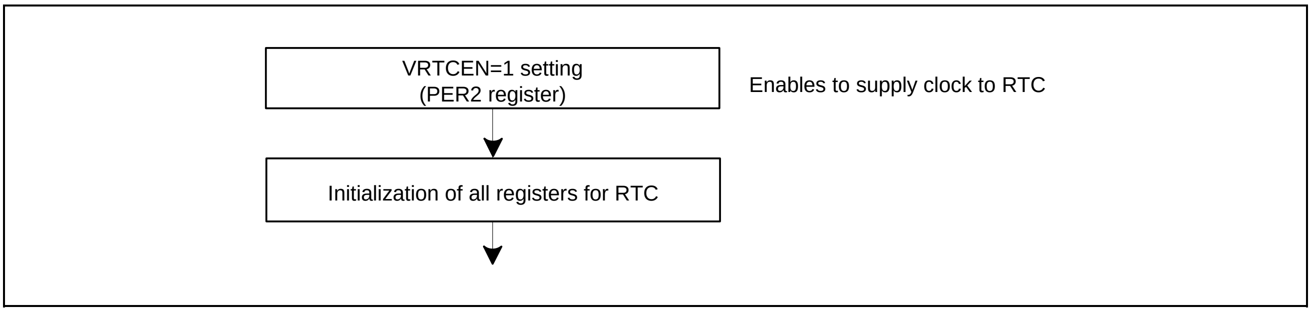

Power on
Initialization
Initialization of setting registers
Clock and count mode settings
Clock supply setting and count mode setting
Set the time
Time setting in the time counter and initial setting of the time error adjustment register
Set the alarm
Initial setting of the alarm register
Set the interrupt
Initial setting of the interrupt control register

Remark The minimum operating voltage of V_{DD} is 1.6 V or 1.8 V, although the minimum operating voltage of VRTC is 1.6 V.

9.3.2 Initialization procedure

Figure 9-53 shows an initialization procedure.

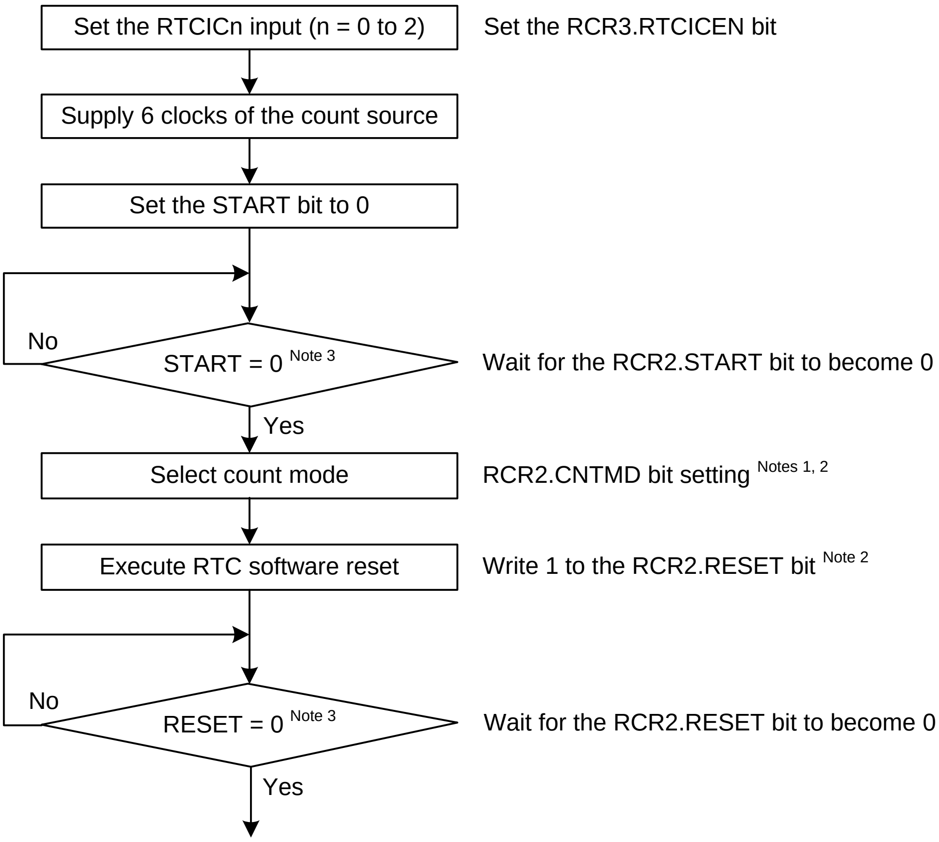
Figure 9-53. Initialization Procedure


VRTCEN=1 setting
(PER2 register)
Enables to supply clock to RTC
Initialization of all registers for RTC

9.3.3 Clock and count mode setting procedure

Figure 9-54 shows how to set the clock and the count mode.

Figure 9-54. Clock and Count Mode Setting Procedure


Set the RTCICn input (n = 0 to 2)
Set the RCR3.RTCICEN bit
Supply 6 clocks of the count source
Set the START bit to 0
No
START = 0 Note 3
Wait for the RCR2.START bit to become 0
Yes
Select count mode
RCR2.CNTMD bit setting Notes 1, 2
Execute RTC software reset
Write 1 to the RCR2.RESET bit Note 2
No
RESET = 0 Note 3
Wait for the RCR2.RESET bit to become 0
Yes

Notes 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0.

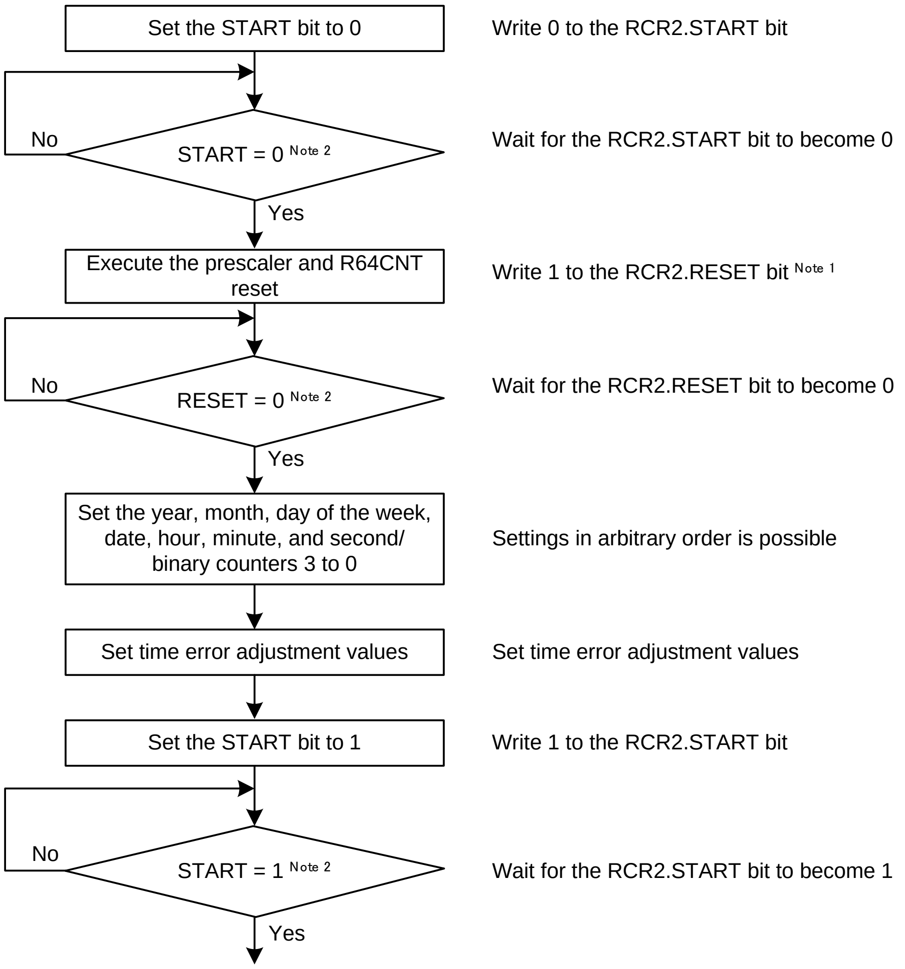
2. Rewriting the RCR2 register with a way of read-modify-writing is prohibited.

3. Be sure to maintain the value of the VRTCEN bit as 1.

9.3.4 Setting the time procedure

Figure 9-55 shows how to set the time.

Figure 9-55. Setting the Time


Set the START bit to 0
Write 0 to the RCR2.START bit
No
START = 0 Note 2
Wait for the RCR2.START bit to become 0
Yes
Execute the prescaler and R64CNT reset
Write 1 to the RCR2.RESET bit Note 1
No
RESET = 0 Note 2
Wait for the RCR2.RESET bit to become 0
Yes
Set the year, month, day of the week, date, hour, minute, and second/ binary counters 3 to 0
Settings in arbitrary order is possible
Set time error adjustment values
Set time error adjustment values
Set the START bit to 1
Write 1 to the RCR2.START bit
No
START = 1 Note 2
Wait for the RCR2.START bit to become 1
Yes

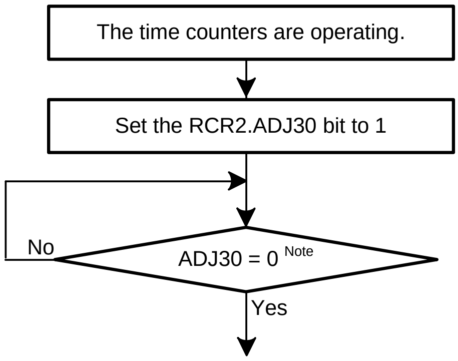
Notes 1. This step is not necessary for the time-setting procedure because an RTC software reset is executed in the clock setting procedure of the initial settings for the power supply.

2. Be sure to maintain the value of the VRTCEN bit as 1.

9.3.5 30-second adjustment procedure

Figure 9-56 shows how to execute 30-second adjustment. The 30-second adjustment function is only available in calendar count mode.

Figure 9-56. 30-Second Adjustment Procedure



Execute 30-second adjustment while the time counters are operating (the RCR2.START bit is 1)

Write 1 to the RCR2.ADJ30 bit

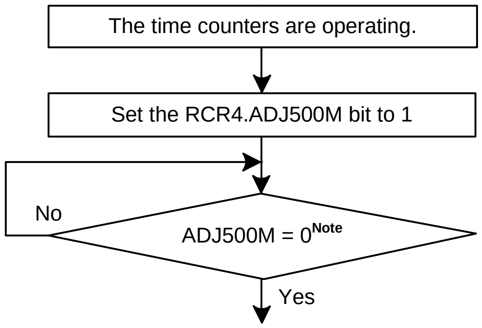
Wait for the RCR2.ADJ30 bit to become 0

Note Be sure to maintain the value of the VRTCEN bit as 1 until the 30-second adjustment is completed.

9.3.6 0.5-second adjustment procedure

Figure 9-57 shows how to execute 0.5-second adjustment. The 0.5-second adjustment function is only available in calendar count mode.

Figure 9-57. 0.5-Second Adjustment Procedure


The time counters are operating.
Set the RCR4.ADJ500M bit to 1
No
ADJ500M = 0Note
Yes

Execute 0.5-second adjustment while the time counters are operating (the RCR2.START bit is 1)

Write 1 to the RCR4.ADJ500M bit

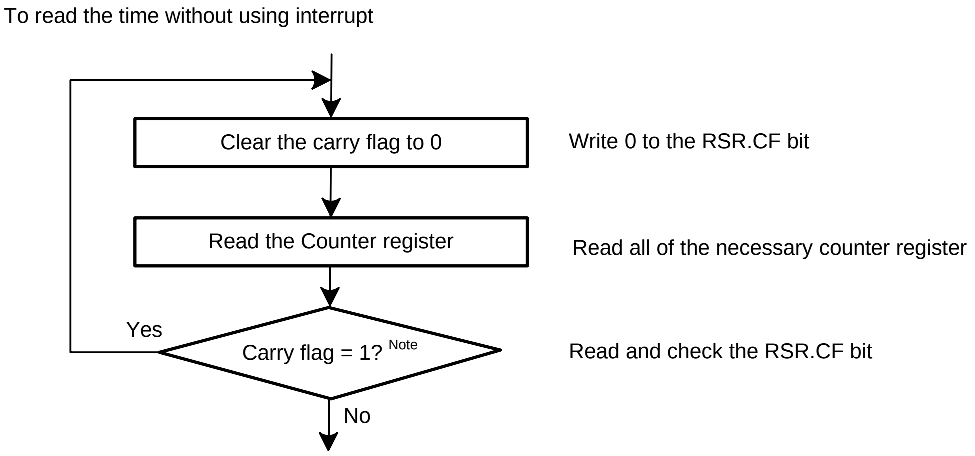
Wait for the RCR4.ADJ500M bit to become 0

Note Be sure to maintain the value of the VRTCEN bit as 1 until the 0.5-second adjustment is completed.

9.3.7 Reading 64-Hz counter and time

Figure 9-58 shows how to read the 64-Hz counter and time.

Figure 9-58. Reading Time


To read the time without using interrupt
Clear the carry flag to 0
Write 0 to the RSR.CF bit
Read the Counter register
Read all of the necessary counter register
Yes
Carry flag = 1? Note
Read and check the RSR.CF bit
No

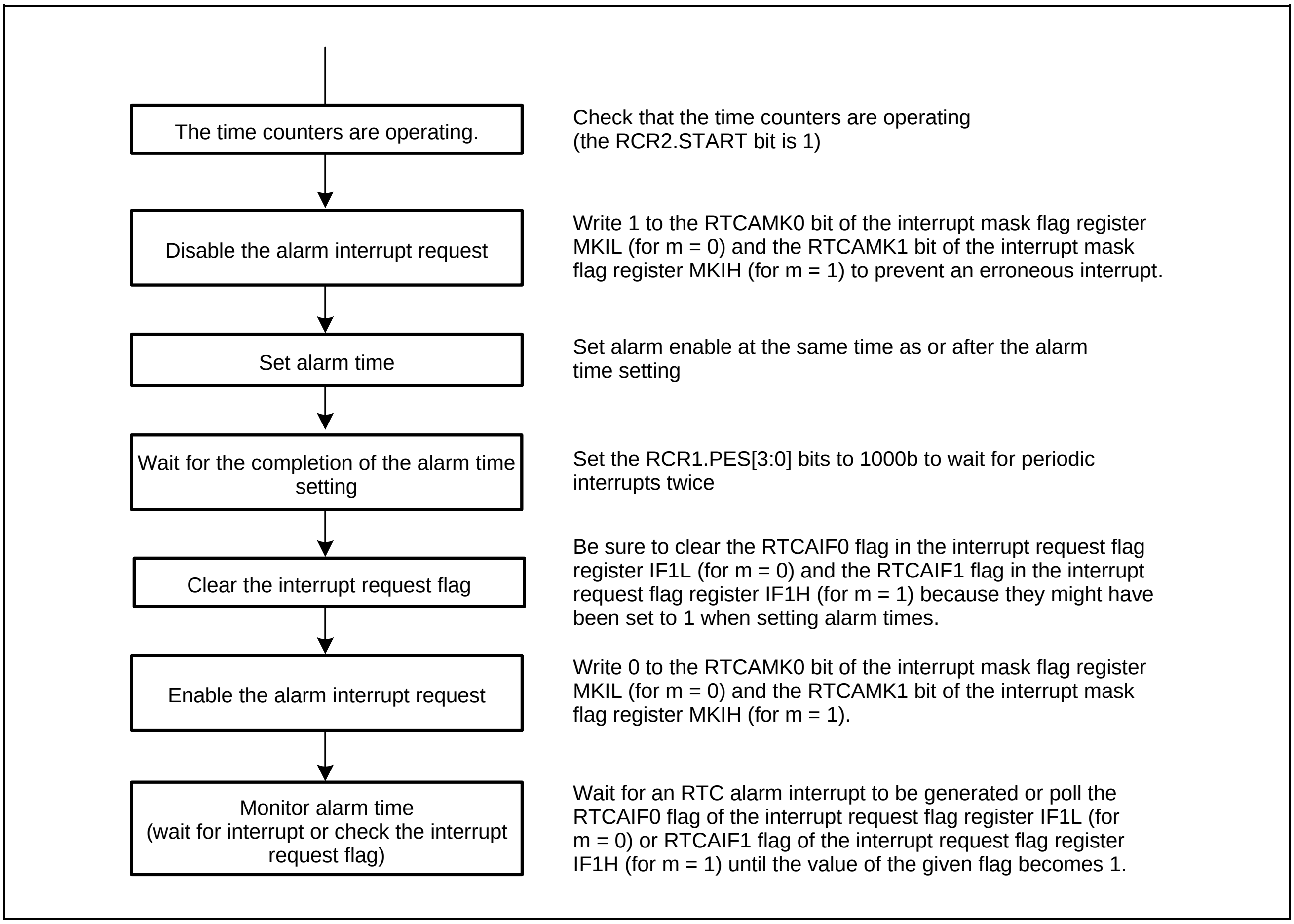
Note The carry flag (CF) will not be set to 1 while the VRTCEN bit is 0. The VRTCEN bit must be 1 for the carry flag (CF) to be read.

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in **Figure 9-58**.

9.3.8 Alarm function

Figure 9-59 shows how to use the alarm function m.

Figure 9-59. Using Alarm Function m


The time counters are operating.
Disable the alarm interrupt request
Set alarm time
Wait for the completion of the alarm time setting
Clear the interrupt request flag
Enable the alarm interrupt request
Monitor alarm time (wait for interrupt or check the interrupt request flag)

Check that the time counters are operating (the RCR2.START bit is 1)

Write 1 to the RTCAMK0 bit of the interrupt mask flag register MKIL (for m = 0) and the RTCAMK1 bit of the interrupt mask flag register MKIH (for m = 1) to prevent an erroneous interrupt.

Set alarm enable at the same time as or after the alarm time setting

Set the RCR1.PES[3:0] bits to 1000b to wait for periodic interrupts twice

Be sure to clear the RTCAIF0 flag in the interrupt request flag register IF1L (for m = 0) and the RTCAIF1 flag in the interrupt request flag register IF1H (for m = 1) because they might have been set to 1 when setting alarm times.

Write 0 to the RTCAMK0 bit of the interrupt mask flag register MKIL (for m = 0) and the RTCAMK1 bit of the interrupt mask flag register MKIH (for m = 1).

Wait for an RTC alarm interrupt to be generated or poll the RTCAIF0 flag of the interrupt request flag register IF1L (for m = 0) or RTCAIF1 flag of the interrupt request flag register IF1H (for m = 1) until the value of the given flag becomes 1.

Remark m = 0, 1

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the RTCAIF0 flag of the interrupt request flag register IF1L (for m = 0) or RTCAIF1 flag of the interrupt request flag register IF1H (for m = 1) is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

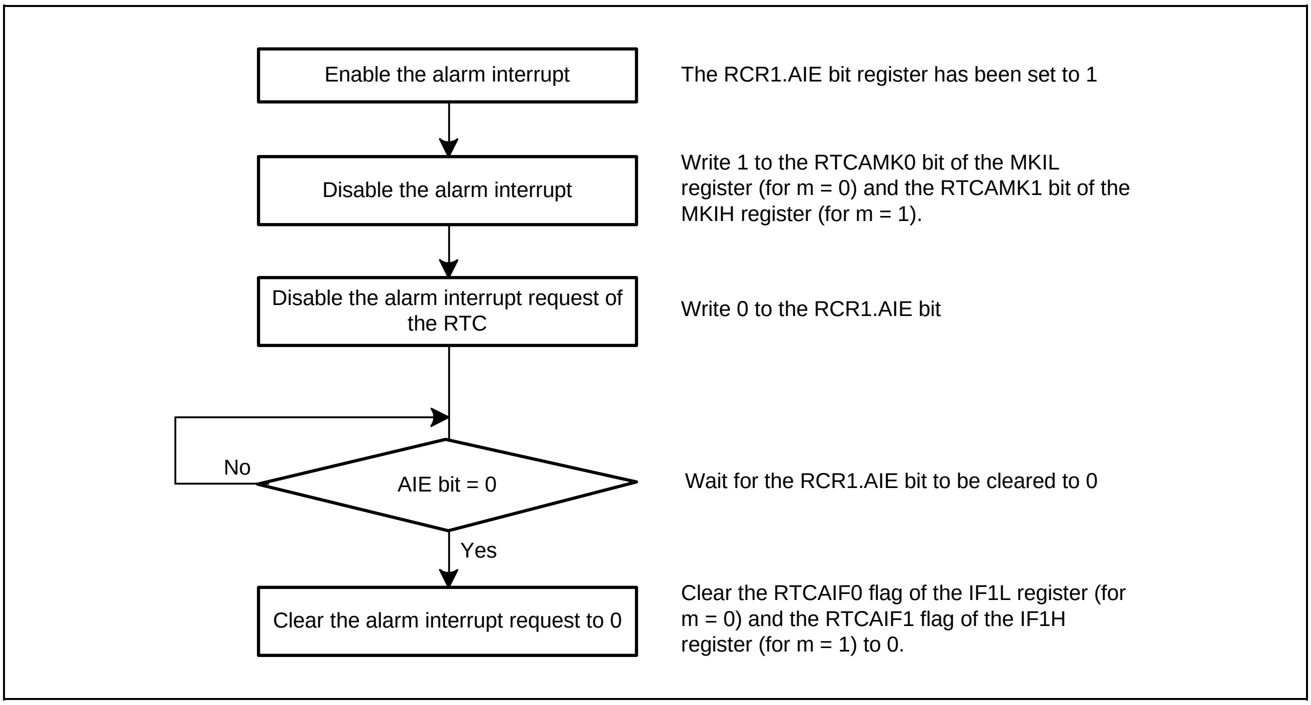
The interrupt request flags of the interrupt request flag registers (IF1H and IF1L) are cleared by writing 0 to these flags.

When the counter and the alarm time match in standby mode, the MCU returns from standby mode.

9.3.9 Procedure for disabling alarm interrupt m

Figure 9-60 shows the procedure for disabling the enabled alarm interrupt m request.

Figure 9-60. Procedure for Disabling Alarm Interrupt m Request


Enable the alarm interrupt
The RCR1.AIE bit register has been set to 1
Disable the alarm interrupt
Write 1 to the RTCAMK0 bit of the MKIL register (for m = 0) and the RTCAMK1 bit of the MKIH register (for m = 1).
Disable the alarm interrupt request of the RTC
Write 0 to the RCR1.AIE bit
No
AIE bit = 0
Wait for the RCR1.AIE bit to be cleared to 0
Yes
Clear the alarm interrupt request to 0
Clear the RTCAIF0 flag of the IF1L register (for m = 0) and the RTCAIF1 flag of the IF1H register (for m = 1) to 0.

Remark m = 0, 1

9.3.10 Time error adjustment function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.

Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

9.3.10.1 Automatic adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.

Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz

Adjustment procedure:

When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz

Adjustment procedure:

At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

9.3.10.2 Adjustment by software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.

Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.

An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

9.3.10.3 Procedure for changing the mode of adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from automatic adjustment to adjustment by software:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

9.3.10.4 Procedure for stopping adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

9.3.10.5 Time capture function

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in **Figure 9-61** and operation when the noise filter is on is shown in **Figure 9-62**.

Figure 9-61. Timing of a Time Capture Function Operation (with the Filter Off)

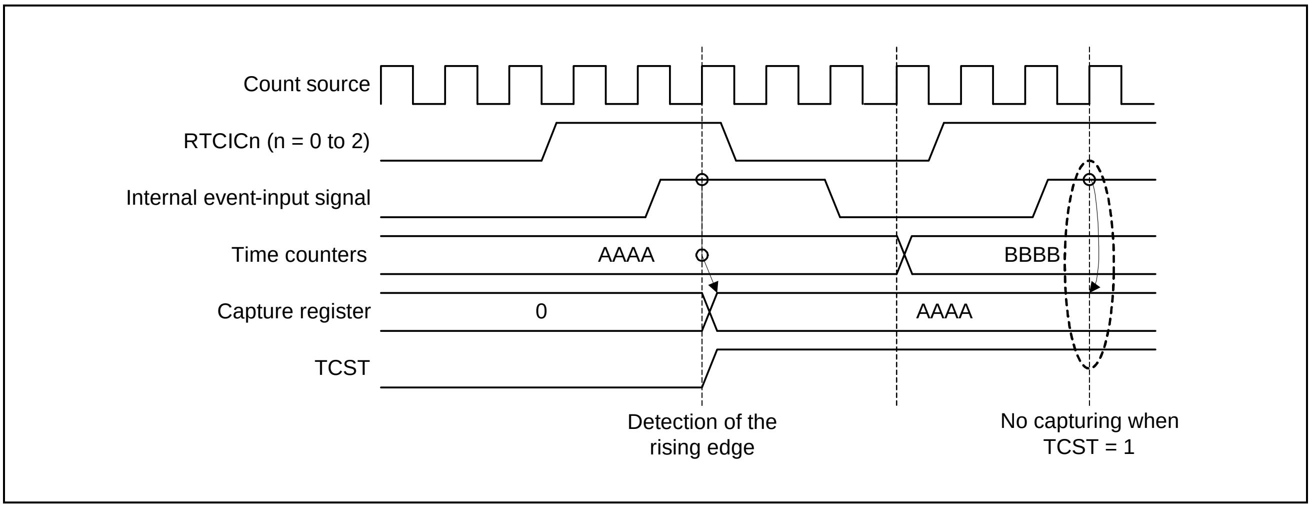
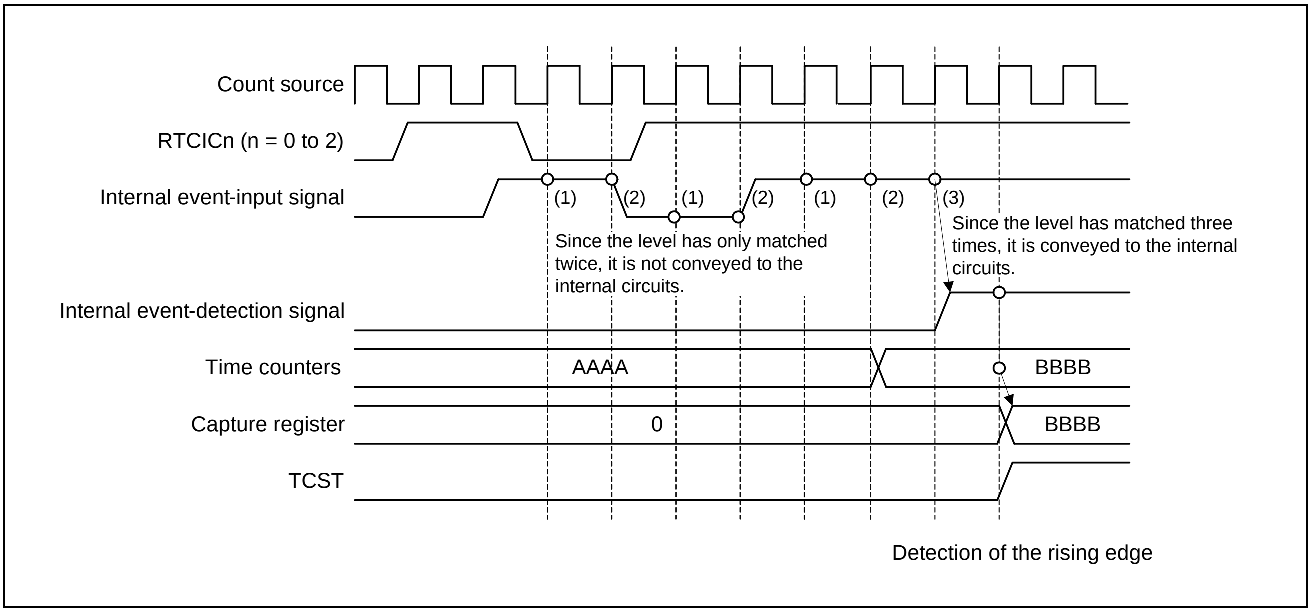

Count source
RTCICn (n = 0 to 2)
Internal event-input signal
Time counters
AAAA
BBBB
Capture register
0
AAAA
TCST
Detection of the rising edge
No capturing when TCST = 1

Figure 9-62. Timing of a Time Capture Function Operation (with the Filter On)

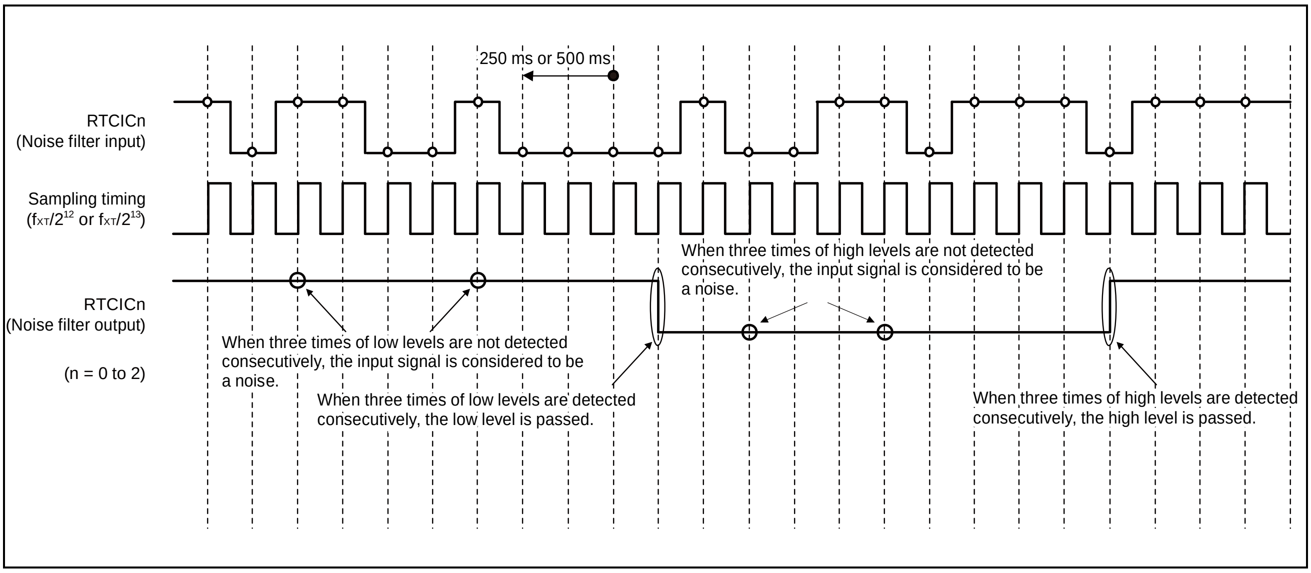

Count source
RTCICn (n = 0 to 2)
Internal event-input signal
(1)
(2)
(1)
(2)
(1)
(2)
(3)
Since the level has only matched twice, it is not conveyed to the internal circuits.
Since the level has matched three times, it is conveyed to the internal circuits.
Internal event-detection signal
Time counters
AAAA
BBBB
Capture register
0
BBBB
TCST
Detection of the rising edge

9.3.11 Noise filter operation for RTCICn pin (n = 0 to 2)

The RTCICn pin (n = 0 to 2) can be used as the RTC time capture event input. A noise filter for the RTCICn pin is incorporated to prevent an unnecessary time capturing caused by chattering of the RTCICn pin. A sampling clock is selected with the RTCICnNF0 bit of the RTCICNFEN register. The RTCICn pin input signal is sampled, and it can pass when matching with the detection level three times continuously.

Example of noise filter operation is shown in **Figure 9-63**.

Figure 9-63. Noise Filter Operation


250 ms or 500 ms
RTCICn
(Noise filter input)
Sampling timing
($f_{XT}/2^{12}$ or $f_{XT}/2^{13}$)
RTCICn
(Noise filter output)
(n = 0 to 2)
When three times of high levels are not detected consecutively, the input signal is considered to be a noise.
When three times of low levels are not detected consecutively, the input signal is considered to be a noise.
When three times of low levels are detected consecutively, the low level is passed.
When three times of high levels are detected consecutively, the high level is passed.

Setting of the noise filter for the RTCICn pin can be selected with two registers (RTCICNFEN and RTCCRn.TCNF1-0). A setting list is shown in the following table.

Table 9-4. Noise Filter Operation for RTCICn Pin (n = 0 to 2)

RCR3	RTCCRn		RTCICNFEN		Noise filter setting for RTCICn pin
RTCICEN	TCNF1	TCNF0	RTCICnNF1	RTCICnNF0	
0	0	0	0	x	RTCICn pin input is invalid.
0	0	0	0	x	Noise filter OFF
			1	0	Noise filter ON (RTC count source divided by 2^{12})
			1	1	Noise filter ON (RTC count source divided by 2^{13})
	1	0	0	0	Noise filter ON (RTC count source)
	1	1	0	0	Noise filter ON (RTC count source divided by 32)
Other than above					Setting prohibited

Remark n = 0 to 2

9.4 Interrupt Sources

There are three interrupt sources in the realtime clock. **Table 9-5** lists interrupt sources for the RTC.

Table 9-5. RTC Interrupt Sources

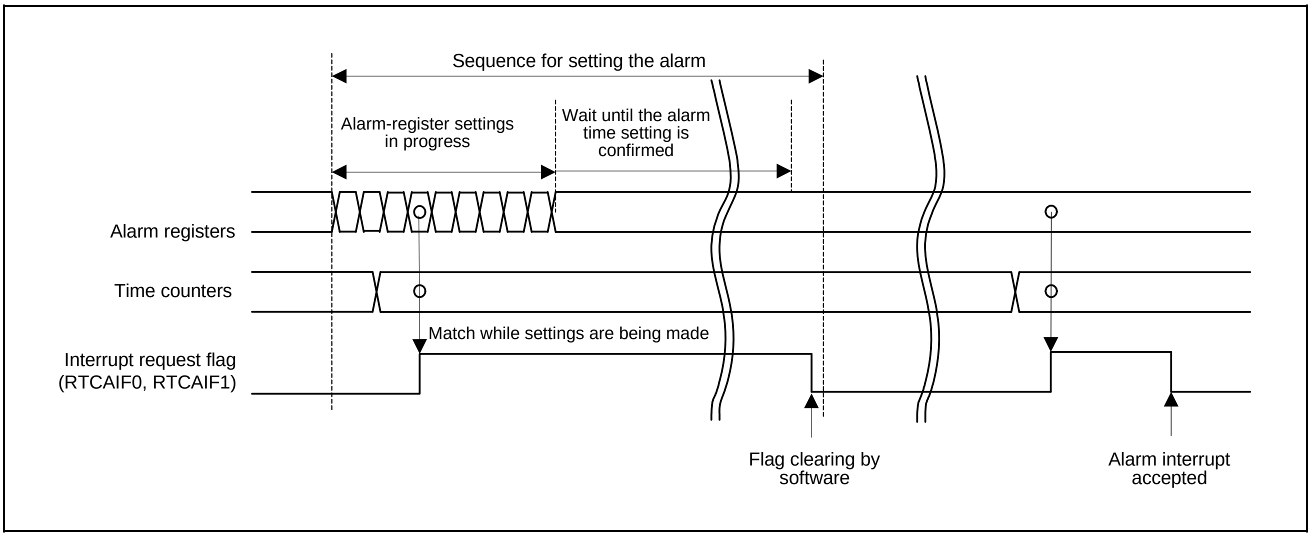
Name	Interrupt Sources
INTRTCALMm	RTC alarm interrupt m (ALM)
INTRTCPRD	Periodic interrupt (PRD)

(1) RTC alarm interrupt m (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and time counters (for details, refer to **9.3.8 Alarm function**).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the time counters, wait for the alarm time settings to be secured and clear the RTCAIF0 flag of the interrupt request flag register IF1L (for m = 0) and the RTCAIF1 flag of the interrupt request flag register IF1H (for m = 1) to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and time counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

Figure 9-64. Timing Chart for the RTC Alarm Interrupt m (ALM)


Sequence for setting the alarm
Alarm-register settings in progress
Wait until the alarm time setting is confirmed
Alarm registers
Time counters
Interrupt request flag (RTCAIF0, RTCAIF1)
Match while settings are being made
Flag clearing by software
Alarm interrupt accepted

Caution Use the RTCAIF0 flag of IF1L (for m = 0) or RTCAIF1 flag of IF1H (for m = 1) to confirm the generation of RTC alarm interrupt m.

Remark m = 0, 1

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

Caution Use the RTCRIF flag of the interrupt request flag register (IF1H) to generate the periodic interrupt.

9.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Caution If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

9.5.1 Interrupt handling and event linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Caution Although alarm and periodic interrupts can still be output during STOP mode, the periodic event signals for the ELC are not output.

9.6 Usage Notes

9.6.1 Register writing during counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

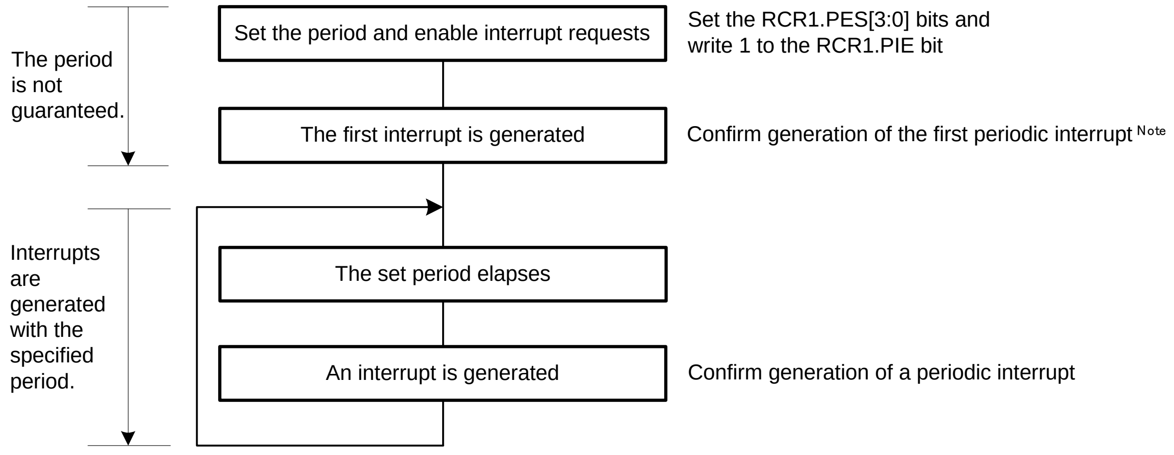
9.6.2 Use of periodic interrupts

The procedure for using periodic interrupts is shown in **Figure 9-65**.

The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.

Furthermore, stopping and starting the counter, resetting of the RTC by software, and 30-second adjustment by changing the value of the RCR2 register, and 0.5-second adjustment by changing the value of the RCR4 register all affect the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

Figure 9-65. Using Periodic Interrupt Function


The period is not guaranteed.
Set the period and enable interrupt requests
Set the RCR1.PES[3:0] bits and write 1 to the RCR1.PIE bit
The first interrupt is generated
Confirm generation of the first periodic interrupt Note
Interrupts are generated with the specified period.
The set period elapses
An interrupt is generated
Confirm generation of a periodic interrupt

Note When a interrupt generation period is changed while the periodic interrupt is used, an interrupt may be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

9.6.3 RTCOUT (1-Hz/64-Hz) clock output

Furthermore, stopping and starting the counter, resetting of the RTC by software, and 30-second adjustment by changing the value of the RCR2 register, and 0.5-second adjustment by changing the value of the RCR4 register all affect the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

9.6.4 Notes when writing to and reading from registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in **9.3.6 Reading 64-Hz counter and time**.
- Values written to the count registers, alarm registers, year alarm enable register, AADJE, AADJP, and HR24 bits of the RCR2 register, RCR3 register, or RCR4 register will be read correctly from the 4th cycle of the CPU clock (f_{CLK}) after writing.
- Values written to the SCMC, SCSC, RTCPORSR, RTCICNFEN, and RSR registers, RCR1.RTCOS and RCR2.RTCOE bits can be read immediately after writing.
- After a reset is generated, write to the RTC register when six cycles of the count source clock have elapsed. When the power supply from the VRTC pin is stopped, setting an RTC related register is prohibited.

9.6.5 Changing the count mode

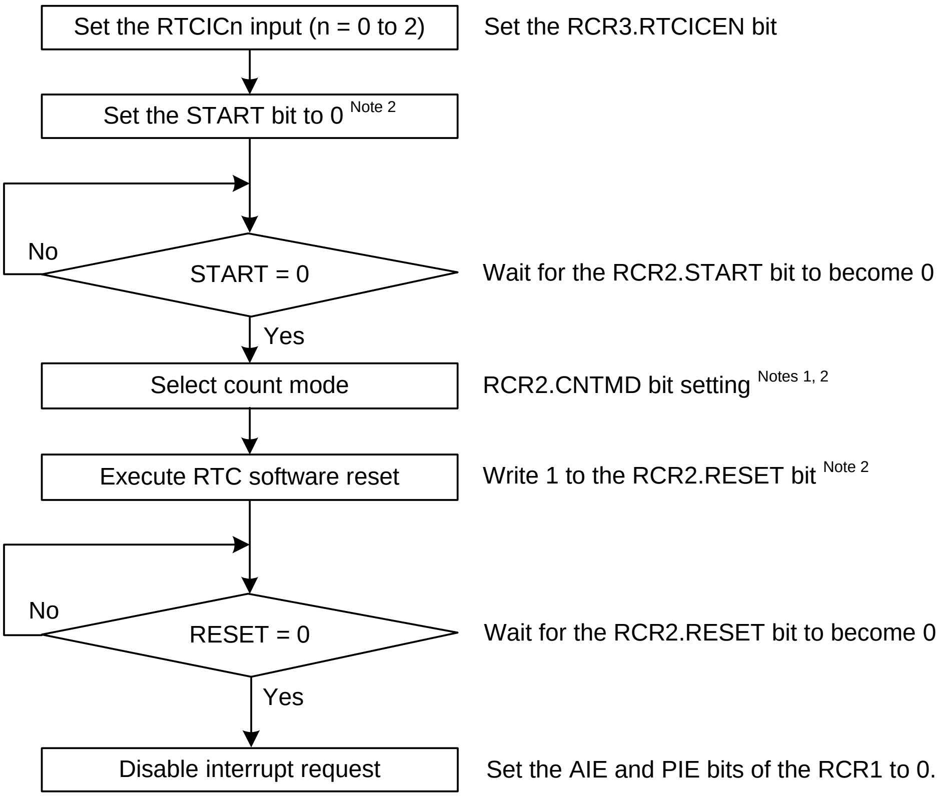
When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to **9.3.1 Outline of initial settings of registers after power on**.

9.6.6 Stop procedure

The operation of the realtime clock with independent power supply is undefined immediately after release from the RTC power-on reset state.

When it is not used, stop it according to following procedure shown in **Figure 9-66**.

Figure 9-66. Stop Setting Procedure


Set the RTCICn input (n = 0 to 2)
Set the RCR3.RTCICEN bit
Set the START bit to 0 Note 2
No
START = 0
Wait for the RCR2.START bit to become 0
Yes
Select count mode
RCR2.CNTMD bit setting Notes 1, 2
Execute RTC software reset
Write 1 to the RCR2.RESET bit Note 2
No
RESET = 0
Wait for the RCR2.RESET bit to become 0
Yes
Disable interrupt request
Set the AIE and PIE bits of the RCR1 to 0.

Notes 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0.

2. Rewriting the RCR2 register with a way of read-modify-writing is prohibited.

9.6.7 Point for caution on time capture event function

In the RL78/I1C (512 KB), even when the supply of the power from the V_{DD} power supply pin is stopped, the time capture function of the time capture event input pin (RTCICn) can be used.

CHAPTER 10 FREQUENCY MEASURE CIRCUIT

10.1 Frequency Measurement Circuit

The frequency measurement circuit is used to measure the frequency of the sub clock (f_{SX}) or low-speed on-chip oscillator clock (f_{IL}), by inputting the high-accuracy reference clock externally.

10.2 Configuration of Frequency Measurement Circuit

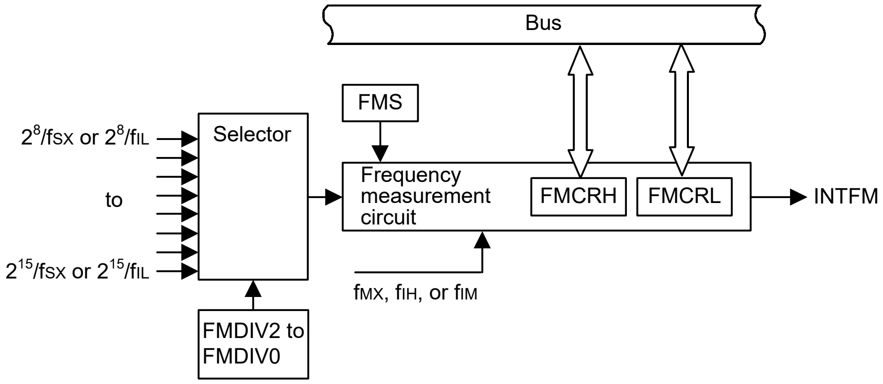
The frequency measurement circuit includes the following hardware.

Table 10-1. Configuration of Frequency Measurement Circuit

Item	Configuration
Counter	Counter (32-bit)
Control registers	Peripheral enable register 1 (PER1)
	Subsystem clock supply option control register (OSMC)
	Frequency measurement count register L (FMCRL)
	Frequency measurement count register H (FMCRH)
	Frequency measurement control register (FMCTL)
	Frequency measurement clock select register (FMCKS)

Figure 10-1 shows the frequency measurement circuit diagram.

Figure 10-1. Frequency Measurement Circuit Diagram


Bus
FMS
$2^8/f_{SX}$ or $2^8/f_{IL}$
to
$2^{15}/f_{SX}$ or $2^{15}/f_{IL}$
Selector
Frequency measurement circuit
FMCRH
FMCRL
INTFM
f_{MX}, f_{IH}, or f_{IM}
FMDIV2 to FMDIV0

10.3 Registers Controlling Frequency Measurement Circuit

The frequency measurement circuit is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Subsystem clock supply option control register (OSMC)
- Frequency measurement count register L (FMCRL)
- Frequency measurement count register H (FMCRH)
- Frequency measurement control register (FMCTL)
- Frequency measurement clock select register (FMCKS)

10.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

Of the registers that are used to control the frequency measurement circuit can be set by setting bit 6 (FMCEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

FMCEN	Control of frequency measurement circuit input clock supply
0	Stops input clock supply. ● SFR used by the frequency measurement circuit cannot be written. The read value is 00H. ● The frequency measurement circuit and SFR are in the reset state.
1	Enables input clock supply. ● SFR used by the frequency measurement circuit can be read and written.

Cautions 1. Be sure to clear bit 7 to 0.

2. Do not change the target bit in the PER1 register while operation of each peripheral function is enabled. Change the setting specified by PER1 while operation of each peripheral function assigned to PER1 is stopped.

10.3.2 Subsystem clock supply option control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 is stopped in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock.

In addition, the OSMC register can be used to select the operating clock of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

The low-speed on-chip oscillator clock cannot be selected as the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1. When the serial interface UARTMG0 or UARTMG1, sampling output timer detector 0, or sampling output timer detector 1 is to be used, select the sub clock (f_{SX}) as the operating clock by setting the WUTMMCK0 bit to 0.

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

Reset signal generation clears this register to 00H.

Figure 10-3. Format of Subsystem Clock Supply Option Control Register (OSMC)

Address: F00F3H After reset: 00H R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL}) Notes 2, 3, 5, 6	Low-speed on-chip oscillator clock (f_{IL}) selected Note 5	Clock output is prohibited. Note 4	Setting prohibited

Notes 1. Be sure to clear bits 6, 5, and 3 to 0 to 0.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (f_{IL}) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

5. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

6. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

10.3.3 Frequency measurement count register L (FMCRL)

This register represents the lower 16 bits of the frequency measurement count register (FMCR) in the frequency measurement circuit.

The FMCRL register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears the FMCRL register to 0000H.

Figure 10-4. Format of Frequency Measurement Count Register L (FMCRL)

Address:F0312H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCRL																

Cautions 1. Do not read the value of FMCRL when FMS = 1.

2. Read the value of FMCRL after the frequency measurement complete interrupt is generated.

10.3.4 Frequency measurement count register H (FMCRH)

This register represents the upper 16 bits of the frequency measurement count register (FMCR) in the frequency measurement circuit.

The FMCRH register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears the FMCRH register to 0000H.

Figure 10-5. Format of Frequency Measurement Count Register H (FMCRH)

Address: F0314H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCRH																

Cautions 1. Do not read the value of FMCRH when FMS = 1.

2. Read the value of FMCRH after the frequency measurement complete interrupt is generated.

Figure 10-6. Frequency Measurement Count Register (FMCRH, FMCRL)

	31 16	15 0
FMCR	FMCRH	FMCRL

10.3.5 Frequency measurement control register (FMCTL)

The FMCTL register is used to set the operation of the frequency measurement circuit. This register is used to start operation and set the period of frequency measurement.

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

Reset signal generation clears the FMCTL register to 00H.

Figure 10-7. Format of Frequency Measurement Control Register (FMCTL)

Address: F0316H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
FMCT	FMS	0	0	0	0	FMDIV2	FMDIV1	FMDIV0

FMS	Frequency measurement circuit operation enable
0	Stops the frequency measurement circuit.
1	Operates the frequency measurement circuit. Starts counting on the rising edge of the operating clock and stops counting on the next rising edge of the operating clock.

FMDIV2	FMDIV1	FMDIV0	Frequency measurement period setting
0	0	0	$2^8/f_{SX}$ or $2^8/f_{IL}$ (7.8125 ms)
0	0	1	$2^9/f_{SX}$ or $2^9/f_{IL}$ (15.625 ms)
0	1	0	$2^{10}/f_{SX}$ or $2^{10}/f_{IL}$ (31.25 ms)
0	1	1	$2^{11}/f_{SX}$ or $2^{11}/f_{IL}$ (62.5 ms)
1	0	0	$2^{12}/f_{SX}$ or $2^{12}/f_{IL}$ (0.125 s)
1	0	1	$2^{13}/f_{SX}$ or $2^{13}/f_{IL}$ (0.25 s)
1	1	0	$2^{14}/f_{SX}$ or $2^{14}/f_{IL}$ (0.5 s)
1	1	1	$2^{15}/f_{SX}$ or $2^{15}/f_{IL}$ (1s)

Caution Do not read the value of the FMDIV2 to FMDIV0 bits when FMS = 1.

Remark The frequency measurement resolution can be calculated by the formula below.

- Frequency measurement resolution = 10^6/(frequency measurement period × reference clock frequency (f_{MX}) [Hz]) [ppm]

  Example 1) When FMDIV2 to FMDIV0 = 000B and f_{MX} = 20 MHz, measurement resolution = 6.4 ppm

  Example 2) When FMDIV2 to FMDIV0 = 111B and f_{MX} = 1 MHz, measurement resolution = 1 ppm

10.3.6 Frequency measurement clock select register (FMCKS)

The FMCKS register is used to select the operating clock and frequency count clock to be input to the frequency measurement circuit.

The FMCKS register can be used by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the FMCKS register to 00H.

Figure 10-8. Format of Frequency Measurement Clock Select Register (FMCKS)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FMCKS	0	0	0	0	0	0	FMCKSEL1	FMCKSEL0

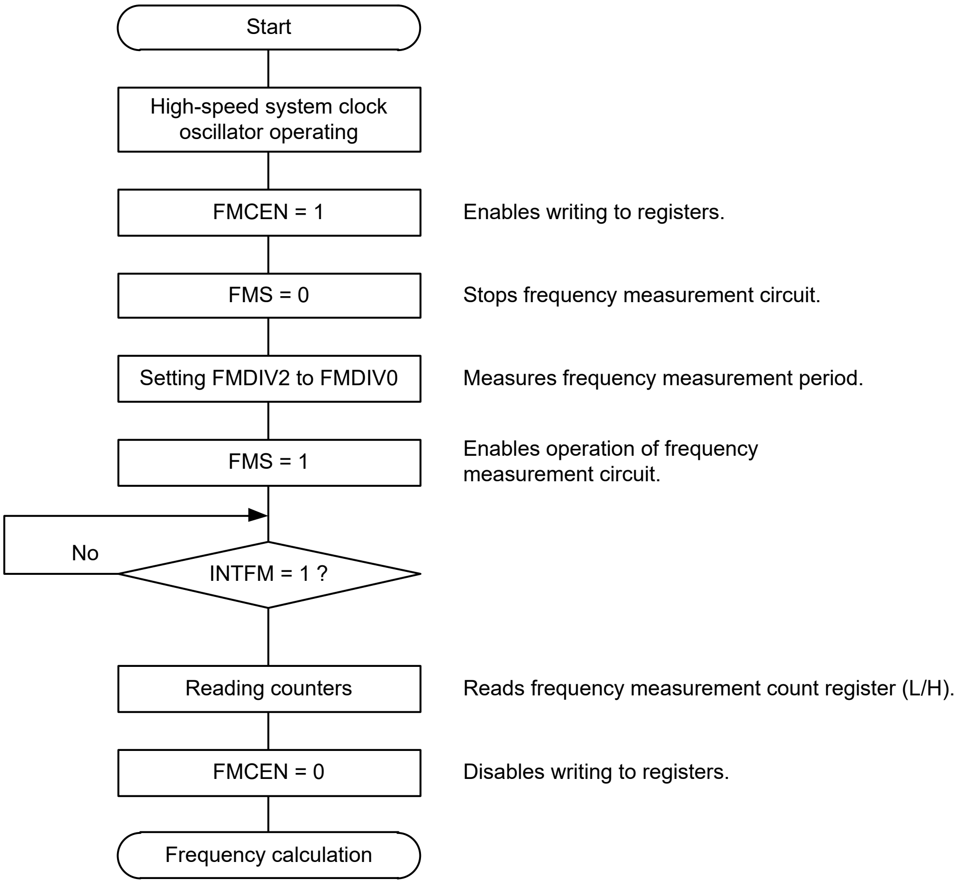
FMCKSEL1	FMCKSEL0	Selection of frequency count clock
0	0	f_{MX} selected
0	1	f_{IM} selected
1	×	f_{IH} selected

10.4 Frequency Measurement Circuit Operation

10.4.1 Setting frequency measurement circuit

Set frequency measurement circuit after setting 0 to FMS first.

Figure 10-9. Procedure for Setting Frequency Measurement Circuit Using Reference Clock


Start
High-speed system clock oscillator operating
FMCEN = 1
Enables writing to registers.
FMS = 0
Stops frequency measurement circuit.
Setting FMDIV2 to FMDIV0
Measures frequency measurement period.
FMS = 1
Enables operation of frequency measurement circuit.
No
INTFM = 1 ?
Reading counters
Reads frequency measurement count register (L/H).
FMCEN = 0
Disables writing to registers.
Frequency calculation

Caution After the frequency measurement count register (L/H) is read, be sure to set FMCEN to 0.

The f_{SX} or f_{IL} oscillation frequency is calculated by using the following expression.

$$f_{SX} \text{ or } f_{IL} \text{ oscillation frequency} = \frac{\text{Reference clock frequency [Hz]} \times \text{operation trigger division ratio}}{\text{Frequency measurement count register value (FMCR)}} \text{ [Hz]}$$

For example, when the frequency is measured under the following conditions

- Count clock frequency: f_{MX} = 10 MHz
- Frequency measurement period setting register: FMDIV2 to FMDIV0 = 111B (operation trigger division ratio: 2^{15})

and the measurement result is as follows,

- Frequency measurement count register: FMCR = 10000160D

the f_{SX} or f_{IL} oscillation frequency is obtained as below.

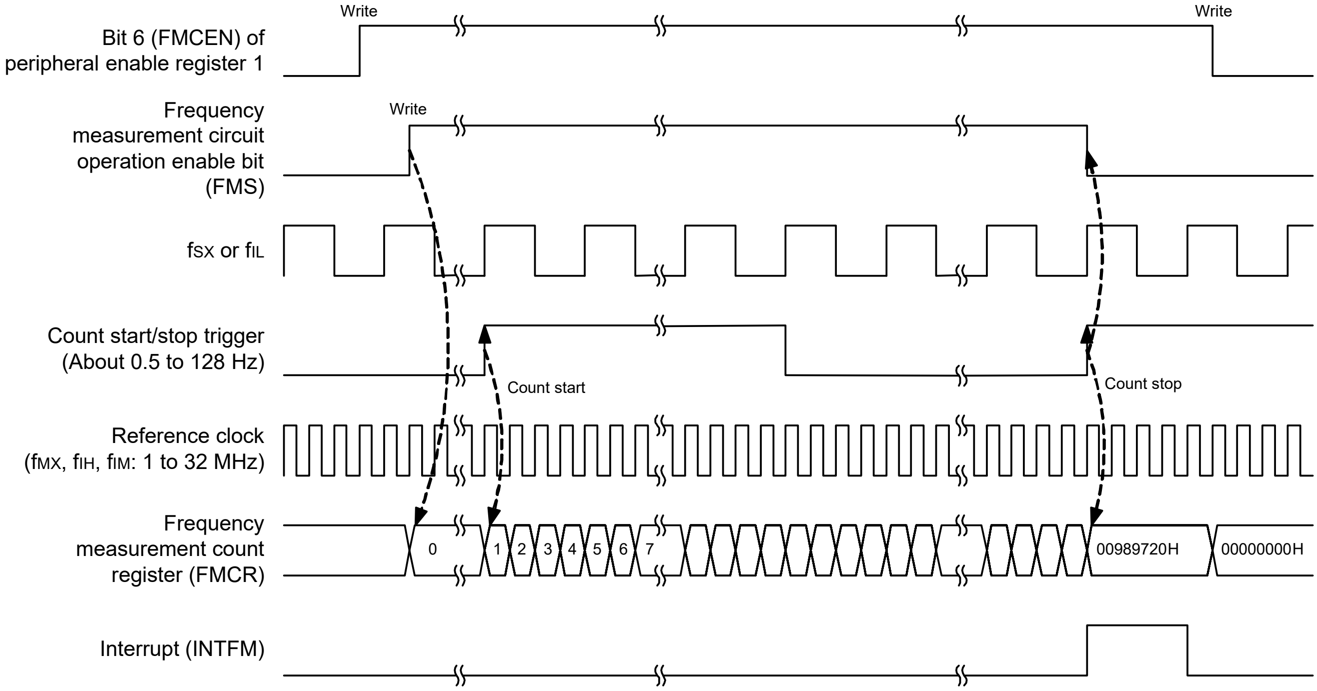
$$f_{SX} \text{ or } f_{IL} \text{ oscillation frequency} = \frac{(10 \times 10^6) \times 2^{15}}{10000160} = 32767.47572 \text{ [Hz]}$$

10.4.2 Frequency measurement circuit operation timing

The operation timing of the frequency measurement circuit is shown in **Figure 10-10**.

After the frequency measurement circuit operation enable bit (FMS) is set to 1, counting is started by the count start trigger set with the frequency measurement period setting bits (FMDIV2 to FMDIV0) and stopped by the next trigger. After counting is stopped, the count value is retained, and the frequency measurement circuit operation enable bit (FMS) is reset to 0. An interrupt is also generated for one clock of fSX or fIL. After the operation of the frequency measurement circuit is completed (FMS = 0) and the frequency measurement count register (L/H) is read, be sure to set bit 6 (FMCEN) of peripheral enable register 1 to 0.

Figure 10-10. Frequency Measurement Circuit Operation Timing


Bit 6 (FMCEN) of peripheral enable register 1
Write
Write
Frequency measurement circuit operation enable bit (FMS)
Write
fSX or fIL
Count start/stop trigger (About 0.5 to 128 Hz)
Count start
Count stop
Reference clock (fMX, fIH, fIM: 1 to 32 MHz)
Frequency measurement count register (FMCR)
0
1
2
3
4
5
6
7
00989720H
00000000H
Interrupt (INTFM)

CHAPTER 11 12-BIT INTERVAL TIMER

11.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode.

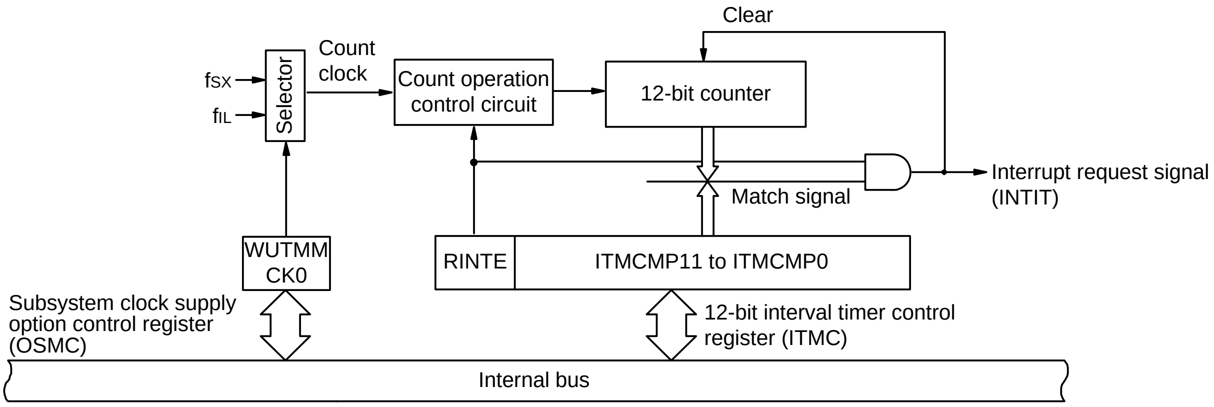
11.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 11-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 2 (PER2)
	Peripheral reset control register 2 (PRR2)
	Subsystem clock supply option control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 11-1. Block Diagram of 12-bit Interval Timer


Clear
fSX
fIL
Selector
Count clock
Count operation control circuit
12-bit counter
Match signal
Interrupt request signal (INTIT)
WUTMMCK0
RINTE
ITMCMP11 to ITMCMP0
Subsystem clock supply option control register (OSMC)
12-bit interval timer control register (ITMC)
Internal bus

11.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Subsystem clock supply option control register (OSMC)
- 12-bit interval timer control register (ITMC)

11.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the 12-bit interval timer is to be used, be sure to set bit 7 (TMKAEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-2. Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. ● SFR used by the 12-bit interval timer cannot be written. The read value is 00H. However, the SFR is not initialized. Note
1	Enables input clock supply. ● SFR used by the 12-bit interval timer can be read and written.

Note To initialize the 12-bit interval timer and the SFR used by the 12-bit interval timer, use bit 7 (TMKARES) of PRR2.

Cautions 1. Be sure to clear the following bits to 0.
Bits 3 and 1

2. Do not change the target bit in the PER2 register while operation of each peripheral function is enabled. Change the setting specified by PER2 while operation of each peripheral function assigned to PER2 is stopped (except for FMCEN).

11.3.2 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.

To place the 12-bit interval timer in the reset state, be sure to set bit 7 (TMKARES) to 1.

The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-3. Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	0
PRR2	TMKARES	OSDCRES	UARTMG1 RES	UARTMG0 RES	0	MACRES	0	0

TMKARES	Control resetting of the 12-bit interval timer
0	The 12-bit interval timer is released from the reset state.
1	The 12-bit interval timer is in the reset state.

11.3.3 Subsystem clock supply option control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer operation clock.

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

Reset signal generation clears this register to 00H.

Figure 11-4. Format of Subsystem Clock Supply Option Control Register (OSMC)

Address: F00F3H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL})[Notes 2, 3, 5, 6]	Low-speed on-chip oscillator clock (f_{IL}) selected[Note 5]	Clock output is prohibited. [Note 4]	Setting prohibited

Notes 1. Be sure to clear bits 6, 5, and 3 to 0 to 0.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (f_{IL}) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

5. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

6. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

11.3.4 12-bit interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 11-5. Format of 12-bit Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate a fixed-cycle interrupt (count clock cycles × (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0

- ITCMP11 to ITCMP0 = 001H, count clock: when f_{SX} = 32.768 kHz
  1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] $\cong$ 61.03 [µs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SX} = 32.768 kHz
  1/32.768 [kHz] × (4095 + 1) = 125 [ms]

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the TMKAIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

4. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

11.4 12-bit Interval Timer Operation

11.4.1 12-bit interval timer operation timing

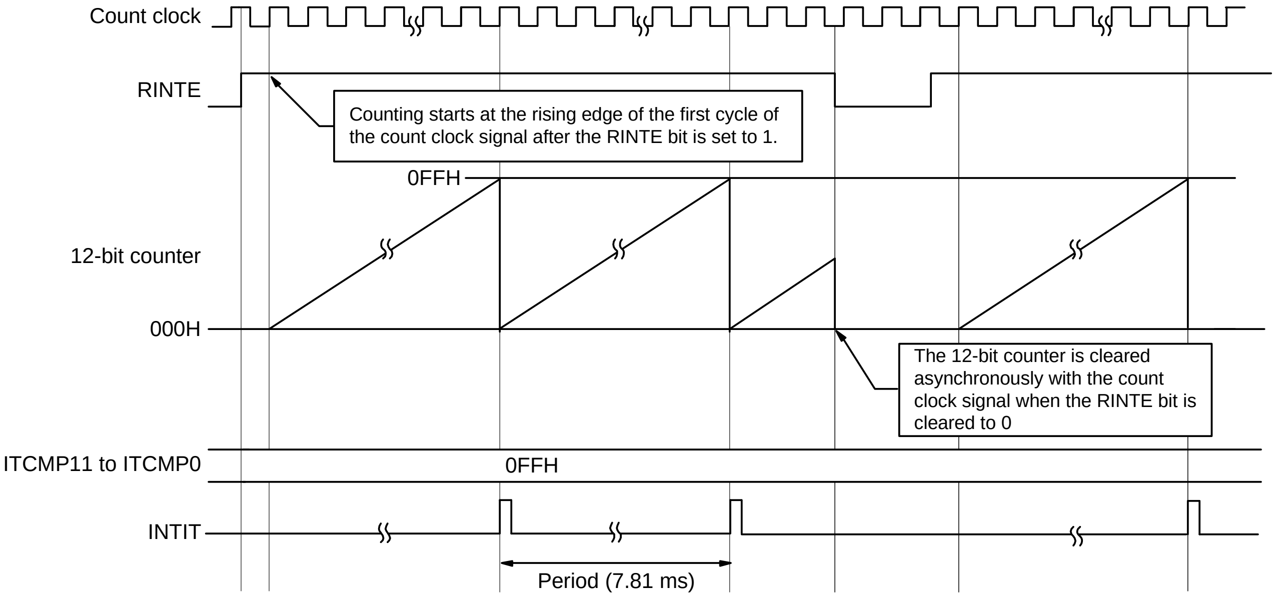
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate a 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 11-6. 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SX} = 32.768 kHz)


Count clock
RINTE
Counting starts at the rising edge of the first cycle of the count clock signal after the RINTE bit is set to 1.
0FFH
12-bit counter
000H
The 12-bit counter is cleared asynchronously with the count clock signal when the RINTE bit is cleared to 0
ITCMP11 to ITCMP0
0FFH
INTIT
Period (7.81 ms)

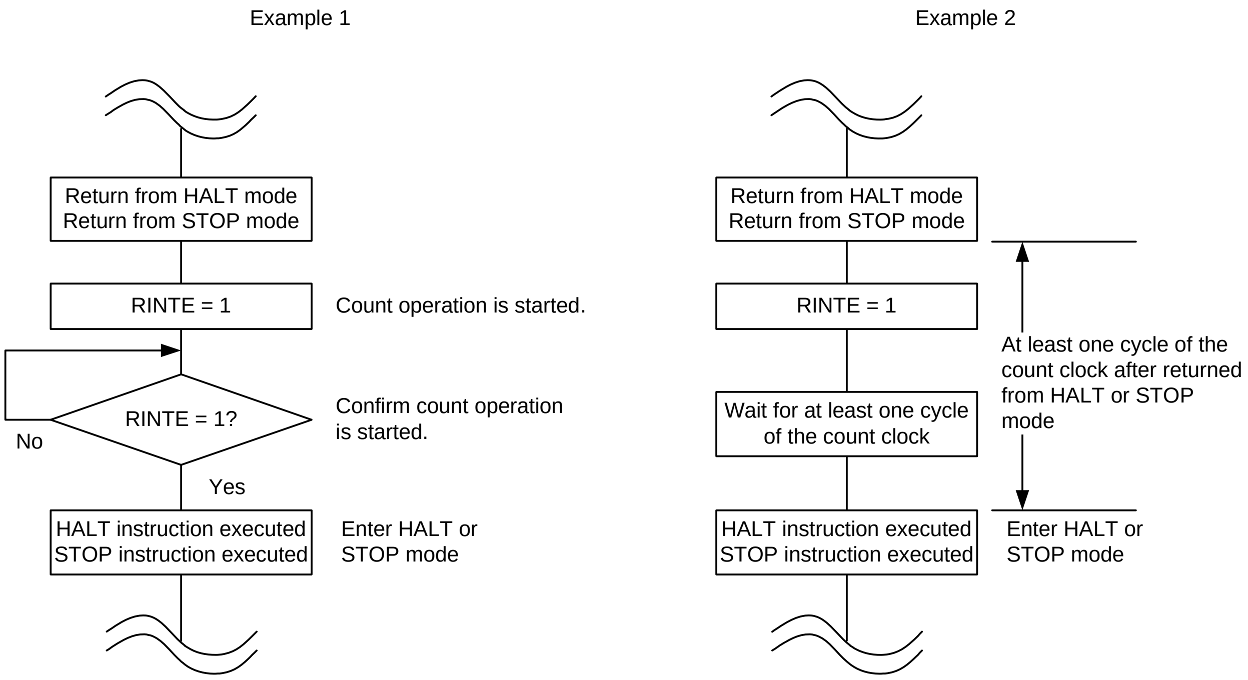
11.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.

Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 11-7**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 11-7**).

Figure 11-7. Procedure of Entering to HALT or STOP Mode after Setting RINTE to 1



CHAPTER 12 8-BIT INTERVAL TIMER

The RL78/I1C (512 KB) has four 8-bit interval timers.

Each 8-bit interval timer has two 8-bit timers 0 and 1, and the timers operate independently of each other. These 8-bit timers can also be combined to function as a 16-bit timer.

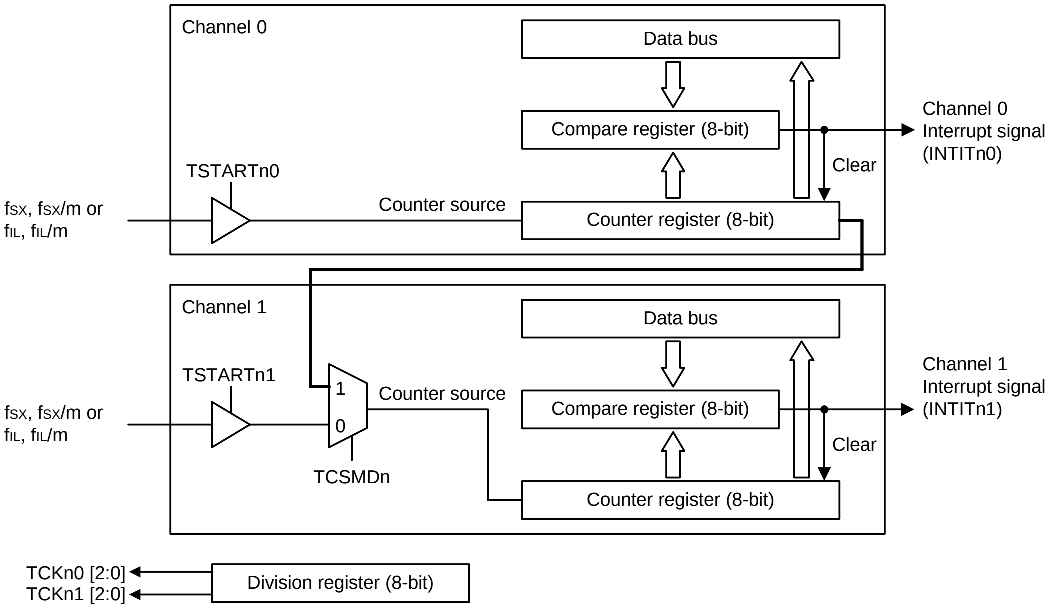
12.1 Overview

The 8-bit interval timer is an 8-bit timer that operates using the f_{SX} or f_{IL} clock that is asynchronous with the CPU. **Table 12-1** lists the 8-bit interval timer specifications and **Figure 12-1** shows the 8-bit interval timer block diagram.

Table 12-1. 8-Bit Interval Timer Specifications

Item	Description
Count source (operating clock)	● f_{SX}, $f_{SX}/2$, $f_{SX}/4$, $f_{SX}/8$, $f_{SX}/16$, $f_{SX}/32$, $f_{SX}/64$, $f_{SX}/128$ ● f_{IL}, $f_{IL}/2$, $f_{IL}/4$, $f_{IL}/8$, $f_{IL}/16$, $f_{IL}/32$, $f_{IL}/64$, $f_{IL}/128$
Operating mode	● 8-bit counter mode Channel 0 and channel 1 operate independently as an 8-bit counter ● 16-bit counter mode Channel 0 and channel 1 are connected to operate as a 16-bit counter
Interrupt	● Output when the counter matches the compare value

Figure 12-1. 8-Bit Interval Timer Block Diagram



TSTARTni (i = 0, 1), TCSMDn, TCLKENn: Bits in TRTCRn register

TCKni [2:0]: Bit in TRTMDn register

Remark m = 2, 4, 8, 16, 32, 64, 128

n = 0 to 3

12.2 I/O Pins

The 8-bit interval timer does not have an I/O pin.

12.3 Registers

Table 12-2 lists the 8-bit interval timer register configuration.

Table 12-2. Registers

Item	Configuration
Control registers	8-bit interval timer counter register n0 (TRTn0)[Note 1]
	8-bit interval timer counter register n1 (TRTn1)[Note 1]
	8-bit interval timer counter register n (TRTn)[Note 2]
	8-bit interval timer compare register n0 (TRTCMPn0)[Note 1]
	8-bit interval timer compare register n1 (TRTCMPn1)[Note 1]
	8-bit interval timer compare register n (TRTCMPn)[Note 2]
	8-bit interval timer control register n (TRTCRn)
	8-bit interval timer division register n (TRTMDn)

Notes 1. Can be accessed only when the TCSMDn bit in the TRTCRn register = 0.

2. Can be accessed only when the TCSMDn bit in the TRTCRn register = 1.

Remark n = 0 to 3

12.3.1 8-bit interval timer counter register ni (TRTni) (n = 0 to 3, i = 0, 1)

This is the 8-bit interval timer counter register. It is used as a counter that counts up based on the count clock.

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

Reset signal generation clears this register to 00H.

Figure 12-2. Format of 8-bit Interval Timer Counter Register ni (TRTni)

Address: F0540H (TRT00), F0541H (TRT01), F0548H (TRT10), F0549H (TRT11), After reset: 00H R[Notes 1, 2]
F0550H (TRT20), F0551H (TRT21), F0558H (TRT30), F0559H (TRT31)

Symbol	7	6	5	4	3	2	1	0
TRTni								

Notes 1. The TRTni register is set to 00H two cycles of the count clock after the compare register TRTCMPni is write-accessed. Refer to **12.4.4 Timing for updating compare register values**.

2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

12.3.2 8-bit interval timer counter register n (TRTn) (n = 0 to 3)

This is a 16-bit counter register when the 8-bit interval timer is used in 16-bit interval timer mode.

The TRTn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0000H.

Figure 12-3. Format of 8-bit Interval Timer Counter Register n (TRTn)

Address: F0540H (TRT0), F0548H (TRT1), After reset: 0000H R[Notes 1, 2]
F0550H (TRT2), F0558H (TRT3)

	F0541H (TRT01) F0549H (TRT11) F0551H (TRT21) F0559H (TRT31)								F0540H (TRT00) F0548H (TRT10) F0550H (TRT20) F0558H (TRT30)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRTn																

Notes 1. The TRTn register is set to 0000H two cycles of the count clock after the compare register TRTCMPn is write-accessed. Refer to **12.4.4 Timing for updating compare register values**.

2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

12.3.3 8-bit interval timer compare register ni (TRTCMPni) (n = 0 to 3, i = 0, 1)

This is the 8-bit interval timer compare value register.

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

Reset signal generation sets this register to FFH.

The setting range is 01H to FFH[Note 1].

This register is used to store the compare value of registers TRTn0 and TRTn1 (counters).

Write-access clears the count value (TRTn0, TRTn1) to 00H.

Refer to **12.4.4 Timing for updating compare register values** for the timing of rewriting the compare value.

Figure 12-4. Format of 8-bit Interval Timer Compare Register ni (TRTCMPni)

Address: F0350H (TRTCMP00), F0351H (TRTCMP01), After reset: FFH R/W[Notes 1, 2]
F0358H (TRTCMP10), F0359H (TRTCMP11),
F0360H (TRTCMP20), F0361H (TRTCMP21),
F0368H (TRTCMP30), F0369H (TRTCMP31)

Symbol	7	6	5	4	3	2	1	0
TRTCMPni								

Notes 1. The TRTCMPni register must not be set to 00H.

2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

12.3.4 8-bit interval timer compare register n (TRTCMPn) (n = 0 to 3)

This is a compare value register when the 8-bit interval timer is used in 16-bit interval timer mode. The TRTCMPn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to FFFFH. The setting is 0001H to FFFFH[Note 1].

This register is used to store the compare value of the TRTn register (counter).

Write-access clears the count value (TRTn) to 0000H.

Refer to **12.4.4 Timing for updating compare register values** for the timing of rewriting the compare value.

Figure 12-5. Format of 8-bit Interval Timer Compare Register n (TRTCMPn)

Address: F0350H (TRTCMP0), F0358H (TRTCMP1), After reset: FFFFH R/W[Note 2]
F0360H (TRTCMP2), F0368H (TRTCMP3)

	Bits 15 to 8	Bits 7 to 0
	F0351H (TRTCMP01) F0359H (TRTCMP11) F0361H (TRTCMP21) F0369H (TRTCMP31)	F0350H (TRTCMP00) F0358H (TRTCMP10) F0360H (TRTCMP20) F0368H (TRTCMP30)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRTCMPn																

Notes 1. The TRTCMPn register must not be set to 0000H.

2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 1.

12.3.5 8-bit interval timer control register n (TRTCRn) (n = 0 to 3)

This register is used to start and stop counting by the 8-bit interval timer and to switch between using the 8-bit interval timer as an 8-bit counter or a 16-bit counter.

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

Reset signal generation resets this register to 00H.

Figure 12-6. Format of 8-bit Interval Timer Control Register n (TRTCRn)

Address: F0352H (TRTCR0), F035AH (TRTCR1), After reset: 00H R/W[Note 3]
F0362H (TRTCR2), F036AH (TRTCR3)

Symbol	7	6	5	4	3	<2>	1	<0>
TRTCRn	TCSMDn	0	0	TCLKENn	0	TSTARTn1	0	TSTARTn0

TCSMDn	Mode select
0	Operates as 8-bit counter
1	Operates as 16-bit counter (channel 0 and channel 1 are connected)
Refer to **12.4 Operation** for details.	

TCLKENn	8-bit interval timer clock enable[Note 1]
0	Clock is stopped
1	Clock is supplied

TSTARTn1	8-bit interval timer 1 count start[Notes 1, 2]
0	Count stops
1	Count starts
In 8-bit interval timer mode, writing 1 to the TSTARTn1 bit starts the TRTn1 count and writing 0 stops the count. In 16-bit interval timer mode, this bit is invalid because it is not used. Refer to **12.4 Operation** for details.	

TSTARTn0	8-bit interval timer 0 count start[Notes 1, 2]
0	Count stops
1	Count starts
In 8-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn0 count and writing 0 stops the count. In 16-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn count and writing 0 stops the count. Refer to **12.4 Operation** for details.	

Notes 1. Be sure to set the TCLKENn bit to 1 before setting the 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 (f_{SX} or f_{IL}) have elapsed. Refer to **12.5.3 8-bit interval timer setting procedure** for details.

2. Refer to **12.5.1 Changing settings of operating mode** for the notes on using bits TSTARTn0, TSTARTn1, and TCSMDn.

3. Bits 6, 5, 3, and 1 are read-only. When writing, write 0. When reading, 0 is read.

12.3.6 8-bit interval timer division register n (TRTMDn) (n = 0 to 3)

This register is used to select the division ratio of the count source used by the 8-bit interval timer.

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

Reset signal generation sets this register to 00H.

Figure 12-7. Format of 8-bit Interval Timer Division Register n (TRTMDn)

Address: F0353H (TRTMD0), F035BH (TRTMD1), After reset: 00H R/W[Note 4]
F0363H (TRTMD2), F036BH (TRTMD3)

Symbol	7	6	5	4	3	2	1	0
TRTMDn	—		TCKn1		—		TCKn0	

TCKn1			8-bit interval timer 1 division select[Notes 1, 2, 3]
Bit 6	Bit 5	Bit 4	
0	0	0	f_{SX} or f_{IL}
0	0	1	$f_{SX}/2$ or $f_{IL}/2$
0	1	0	$f_{SX}/4$ or $f_{IL}/4$
0	1	1	$f_{SX}/8$ or $f_{IL}/8$
1	0	0	$f_{SX}/16$ or $f_{IL}/16$
1	0	1	$f_{SX}/32$ or $f_{IL}/32$
1	1	0	$f_{SX}/64$ or $f_{IL}/64$
1	1	1	$f_{SX}/128$ or $f_{IL}/128$
In 8-bit interval timer mode, TRTn1 counts using the count source set in TCKn1. In 16-bit interval timer mode, set these bits to 000 because they are not used. Refer to **12.4 Operation** for details.			

TCKn0			8-bit interval timer 0 division select[Notes 1, 2, 3]
Bit 2	Bit 1	Bit 0	
0	0	0	f_{SX} or f_{IL}
0	0	1	$f_{SX}/2$ or $f_{IL}/2$
0	1	0	$f_{SX}/4$ or $f_{IL}/4$
0	1	1	$f_{SX}/8$ or $f_{IL}/8$
1	0	0	$f_{SX}/16$ or $f_{IL}/16$
1	0	1	$f_{SX}/32$ or $f_{IL}/32$
1	1	0	$f_{SX}/64$ or $f_{IL}/64$
1	1	1	$f_{SX}/128$ or $f_{IL}/128$
In 8-bit interval timer mode, TRTn0 counts using the count source set in TCKn0. In 16-bit interval timer mode, TRTn counts using the count source set in TCKn0. Refer to **12.4 Operation** for details.			

Notes 1. Do not switch the count source during count operation. When switching the count source, set these bits while the TSTARTni bit in the TRTCRn register is 0 (count stops).

2. Set TCKni of the unused channel to 000B.

3. Be sure to set the TCKni (i = 0, 1) bit before setting the TRTCMPni register.

4. Bits 7 and 3 are read-only. When writing, write 0. When reading, 0 is read.

12.4 Operation

12.4.1 Count mode

The following two modes are supported: 8-bit counter mode and 16-bit counter mode. **Table 12-3** lists the registers and settings used in 8-bit counter mode and **Table 12-4** lists the registers and settings used in 16-bit counter mode.

Table 12-3. Registers and Settings Used in 8-Bit Counter Mode

Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n0 (TRTn0)	b7 to b0	8-bit counter of channel 0. The count value can be read.
8-bit interval timer counter register n1 (TRTn1)	b7 to b0	8-bit counter of channel 1. The count value can be read.
8-bit interval timer compare register n0 (TRTCMPn0)	b7 to b0	8-bit compare value of channel 0. Set the compare value.
8-bit interval timer compare register n1 (TRTCMPn1)	b7 to b0	8-bit compare value of channel 1. Set the compare value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Select whether to start/stop the count of channel 0.
	TSTARTn1	Select whether to start/stop the count of channel 1.
	TCLKENn	Set to 1.
	TCSMDn	Set to 0.
8-bit interval timer division register n (TRTMDn)	TCKn0	Select the count clock of channel 0.
	TCKn1	Select the count clock of channel 1.

Remark n = 0 to 3

Table 12-4. Registers and Settings Used in 16-Bit Counter Mode

Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n (TRTn)	b15 to b0	16-bit counter. The count value can be read.
8-bit interval timer compare register n (TRTCMPn)	b15 to b0	16-bit compare value. Set the compare value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Select whether to control starting/stopping the count.
	TSTARTn1	Set to 0.
	TCLKENn	Set to 1.
	TCSMDn	Set to 1.
8-bit interval timer division register n (TRTMDn)	TCKn0	Select the count clock.
	TCKn1	Set to 000B.

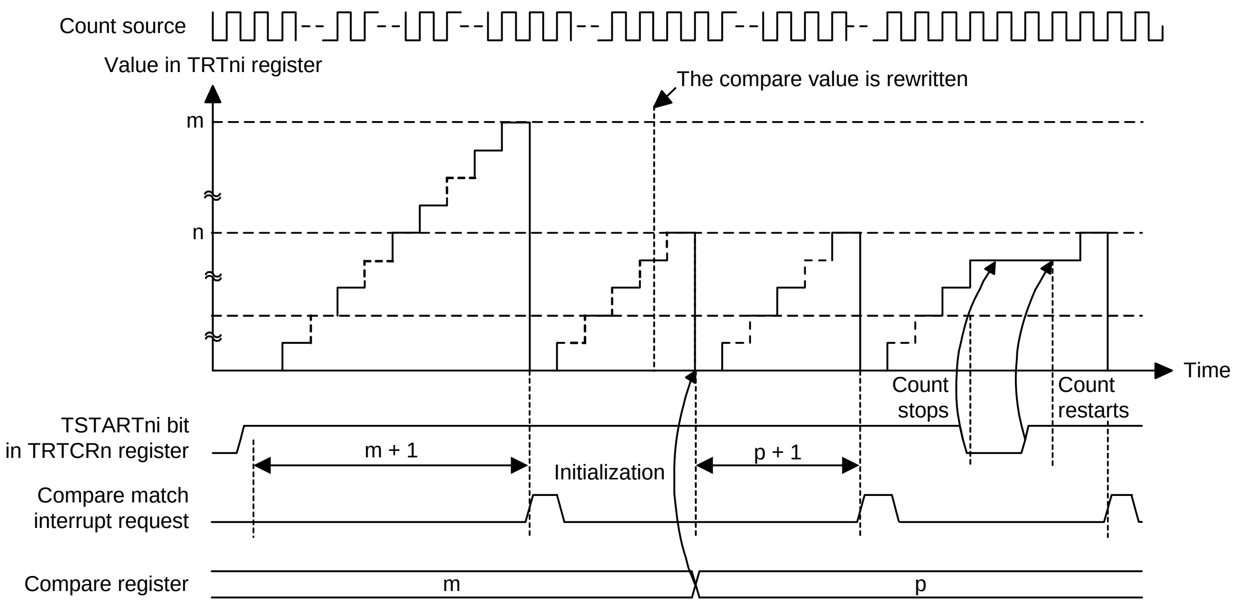
Remark n = 0 to 3

12.4.2 Timer operation

The counter is incremented by the count source selected by the TCKni (n = 0 to 3, i = 0, 1) bit in the division register (TRTMDn). The count value is decremented each time the count source is input. After the count value is set to the compare value, the value is compared and matched when the next count source is input, and then an interrupt is generated. The interrupt request is output with a single pulse that is synchronized with the count source. Note that the interrupt request continues to be generated when the TSTARTni bit in the TRTCRn register is set to 0 and counting is stopped at 00h.

When operation is stopped, the counter continues retaining the count value immediately before operation is stopped. To clear the count value, set the compare value in the TRTCMPni register again. After the TRTCMPni register is written, the count value is cleared after two cycles of the count source.

Figure 12-8. Example of Timer Operation



Remark n = 0 to 3 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 TRTCR register.

- When the count source (f_{SX} or f_{IL}) is selected
  Maximum: Two cycles of the count source
  Minimum: One cycle of the count source
- When the count source ($f_{SX}/2^m$ or $f_{IL}/2^m$) is selected
  Maximum: One cycle of the count source
  Minimum: One cycle of the selected clock (f_{SX} or f_{IL})

When the count value matches the compare value, the count value is cleared by the next count source. When the compare value in the TRTCMPni register is rewritten, the count value is also cleared two cycles of the count source after writing.

Table 12-5 lists the interrupt sources in 8-bit/16-bit count mode.

Table 12-5. Interrupt Sources in 8-Bit/16-Bit Count Mode

Interrupt Name	8-Bit Count Mode Source	16-Bit Count Mode Source
INTITn0	Rising edge of the next count source after compare match of channel 0	Rising edge of the next count source after compare match
INTITn1	Rising edge of the next count source after compare match of channel 1	Not generated

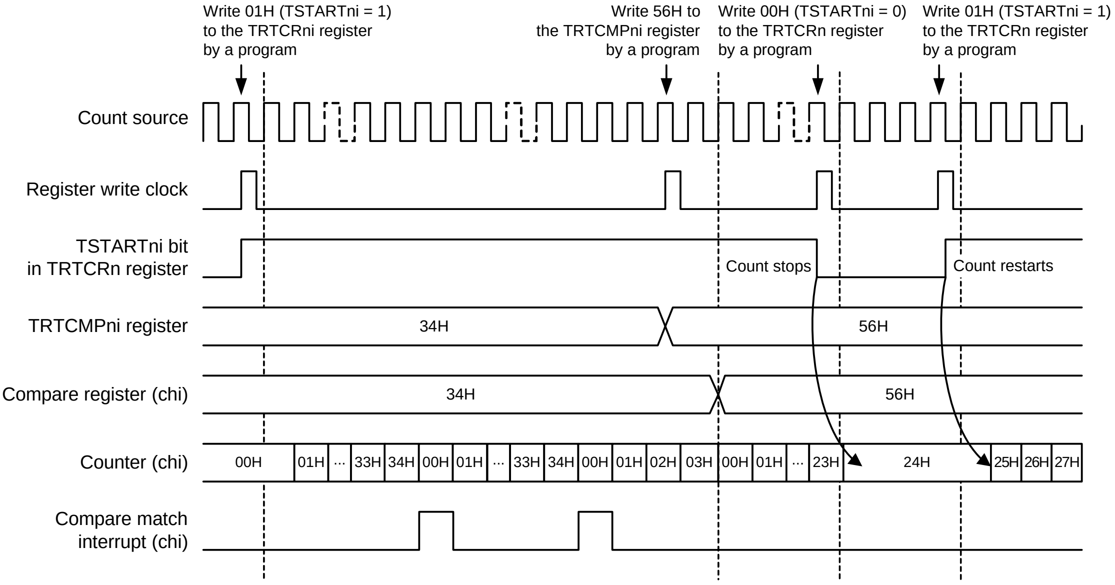
Remark n = 0 to 3

12.4.3 Start/stop timing

12.4.3.1 When count source (fSX) is selected

After 1 is written to the TSTARTni (n = 0 to 3, i = 0, 1) bit in the TRTCRn register, the count is started by the next sub clock (fSX), and then the counter is incremented from 00H to 01H by the next count source (fSX). Likewise, after 0 is written to the TSTARTni bit, the count is stopped after the counter is incremented by the sub clock (fSX). **Figure 12-9** shows the timing for starting/stopping count operation, and **Figure 12-10** shows the timing of count stop → compare setting (count clearing) → count start. **Figure 12-9** and **Figure 12-10** show the update timing in 8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

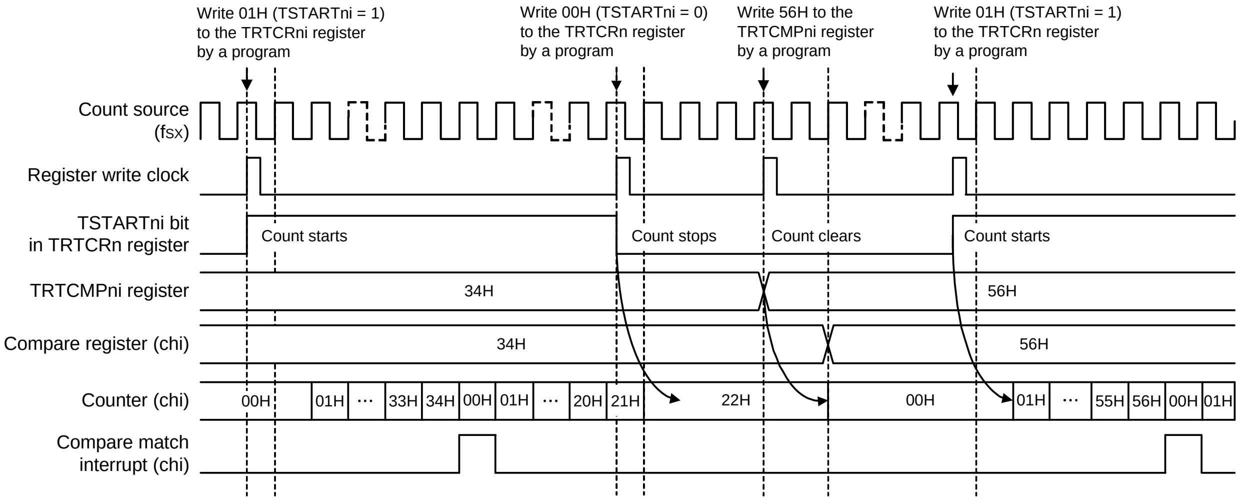
Figure 12-9. Example of Count Start/Stop Operation (fSX Selected)


Write 01H (TSTARTni = 1) to the TRTCRni register by a program
Write 56H to the TRTCMPni register by a program
Write 00H (TSTARTni = 0) to the TRTCRn register by a program
Write 01H (TSTARTni = 1) to the TRTCRn register by a program
Count source
Register write clock
TSTARTni bit in TRTCRn register
Count stops
Count restarts
TRTCMPni register
34H
56H
Compare register (chi)
34H
56H
Counter (chi)
00H
01H
...
33H
34H
00H
01H
...
33H
34H
00H
01H
02H
03H
00H
01H
...
23H
24H
25H
26H
27H
Compare match interrupt (chi)

The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0 to 3 i = 0, 1

Figure 12-10. Example of Count Stop → Count Clearing → Count Start Operation (fSX Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0 to 3 i = 0, 1

12.4.3.2 When count source (fSX/2^m) is selected

After 1 is written to the TSTARTni (n = 0 to 3, i = 0, 1) bit in the TRTCRn register, the count is started with the next sub clock (fSX), and then the counter is incremented from 00H to 01H by the next count source (fSX/2^m). Likewise, after 0 is written to the TSTARTni bit, the count is stopped with the sub clock (fSX).

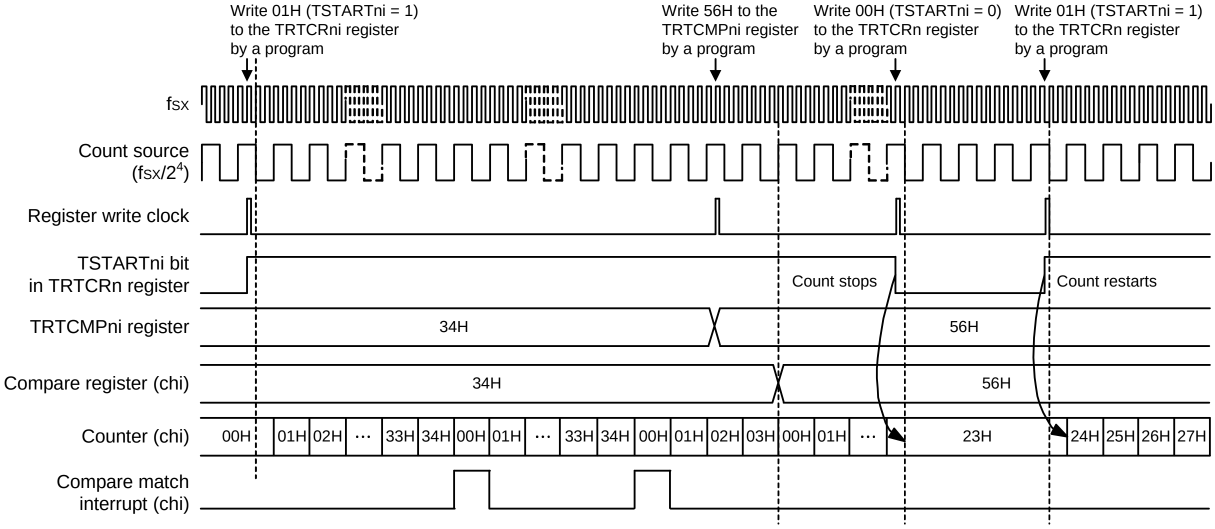
However, the first period to count 00H when the timer starts counting is shorter than one cycle of the count source as below, depending on the timing for writing to the TSTARTni bit and the timing of the next count source.

Minimum: One cycle of the sub clock (fSX)

Maximum: One cycle of the count source

Figure 12-11 shows the timing for starting/stopping count operation, and **Figure 12-12** shows the timing of count stop → compare setting (count clearing) → count start. **Figure 12-11** and **Figure 12-12** show the update timing in 8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

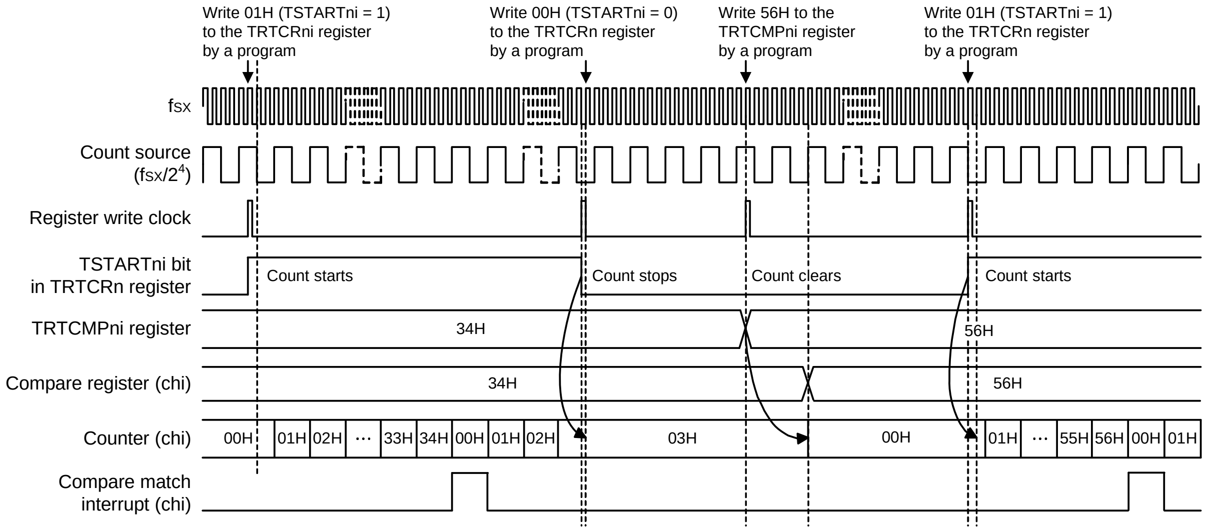
Figure 12-11. Example of Count Start/Stop Operation (fSX/2^m Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0 to 3 i = 0, 1

Figure 12-12. Example of Count Stop → Count Clearing → Count Start Operation ($f_{SX}/2^m$ Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

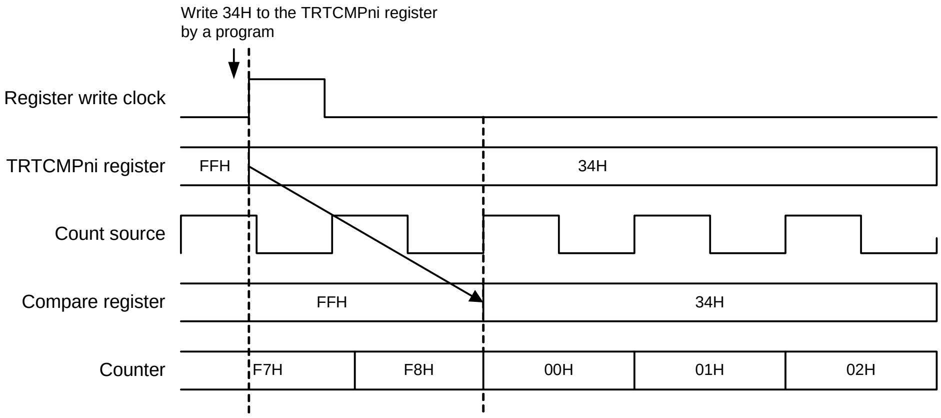
Remark n = 0 to 3 i = 0, 1

12.4.4 Timing for updating compare register values

The timing for updating the value of the TRTCMPni (n = 0 to 3, i = 0, 1) register is the same, regardless of the value of the TSTARTni bit in the TRTCRn register. After TRTCMPni is write-accessed, the value is stored in the compare register after two cycles of the count source. When stored in the compare register, the count value is cleared and set (8-bit count mode: 00H, 16-bit count mode: 0000H).

Figure 12-13 shows the timing of rewrite operation. This figure shows the update timing in 8-bit count mode, but operation is performed at the same timing in 16-bit count mode.

Figure 12-13. Timing of Compare Value Rewrite Operation



Remark n = 0 to 3 i = 0, 1

12.5 Notes on 8-Bit Interval Timer

12.5.1 Changing settings of operating mode

The settings of bits TCSMDn and TCKni (n = 0 to 3, i = 0, 1) must be changed while the TSTARTni bit in the TRTCRn register is 0 (count stops). After the value of the TSTARTni bit is rewritten from 1 to 0 (count stops), allow at least one cycle of f_{SX} or f_{IL} to elapse before accessing the registers (TRTCRn and TRTMDn) associated with the 8-bit interval timer.

12.5.2 Accessing compare registers

Do not write to the same compare registers (TRTCMPn0, TRTCMPn1, and TRTCMPn) successively. When writing successively, allow at least two cycles of the count source between writes.

Writing to the compare register (TRTCMPn0, TRTCMPn1, 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.

12.5.3 8-bit interval timer setting procedure

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 TSTARTni to 0 and then allow at least one cycle of f_{SX} or f_{IL} to elapse before setting the TCLKENn bit to 0.

CHAPTER 13 TIMER RJ

13.1 Functions of Timer RJn

Timer RJn is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. The RL78/I1C (512 KB) has two timer RJ units.

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 13-1 lists the timer RJn specifications.

Table 13-1. Timer RJn Specifications

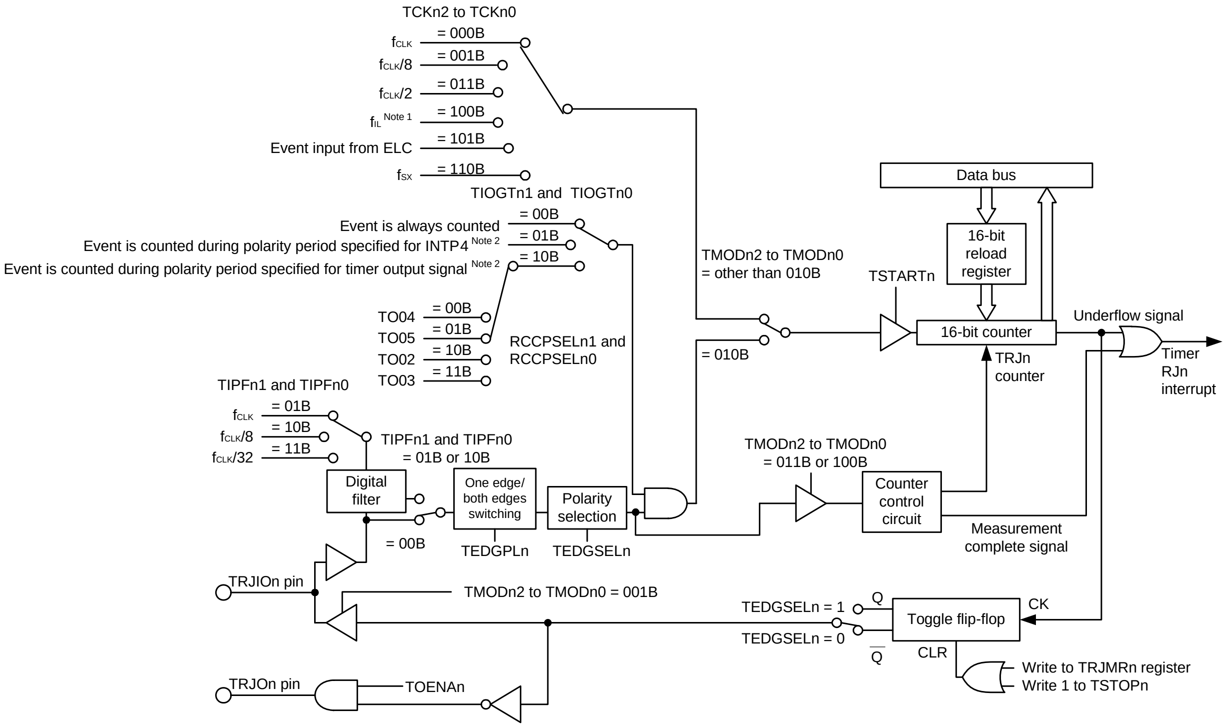
Item		Description
Operating modes	Timer mode	The count source is counted.
	Pulse output mode	The count source is counted and the output is inverted at each underflow of the timer.
	Event counter mode	An external event is counted. Operation is possible in STOP mode.
	Pulse width measurement mode	An external pulse width is measured.
	Pulse period measurement mode	An external pulse period is measured.
Count source (Operating clock)		f_{CLK}, $f_{CLK}/2$, $f_{CLK}/8$, f_{IL}, f_{SX}, or event input from the event link controller (ELC) selectable
Interrupt		• When the counter underflows. • When the measurement of the active width of the external input (TRJIOn) is completed in pulse width measurement mode. • When the set edge of the external input (TRJIOn) is input in pulse period measurement mode.
Selectable functions		• Coordination with the event link controller (ELC). Event input from the ELC is selectable as a count source.

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

13.2 Configuration of Timer RJn

Figure 13-1 shows the timer RJn block diagram and **Table 13-2** lists the timer RJn pin configuration.

Figure 13-1. Timer RJn Block Diagram



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

Notes 1. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.

2. The polarity can be selected by the RCCPSELn2 bit in the TRJISRn register.

Table 13-2. Timer RJn Pin Configuration

Pin Name	I/O	Function
INTP4	Input	Event counter mode control for timer RJn
TRJIOn	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)

13.3 Registers Controlling Timer RJn

Table 13-3 lists the registers controlling timer RJn.

Table 13-3. Registers Controlling Timer RJn

Register Name	Symbol
Peripheral enable register 1	PER1
Peripheral reset control register 1	PRR1
Subsystem clock supply mode control register	OSMC
Timer RJ counter register n Note	TRJn
Timer RJ control register n	TRJCRn
Timer RJ I/O control register n	TRJIOCn
Timer RJ mode register n	TRJMRn
Timer RJ event pin select register n	TRJISRn
Port register 0	P0
Port register 5	P5
Port register 6	P6
Port register 8	P8
Port mode register 0	PM0
Port mode register 5	PM5
Port mode register 6	PM6
Port mode register 8	PM8

Note When the TRJn register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJn register is one clock for both writing and reading.

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

13.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the timer RJ0 is to be used, be sure to set bit 1 (TRJ0EN) to 1.

When the timer RJ1 is to be used, be sure to set bit 2 (TRJ1EN) to 1.

The PER1 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 13-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

TRJnEN	Control of timer RJn input clock supply
0	Stops input clock supply. • SFR used by the timer RJn cannot be written. The read value is 00H. However, the SFR is not initialized. Notes 1, 2
1	Enables input clock supply. • SFR used by the timer RJn can be read and written.

Notes 1. To initialize the timer RJ0 and the SFR used by the timer RJ0, use bit 1 (TRJ0RES) of PRR1.

2. To initialize the timer RJ1 and the SFR used by the timer RJ1, use bit 2 (TRJ1RES) of PRR1.

Cautions 1. When using the timer RJn, be sure to set the following registers first while the TRJnEN bit is set to 1. If TRJnEN = 0, writing to the registers controlling the timer RJn is ignored, and, even if the register is read, only the default value is read (except for the port mode registers 0, 5, 6, 8 (PM0, PM5, PM6, PM8), port registers 0, 5, 6, 8 (P0, P5, P6, P8)).

- **Timer RJ counter register n (TRJn)**
- **Timer RJ control register n (TRJCRn)**
- **Timer RJ I/O control register n (TRJIOCn)**
- **Timer RJ mode register n (TRJMRn)**
- **Timer RJ event pin select register n (TRJISRn)**

2. Be sure to clear bit 7 to 0.

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

13.3.2 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.

To place the timer RJ0 in the reset state, be sure to set bit 1 (TRJ0RES) to 1.

To place the timer RJ1 in the reset state, be sure to set bit 2 (TRJ1RES) to 1.

The PRR1 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 13-3. Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	<2>	<1>	<0>
PRR1	0	0	SMOTD1RES	SMOTD0RES	0	TRJ1RES	TRJ0RES	DSADRES

TRJnRES	Control resetting of the timer RJn
0	The timer RJn is released from the reset state.
1	The timer RJn is in the reset state.

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

13.3.3 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the timer RJn operation clock.

In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 6 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 13-4. Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL}) Notes 2, 3, 5, 6	Low-speed on-chip oscillator clock (f_{IL}) selected[Note 5]	Clock output is prohibited. [Note 4]	Setting prohibited

Notes 1. Be sure to clear bits 6, 5, and 3 to 0 to 0.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (f_{IL}) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

5. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

6. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

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

13.3.4 Timer RJ counter register n (TRJn)

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 are changed depending on the TSTARTn bit in the TRJCRn register. For details, see **13.4.1 Reload Register and Counter Rewrite Operation**.

The TRJn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to FFFFH.

Figure 13-5. Format of Timer RJ Counter Register n (TRJn)

Address: F0560H (TRJ0), F0562H (TRJ1) After reset: FFFFH 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

Notes 1. When 1 is written to the TSTOPn bit in the TRJCRn register, the 16-bit counter is forcibly stopped and set to FFFFH.

2. When the setting of bits TCKn2 to TCKn0 in the TRJMRn register is other than 001B (f_{CLK}/8) or 011B (f_{CLK}/2), if the TRJn register is set to 0000H, a request signal to the DTC and the ELC is generated only once immediately after the count 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 ELC is generated only once immediately after the count 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 0000H or a higher value, a request signal is generated each time TRJn underflows.

Caution When the TRJn register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJn register is one clock for both writing and reading.

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

13.3.5 Timer RJ control register n (TRJCRn)

The TRJCRn register starts or stops count operation and indicates the status of timer RJn.

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

Reset signal generation clears this register to 00H.

Figure 13-6. Format of Timer RJ Control Register n (TRJCRn) (1/2)

Address: F02A0H (TRJCR0), F02A4H (TRJCR1) After reset: 00H 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. [Conditions 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 count forced stop[Note 1]
When 1 is written to this bit, the count is forcibly stopped. The read value is 0.	

TCSTFn	Timer RJn count status flag[Note 2]
0	Count stops
1	Count in progress
[Condition 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. [Conditions 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).	

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

Figure 13-6. Format of Timer RJ Control Register n (TRJCRn) (2/2)

Address: F02A0H (TRJCR0), F02A4H (TRJCR1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJCRn	0	0	TUNDFn	TEDGFn	0	TSTOPn	TCSTFn	TSTARTn

TSTARTn	Timer RJn count start[Note 2]
0	Count stops
1	Count starts
Count operation is started by writing 1 to the TSTARTn bit and stopped by writing 0. When the TSTARTn bit is set to 1 (count starts), the TCSTFn bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTARTn bit, the TCSTFn bit is set to 0 (count stops) in synchronization with the count source. For details, see **13.5.1 Count Operation Start and Stop Control**.	

Notes 1. When 1 (count is forcibly stopped) is written to the TSTOPn bit, bits TSTARTn and TCSTFn are initialized at the same time. The pulse output level is also initialized.

2. For notes on using bits TSTARTn and TCSTFn, see **13.5.1 Count Operation Start and Stop Control**.

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

13.3.6 Timer RJ I/O control register n (TRJIOCn)

The TRJIOCn register sets the input/output of timer RJn.

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

Reset signal generation clears this register to 00H.

Figure 13-7. Format of Timer RJ I/O Control Register n (TRJIOCn)

Address: F02A1H (TRJIOC0), F02A5H (TRJIOC1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJIOCn	TIOGTn1	TIOGTn0	TIPFn1	TIPFn0	0	TOENAn	0	TEDGSELn

TIOGTn1	TIOGTn0	TRJIOn count 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	TRJIOn input filter select
0	0	No filter
0	1	Filter sampled at f_{CLK}
1	0	Filter sampled at f_{CLK}/8
1	1	Filter sampled at f_{CLK}/32
These bits are used to specify the sampling frequency of the filter for the TRJIOn input. If the input to the TRJIOn 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 13-4** and **13-5**).	

Notes 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.

2. Bits TIOGTn0 and TIOGTn1 are enabled only in event counter mode.

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

Table 13-4. TRJIOn I/O Edge and Polarity Switching

Operating Mode	Function
Timer mode	Not used (I/O port)
Pulse output mode	0: Output is started at high (Initialization level: High) 1: Output is started at low (Initialization level: Low)
Event counter mode	0: Count 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 13-5. TRJOn Output Polarity Switching

Operating Mode	Function
All modes	0: Output is started at low (Initialization level: Low) 1: Output is started at high (Initialization level: High)

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

13.3.7 Timer RJ mode register n (TRJMRn)

The TRJMRn register sets the operating mode of timer RJn.

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

Reset signal generation clears this register to 00H.

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

Figure 13-8. Format of Timer RJ mode register n (TRJMRn)

Address: F02A2H (TRJMR0), F02A6H (TRJMR1) After reset: 00H 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	f_{CLK}
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	f_{IL} Note 3
1	0	1	Event input from ELC
1	1	0	f_{SX}
Other than above			Setting prohibited

TEDGPLn	TRJIOn 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

Notes 1. When event counter mode is selected, the external input (TRJIOn) is selected as the count source regardless of the setting of bits TCKn0 to TCKn2.

2. Do not switch count sources during count operation. Count sources should be switched when both the TSTARTn and TCSTFn bits in the TRJCRn register are set to 0 (count stops).

3. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.

4. The TEDGPLn bit is enabled only in event counter mode.

5. The operating mode can be changed only when the count is stopped while both the bits TSTARTn and TCSTFn in the TRJCRn register are set to 0 (count stops). Do not change the operating mode during count operation.

Caution **Write access to the TRJMRn register initializes the output from pins TRJOn and TRJIOn of timer RJn. For details on the output level at initialization, refer to the description of Figure 13-7 Format of Timer RJ I/O control register n (TRJIOCn).**

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

13.3.8 Timer RJ event pin select register n (TRJISRn)

The TRJISRn register selects the timer for controlling the event count period and sets the polarity in event counter mode.

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

Reset signal generation clears this register to 00H.

Figure 13-9. Format of Timer RJ event pin select register n (TRJISRn)

Address: F02A3H (TRJISR0), F02A7H (TRJISR1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJISRn	0	0	0	0	0	RCCPSEn2	RCCPSEn1	RCCPSEn0

RCCPSEn2[Note]	Timer output signal and INTP4 polarity selection
0	An event is counted during the low-level period
1	An event is counted during the high-level period

RCCPSEn1[Note]	RCCPSEn0[Note]	Timer output signal selection
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)

13.3.9 Port mode registers 0, 5, 6, 8 (PM0, PM5, PM6, PM8)

These registers set input/output of ports 0, 5, 6, and 8 in 1-bit units.

When using the ports (P57/SEG39/SCK30/SCL30/TRJIO0, P53/SEG35/TRJO0/RxDMG1, etc.) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P57/SEG39/SCK30/SCL30/TRJIO0 for timer output
Set the PM57 bit of port mode register 5 to 0.
Set the P57 bit of port register 5 to 0.
Set the PFSEG39 bit of LCD port function register 5 to 0.

When using the ports (P57/SEG39/SCK30/SCL30/TRJIO0, etc.) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P57/SEG39/SCK30/SCL30/TRJIO0 for timer input
Set the PM57 bit of port mode register 5 to 1.
Set the P57 bit of port register 5 to 0 or 1.
Set the PFSEG39 bit of LCD port function register 5 to 0.

The PM0, PM5, PM6, and PM8 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark When using the ports to be shared with the segment output pin for timer I/O, be sure to clear the corresponding bit of LCD port function registers 5 and 6 (PFSEG5 and PFSEG6) to “0”.

Figure 13-10. Format of Port Mode Registers 0, 5, 6, 8 (PM0, PM5, PM6, PM8)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	1	1

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	PM62	PM61	PM60

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	PM85	PM84	PM83	PM82	PM81	PM80

PMmn	Selection of I/O mode for Pmn pin (m = 0, 5, 6, 8; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

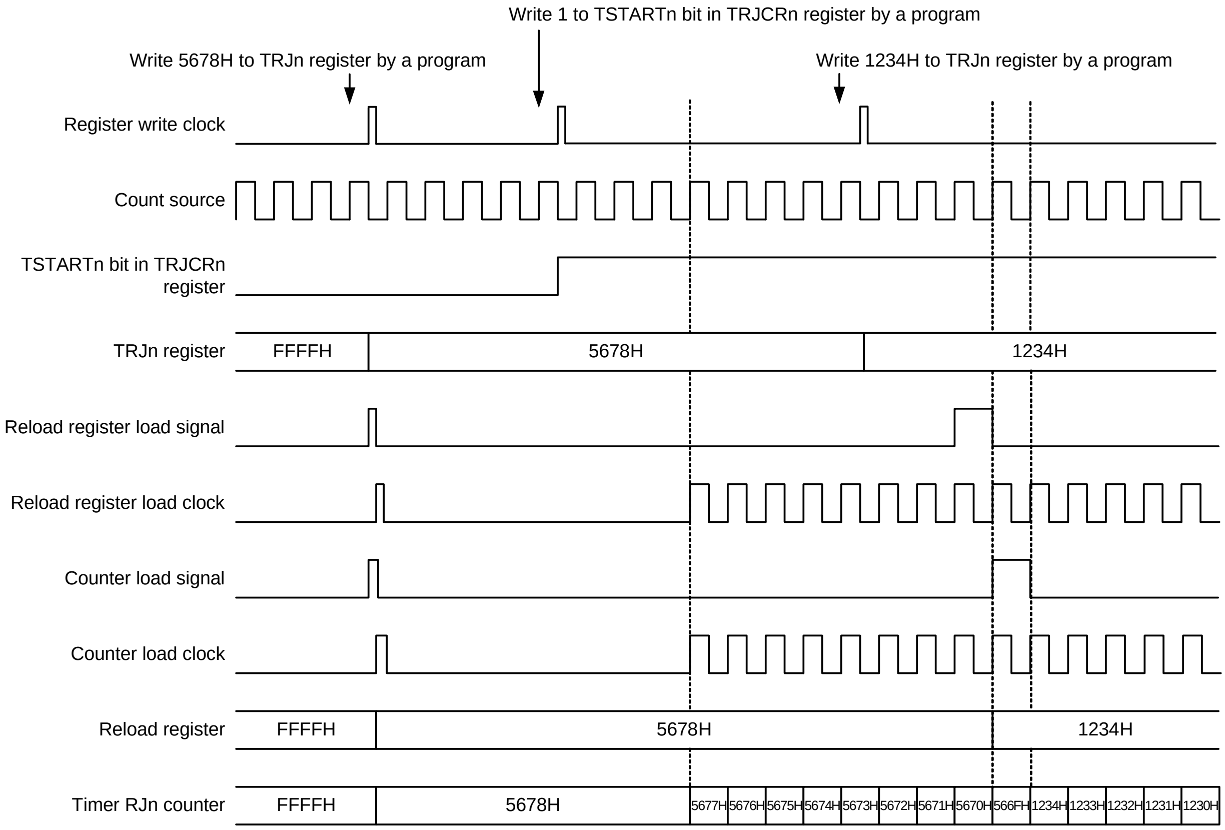
13.4 Timer RJn Operation

13.4.1 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 (count stops), the count value is directly written to the reload register and the counter. When the TSTARTn bit is 1 (count starts), 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 13-11 shows the timing of rewrite operation with TSTARTn bit value.

Figure 13-11. Timing of Rewrite Operation with TSTARTn Bit Value



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

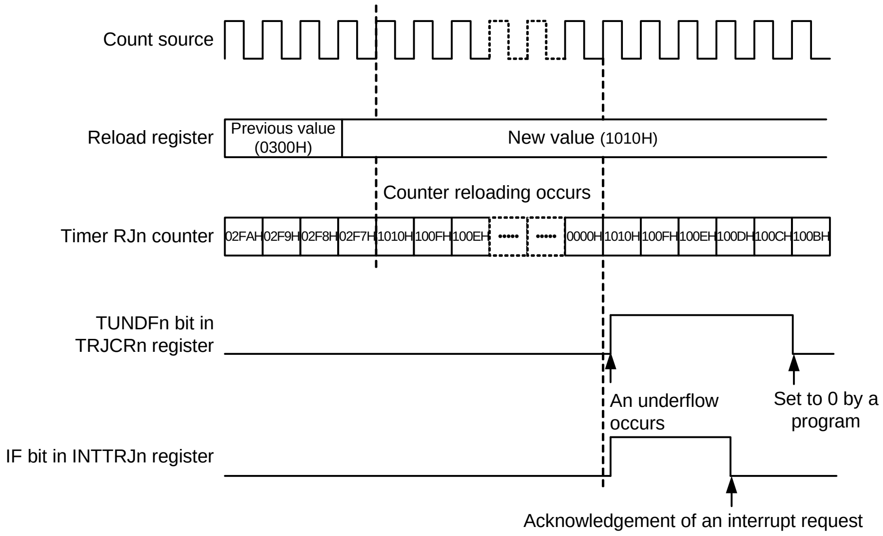
13.4.2 Timer Mode

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 13-12 shows the operation example in timer mode.

Figure 13-12. Operation Example in Timer Mode



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

13.4.3 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 TRJIOn and TRJOn pin is inverted each time an underflow occurs.

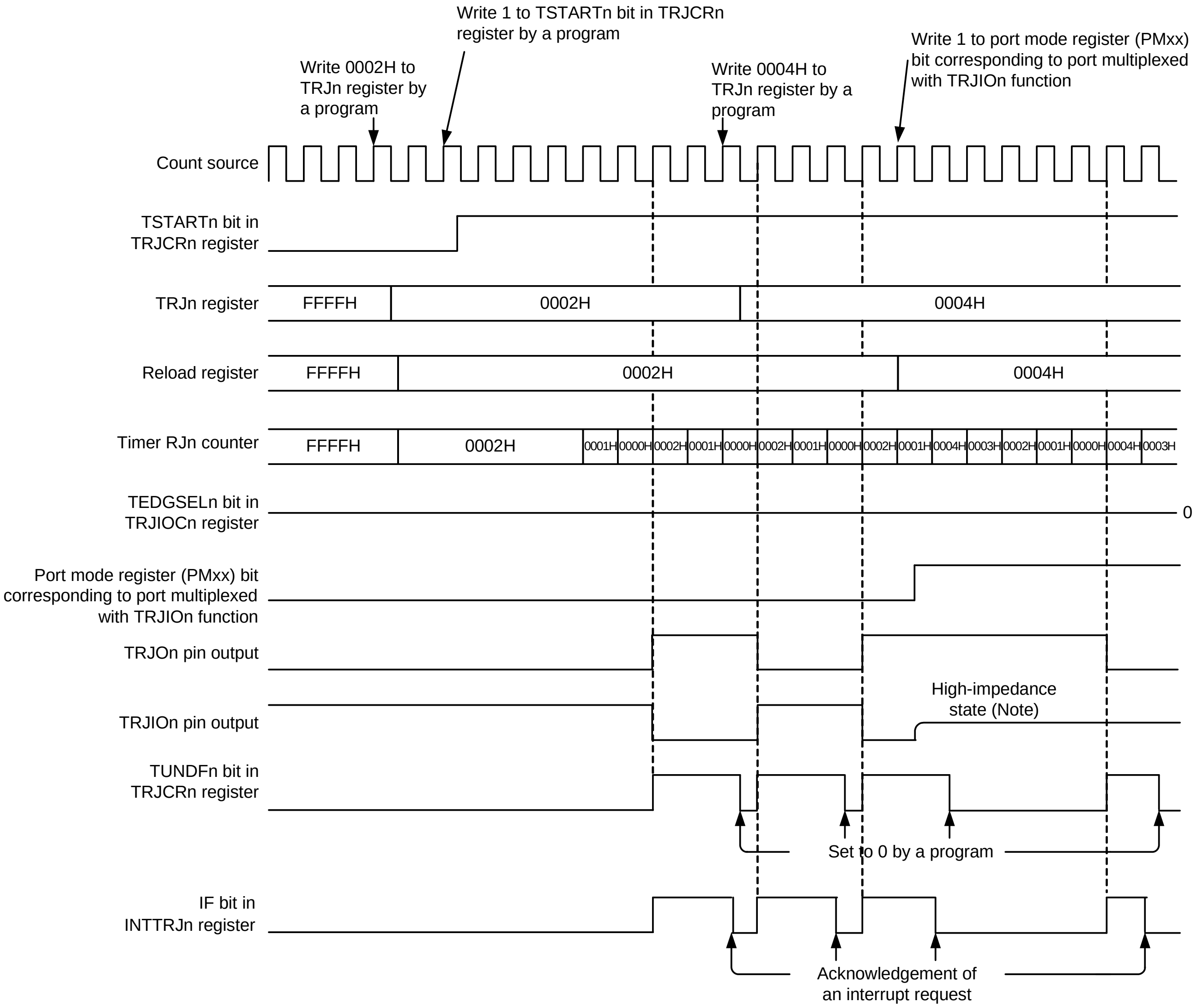
In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

In addition, a pulse can be output from pins TRJIOn 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 13-13 shows the operation example in pulse output mode.

Figure 13-13. Operation Example in Pulse Output Mode


Write 1 to TSTARTn bit in TRJCRn register by a program
Write 0002H to TRJn register by a program
Write 0004H to TRJn register by a program
Write 1 to port mode register (PMxx) bit corresponding to port multiplexed with TRJIOn function
Count source
TSTARTn bit in TRJCRn register
TRJn register
FFFFH
0002H
0004H
Reload register
FFFFH
0002H
0004H
Timer RJn counter
FFFFH
0002H
0001H
0000H
0002H
0001H
0000H
0002H
0001H
0000H
0002H
0001H
0004H
0003H
0002H
0001H
0000H
0004H
0003H
TEDGSELn bit in TRJIOCn register
0
Port mode register (PMxx) bit corresponding to port multiplexed with TRJIOn function
TRJOn pin output
TRJIOn pin output
High-impedance state (Note)
TUNDFn bit in TRJCRn register
Set to 0 by a program
IF bit in INTTRJn register
Acknowledgement of an interrupt request

Note: The TRJIOn pin becomes high impedance by output enable control on the port selected as the TRJIOn function.

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

13.4.4 Event Counter Mode

In this mode, the counter is decremented by an external event signal (count source) input to the TRJIOn pin.

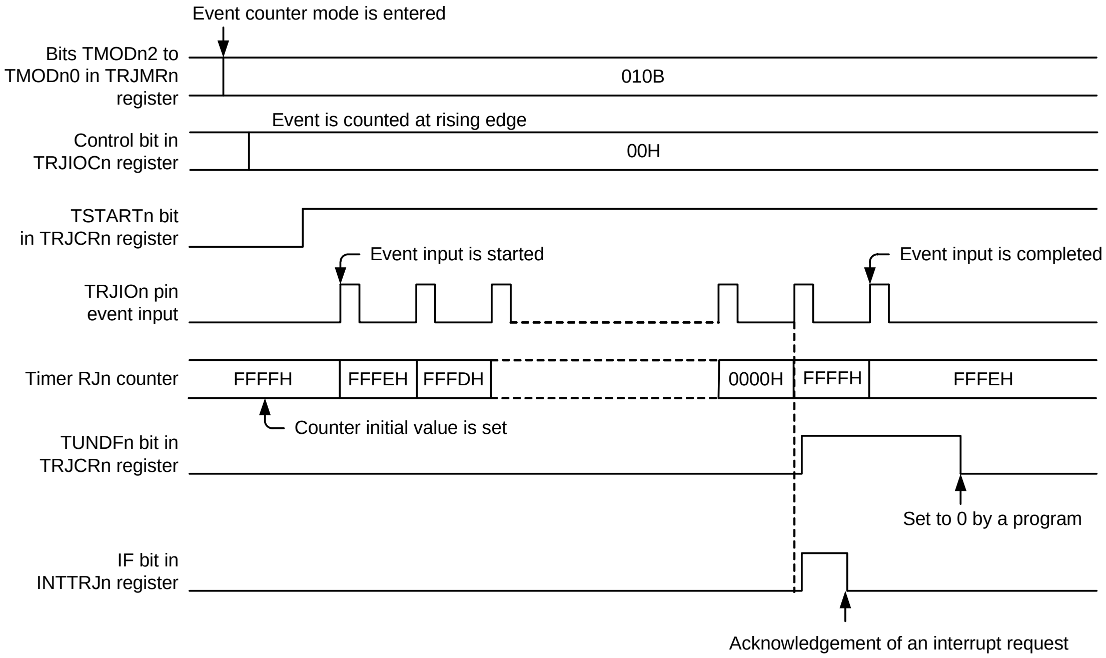
Various periods for counting events can be set by bits TIOGTn0 and TIOGTn1 in the TRJIOCn register and the TRJISRn register. In addition, the filter function for the TRJIOn input can be specified by bits TIPFn0 and TIPFn1 in the TRJIOCn register.

Also, the output from the TRJOn pin can be toggled even in event counter mode.

When event counter mode is used, see **13.5.5 Procedure for Setting Pins TRJOn and TRJIOn**.

Figure 13-14 shows the operation example 1 in event counter mode.

Figure 13-14. Operation Example 1 in Event Counter Mode

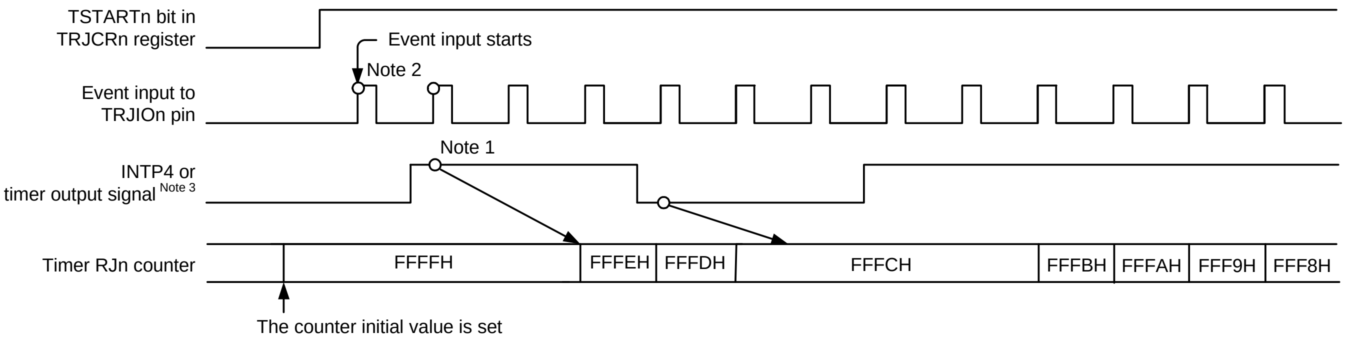

Event counter mode is entered
Bits TMODn2 to TMODn0 in TRJMRn register
010B
Event is counted at rising edge
Control bit in TRJIOCn register
00H
TSTARTn bit in TRJCRn register
Event input is started
Event input is completed
TRJIOn pin event input
Timer RJn counter
FFFFH
FFFEH
FFFDH
0000H
FFFFH
FFFEH
Counter initial value is set
TUNDFn bit in TRJCRn register
Set to 0 by a program
IF bit in INTTRJn register
Acknowledgement of an interrupt request

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

Figure 13-15 shows an operation example for counting during the specified period in event counter mode (bits TIOGTn1 and TIOGTn0 in the TRJIOn0 register are set to 01B or 10B).

Figure 13-15. Operation Example 2 in Event Counter Mode

Timing example when the setting of operating mode is as follows:
TRJMRn register: TMODn2, TMODn1, TMODn0 = 010B (event counter mode)
TRJIOCn register: TIOGTn1, TIOGTn0 = 01B (event is counted during specified period for external interrupt pin)
TIPFn1, TIPFn0 = 00B (no filter)
TEDGSELn = 0 (count at rising edge)
TRJISRn register: RCCPSELn2 = 1 (high-level period is counted)



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 count mode.

Notes 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.

2. Count operation may be performed for two cycles of the count source immediately after the count is started, depending on the previous state before the count is stopped.
To disable the count for two cycles immediately after the count is started, write 1 to the TSTOPn bit in the TRJCRn register to initialize the internal circuit, and then make operation settings before starting count operation.

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)

13.4.5 Pulse Width Measurement Mode

In this mode, the pulse width of an external signal input to the TRJIOn pin is measured.

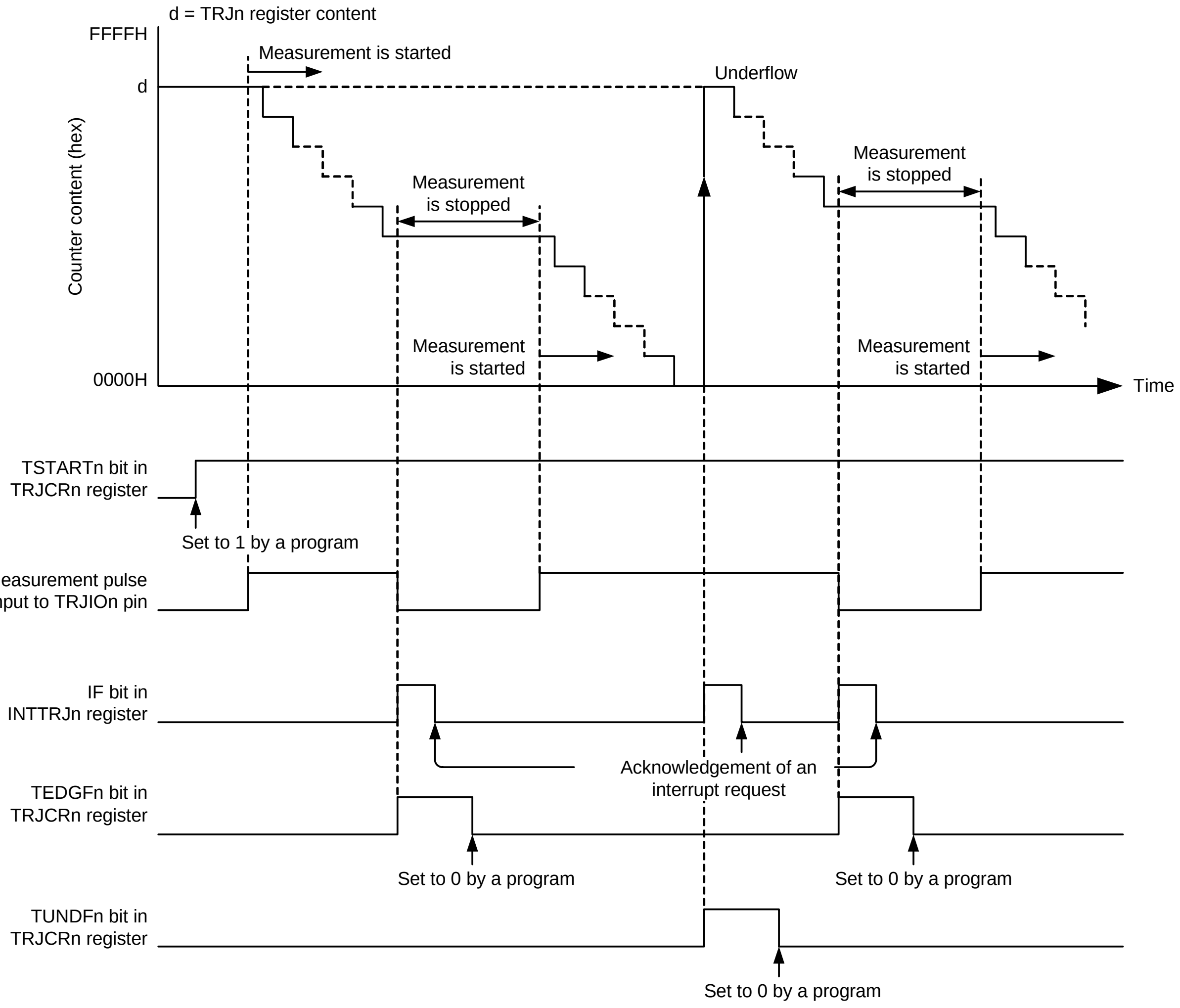
When the level specified by the TEDGSELn bit in the TRJIOCn register is input to the TRJIOn pin, the decrement is started with the selected count source. When the specified level on the TRJIOn pin ends, the counter is stopped, the TEDGFn bit in the TRJCRn register is set to 1 (active edge received), 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 13-16 shows the operation example in pulse width measurement mode.

When accessing bits TEDGFn and TUNDFn in the TRJCR0 register, see **13.5.2 Access to Flags (Bits TEDGFn and TUNDFn in TRJCRn Register)**.

Figure 13-16. Operation Example in Pulse Width Measurement Mode

This example applies when the high-level width of the measurement pulse is measured (TEDGSELn bit in TRJIOCn register = 1)



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

13.4.6 Pulse Period Measurement Mode

In this mode, the pulse period of an external signal input to the TRJIOn pin is measured.

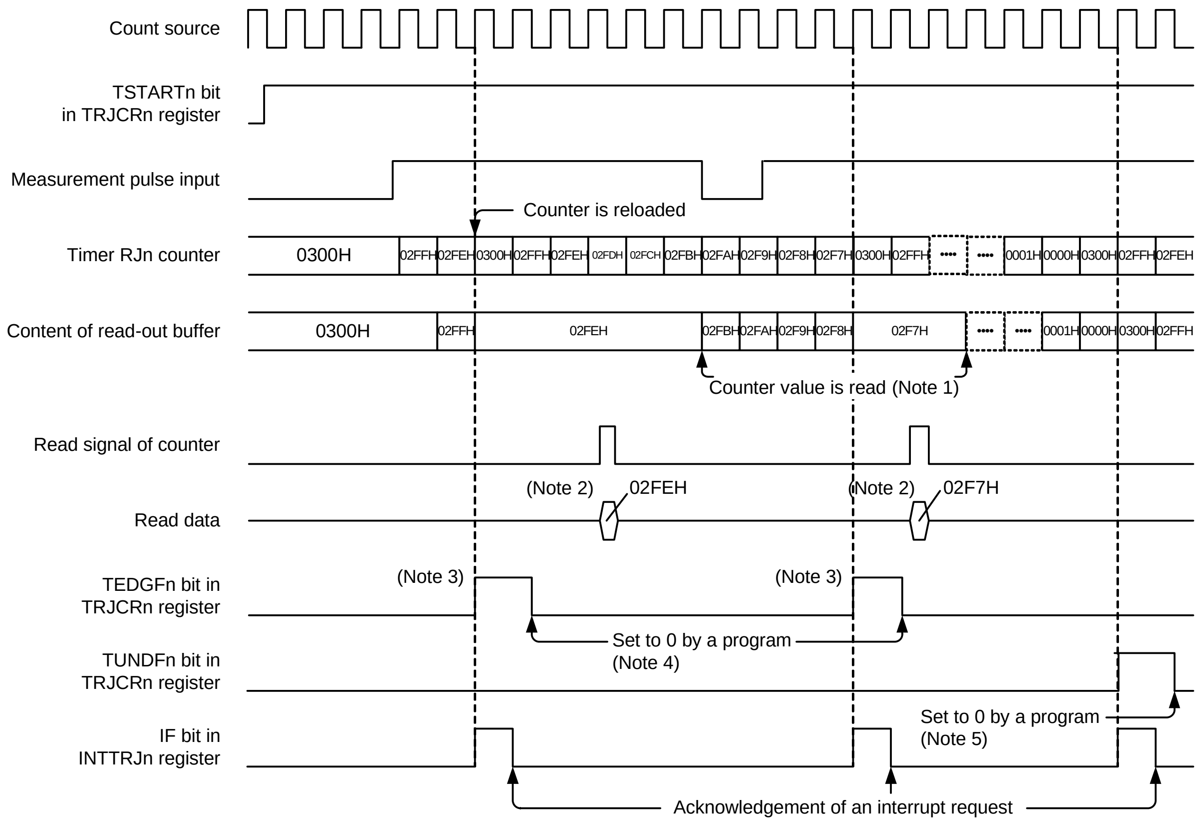
The counter is decremented by the count source selected by bits TCKn0 to TCKn2 in the TRJMRn register.

When a pulse with the period specified by the TEDGSELn bit in the TRJIOCn register is input to the TRJIO0 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 13-17 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 13-17. 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.

Notes 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.

2. When the TRJn register is read in pulse period measurement mode, the content of the read-out buffer is read.

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).

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.

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)

13.4.7 Coordination with Event Link Controller (ELC)

Through coordination with the ELC, event input from the ELC can be set to be the count source. Bits TCKn0 to TCKn2 in the TRJMRn register count at the rising edge of event input from the ELC. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

- Procedure for starting operation

(1) Set the event output destination select register (ELSELRn) for the ELC.
(2) Set the operating mode for the event generation source.
(3) Set the mode for timer RJn.
(4) Start the count operation of 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 the count operation of timer RJn.
(3) Set the event output destination select register (ELSELRn) for the ELC to 0.

Remark n: Channel number (n = 0, 1)
For ELSELRn, n = 00 to 29

13.4.8 Output Settings for Each Mode

Tables 13-6 and **13-7** list the states of pins TRJOn and TRJIOn in each mode.

Table 13-6. 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 13-7. TRJIOn Pin Setting

Operating Mode	TRJIOCn Register		TRJIOn 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 The port mode register (PMxx) bit corresponding to port multiplexed with TRJIOn function.

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

13.5 Cautions for Timer RJn

13.5.1 Count Operation Start and Stop Control

- When event count mode is set or the count source is set to other than the ELC

After 1 (count starts) is written to the TSTARTn bit in the TRJCRn register while the count is stopped, the TCSTFn bit in the TRJCRn register remains 0 (count stops) 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 (count in progress).

After 0 (count stops) is written to the TSTARTn bit during a count operation, the TCSTFn bit remains 1 for three cycles of the count source. When the TCSTFn bit is set to 0, the count is 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 TATARTn bit from 0 to 1. Refer to **CHAPTER 27 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJn: TRJn, TRJCRn, TRJIOCn, TRJMRn, and TRJISRn

- When event count mode is set or the count source is set to the ELC

After 1 (count starts) is written to the TSTARTn bit in the TRJCRn register while the count is stopped, the TCSTFn bit in the TRJCRn register remains 0 (count stops) 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 (count in progress).

After 0 (count stops) is written to the TSTARTn bit during a count operation, the TCSTFn bit remains 1 for two cycles of the CPU clock. When the TCSTFn bit is set to 0, the count is 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 TATARTn bit from 0 to 1. Refer to **CHAPTER 27 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJn: TRJn, TRJCRn, TRJIOCn, TRJMRn, and TRJISRn

13.5.2 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.

13.5.3 Access to Counter Register

When bits TSTARTn and TCSTFn in the TRJCRn register are both 1 (count starts), allow at least three cycles of the count source clock between writes when writing to the TRJn register successively.

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

13.5.4 When Changing Mode

The registers associated with timer RJn operating mode (TRJIOCn, TRJMRn, and TRJISRn) can be changed only when the count is stopped with both the TSTARTn and TCSTFn bits set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with timer RJn operating mode are changed, the values of bits TSTARTn and TCSTFn are undefined. Write 0 (no active edge received) to the TEDGFn bit and 0 (no underflow) to the TUNDFn bit before starting the count.

13.5.5 Procedure for Setting Pins TRJOn and TRJIOn

After a reset, the I/O ports multiplexed with pins TRJOn and TRJIOn function as input ports.

To output from pins TRJOn and TRJIOn, use the following setting procedure:
Changing procedure

(1) Set the mode.

(2) Set the initial value/output enabled.

(3) Set the port register bits corresponding to pins TRJOn and TRJIOn to 0.

(4) Set the port mode register bits corresponding to pins TRJOn and TRJIOn to output mode.
(Output is started from pins TRJOn and TRJIOn)

(5) Start the count (TSTARTn in TRJCRn register = 1).

To input from the TRJIOn pin, use the following setting procedure:

(1) Set the mode.

(2) Set the initial value/edge selected.

(3) Set the port mode register bit corresponding to TRJIOn pin to input mode.
(Input is started from the TRJIOn pin)

(4) Start the count (TSTARTn in TRJMRn register = 1).

(5) Wait until the TCSTFn bit in the TRJCRn register is set to 1 (count in progress).
(In event counter mode only)

(6) Input an external event from the TRJIOn 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)

13.5.6 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).

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

13.5.7 When Timer RJn Operating Clock is Stopped

Supplying or stopping the timer RJn clock can be controlled by the TRJnEN bit in the PER1 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.

13.5.8 Procedure for Setting STOP Mode (Event Counter Mode)

To perform event counter mode operation during STOP mode, first supply the timer RJn clock and then use the following procedure to enter STOP mode.

Setting procedure

(1) Set the operating mode.

(2) Start the count (TSTARTn = 1, TCSTFn = 1).

(3) Stop supplying the timer RJn clock.

To stop event counter mode operation during STOP mode, use the following procedure to stop operation.

(1) Supply the timer RJn clock.

(2) Stop the count (TSTARTn = 0, TCSTFn = 0)

13.5.9 Functional Restriction in STOP Mode (Event Counter Mode Only)

When event counter mode operation is performed during STOP mode, the digital filter function cannot be used.

13.5.10 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOPn bit in the TRJCRn register, do not access the following SFRs for one cycle of the count source.

Registers TRJn, TRJCRn, and TRJMRn

13.5.11 Digital Filter

When the digital filter is used, do not start timer operation for five cycles of the digital filter clock after setting bits TIPFn1 and TIPFn0 in the TRJIOCn.

Also, do not start timer operation for five cycles of the digital filter clock when the TEDGSELn bit in the TRJIOCn register is changed while the digital filter is used.

13.5.12 When Selecting f_{IL} as Count Source

When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.

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

CHAPTER 14 SAMPLING OUTPUT TIMER DETECTOR

14.1 Functions of Sampling Output Timer Detector

The sampling output timer detector (SMOTD) has the sampling clock output function and sampling detector function.

The sampling clock output function outputs the sampling clock signal periodically from the SMOmj pin.

The sampling detector function detects the levels being input on the SMP0 to SMP5 pins on falling edges of the sampling clock signal, and outputs the corresponding interrupt signals (INTSMPmi) if the active level is detected.

Table 14-1 describes the function of the sampling output timer detector.

Table 14-1. Function of Sampling Output Timer Detector

Item	Description	
PWM performance	High pulse width	MIN. 1/fsx MAX. $2^8 \times 2^6$/fsx[Note]
	PWM waveform period	MIN. 2/fsx MAX. $2^8 \times 2^{14}$/fsx
Functions	• Sampling clock output function • Sampling detector function	
Interrupt output	• Sampling output timer interval interrupt (INTSMOTAm) • Sampling output timer compare match interrupt (INTSMOTBm) • Sampling detector detection interrupt (INTSMPmi)	

Note Setting a longer pulse width at high level than the period of the PWM waveform is prohibited. Be sure to set a pulse width that is shorter than the period of the PWM waveform.

Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5), j: Output channel number (j = 0 to 2)

14.2 Configuration of Sampling Output Timer Detector

The sampling output timer detector includes the following hardware.

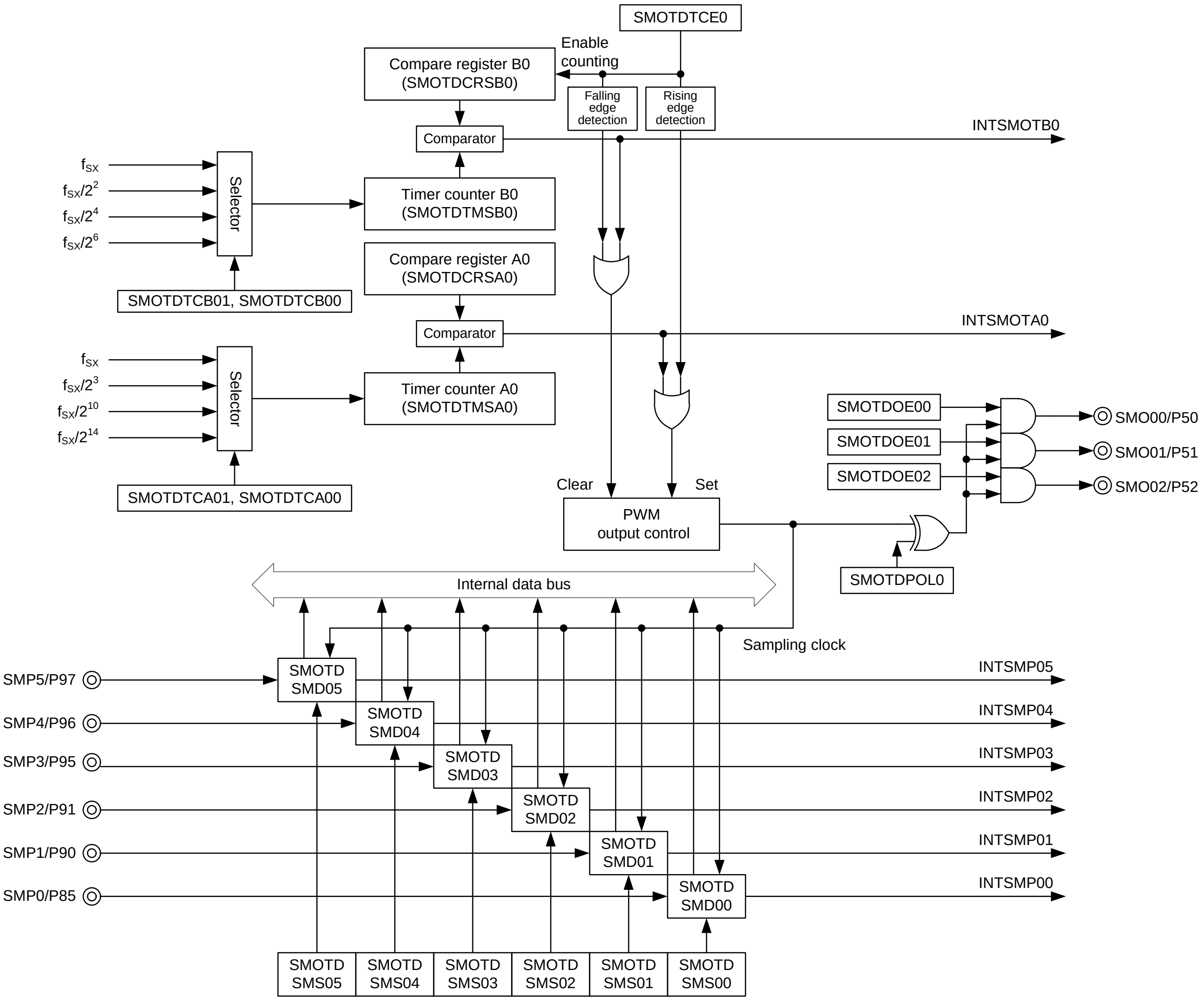
Table 14-2. Configuration of Sampling Output Timer Detector

Item	Configuration
Input clock to sampling output timer	f_{SX} or frequency-divided f_{SX}
Output from sampling output timer	Output signals from SMO00 to SMO02 and SMO10 to SMO12 (PWM output)
Sampling input	Sampling input SMP0 to SMP5
Control registers	Peripheral enable register 1 (PER1) Peripheral reset control register 1 (PRR1) SMOTD timer counter Am (SMOTDTMSAm) SMOTD timer counter Bm (SMOTDTMSBm) SMOTD compare register Am (SMOTDCRSAm) SMOTD compare register Bm (SMOTDCRSBm) SMOTD clock select register m (SMOTDTCSm) SMOTD control register m (SMOTDCRm) SMOTD sampling level setting register m (SMOTDSMSm) SMOTD sampling pin status register m (SMOTDSMDm) SMOTD output control register m (SMOTDOEm) Port mode registers 0, 3, 5, 7 to 9 (PM0, PM3, PM5, PM7 to PM9) Port registers 0, 3, 5, 7 to 9, 13 (P0, P3, P5, P7 to P9, P13)

Remark m: Unit number (m = 0, 1)

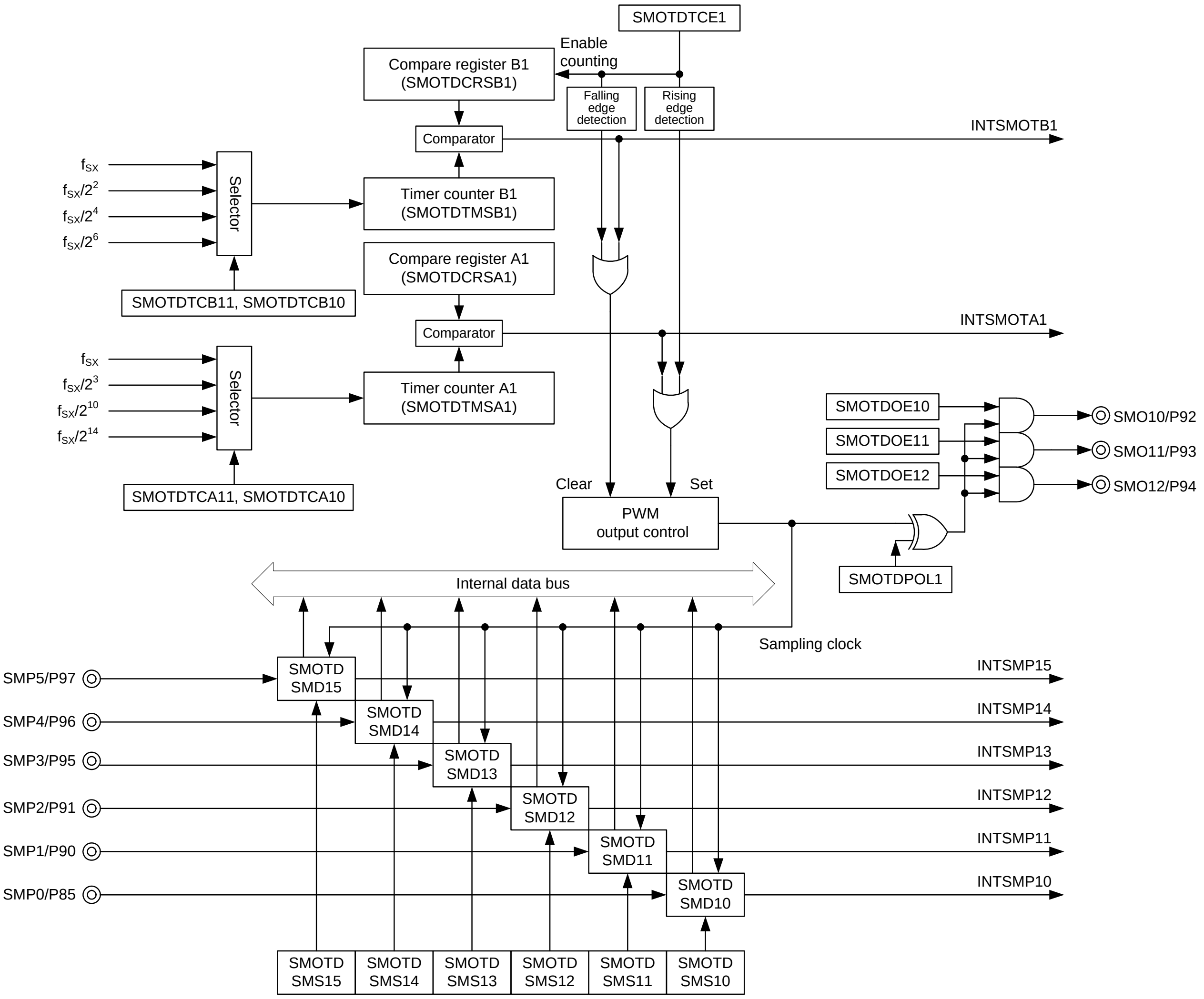
Figures 14-1 and **14-2** show the block diagrams of the sampling output timer detector.

Figure 14-1. Block Diagram of Sampling Output Timer Detector 0



Caution When using the sampling output timer detector, be sure to select the subsystem clock (WUTMMCK0 = 0).

Figure 14-2. Block Diagram of Sampling Output Timer Detector 1



Caution When using the sampling output timer detector, be sure to select the subsystem clock (WUTMMCK0 = 0).

14.3 Registers Controlling the Sampling Output Timer Detector

The sampling output timer detector is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- SMOTD timer counter Am (SMOTDTMSAm)
- SMOTD timer counter Bm (SMOTDTMSBm)
- SMOTD compare register Am (SMOTDCRSAm)
- SMOTD compare register Bm (SMOTDCRSBm)
- SMOTD clock select register m (SMOTDTCSm)
- SMOTD control register m (SMOTDCRm)
- SMOTD sampling level setting register m (SMOTDSMSm)
- SMOTD sampling pin status register m (SMOTDSMDm)
- SMOTD output control register m (SMOTDOEm)
- Port mode registers 0, 3, 5, 7 to 9 (PM0, PM3, PM5, PM7 to PM9)
- Port registers 0, 3, 5, 7 to 9, 13 (P0, P3, P5, P7 to P9, P13)

Remark m: Unit number (m = 0, 1)

(1) Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the sampling output timer detector 0 is to be used, be sure to set bit 4 (SMOTD0EN) to 1.

When the sampling output timer detector 1 is to be used, be sure to set bit 5 (SMOTD1EN) to 1.

The PER1 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 14-3. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

SMOTDmEN	Control of sampling output timer detector m input clock supply
0	Stops input clock supply. ● SFR used by the sampling output timer detector m cannot be written. The read value is 00H. However, the SFR is not initialized. [Notes 1, 2]
1	Enables input clock supply. ● SFR used by the sampling output timer detector m can be read and written.

Notes 1. To initialize the sampling output timer detector 0 and the SFR used by the sampling output timer detector 0, use bit 4 (SMOTD0RES) of PRR1.

2. To initialize the sampling output timer detector 1 and the SFR used by the sampling output timer detector 1, use bit 5 (SMOTD1RES) of PRR1.

Cautions 1. When using the sampling output timer detector m, be sure to set the following registers first while the SMOTDmEN bit is set to 1. If SMOTDmEN = 0, writing to the registers controlling the sampling output timer detector m is ignored, and, even if the register is read, only the default value is read (except for the port mode registers 0, 3, 5, 7 to 9 (PM0, PM3, PM5, PM7 to PM9), port registers 0, 3, 5, 7 to 9, 13 (P0, P3, P5, P7 to P9, P13)).

- **SMOTD timer counter Am (SMOTDTMSAm)**
- **SMOTD timer counter Bm (SMOTDTMSBm)**
- **SMOTD compare register Am (SMOTDCRSAm)**
- **SMOTD compare register Bm (SMOTDCRSBm)**
- **SMOTD clock select register m (SMOTDTCSm)**
- **SMOTD control register m (SMOTDCRm)**
- **SMOTD sampling level setting register m (SMOTDSMSm)**
- **SMOTD sampling pin status register m (SMOTDSMDm)**
- **SMOTD output control register m (SMOTDOEm)**

2. Be sure to clear bit 7 to 0.

Remark m: Unit number (m = 0, 1)

(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.

To place the sampling output timer detector 0 in the reset state, be sure to set bit 4 (SMOTD0RES) to 1.

To place the sampling output timer detector 1 in the reset state, be sure to set bit 5 (SMOTD1RES) to 1.

The PRR1 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 14-4. Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	<2>	<1>	<0>
PRR1	0	0	SMOTD1RES	SMOTD0RES	0	TRJ1RES	TRJ0RES	DSADRES

SMOTDmRES	Control resetting of the sampling output timer detector m
0	The sampling output timer detector m is released from the reset state.
1	The sampling output timer detector m is in the reset state.

Remark m: Unit number (m = 0, 1)

(3) SMOTD timer counter Am (SMOTDTMSAm)

The SMOTDTMSAm counter is an 8-bit counter that is used to generate the period of the PWM waveform of the sampling clock output from the SMOmj pin.

The SMOTDTMSAm counter cannot be directly manipulated by a program (cannot be read/written).

The counter value is cleared to 00H in the following cases.

(i) The reset signal is generated.

(ii) SMOTDTCEm is cleared (when "1" is changed to "0").

(iii) SMOTDTMSAm matches SMOTDCRSAm.

Figure 14-5. Format of SMOTD Timer Counter Am (SMOTDTMSAm)

Address: — After reset: 00H R/W disabled

Symbol	7	6	5	4	3	2	1	0
SMOTDTMSAm	SMOTDTMSAm[7:0]							

SMOTDTMSAm [7:0]	SMOTD timer counter Am
00H to FFH	Counting start condition: SMOTDTCEm is set (when "0" is changed to "1"). Counting condition: Counter is incremented by the clock source selected with the SMOTDTCAm1 and SMOTDTCAm0 bits when SMOTDTCEm = 1. Counter clear condition: • The reset signal is generated. • SMOTDTCEm is cleared (when "1" is changed to "0"). • SMOTDTCAm matches SMOTDCRSAm.

Caution If the counting condition and counter clear condition are generated simultaneously, the counter is cleared.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(4) SMOTD timer counter Bm (SMOTDTMSBm)

The SMOTDTMSBm counter is an 8-bit counter that is used to generate a high pulse width of the PWM waveform of the sampling clock output from the SMOmj pin.

The SMOTDTMSBm counter cannot be directly manipulated by a program (cannot be read/written).

The counter value is cleared to 00H in the following cases.

(i) The reset signal is generated.

(ii) SMOTDTCEm is cleared (when "1" is changed to "0").

(iii) SMOTDTMSBm matches SMOTDCRSBm.

Figure 14-6. Format of SMOTD Timer Counter Bm (SMOTDTMSBm)

Address: — After reset: 00H R/W disabled

Symbol	7	6	5	4	3	2	1	0
SMOTDTMSBm	SMOTDTMSBm[7:0]							

SMOTDTMSBm [7:0]	SMOTD timer counter Bm
00H to FFH	Counting start condition: SMOTDTCEm is set (when "0" is changed to "1"). Counting condition: Counter is incremented by the clock source selected with the SMOTDTCBm bit when SMOTDTCEm = 1. Counter clear condition: • The reset signal is generated. • SMOTDTCEm is cleared (when "1" is changed to "0"). • SMOTDTMSBm matches SMOTDCRSBm

Caution If the counting condition and counter clear condition are generated simultaneously, the counter is cleared.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(5) SMOTD compare register Am (SMOTDCRSAm)

The SMOTDCRSAm register is used to set the period of the PWM waveform of the sampling clock output from the SMOmj pin.

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

Reset signal generation clears this register to 00H.

Figure 14-7. Format of SMOTD Compare Register Am (SMOTDCRSAm)

Address: F02B0H (SMOTDCRSA0), F02B8H (SMOTDCRSA1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SMOTDCRSAm	SMOTDCRSAm[7:0]							

SMOTDCRSAm[7:0]	Setting of compare value for generating the sampling clock output period
00H (default)	Setting prohibited
01H • • • FFH	PWM waveform whose period is $(1/f_{SX})$[Note] × (SMOTDCRSAm + 1) is output from the SMOmj pin. Setting example: SMOTDCRSAm = 01H $(1/f_{SX})$[Note] × (1 + 1) = 52 μs@38.4 kHz SMOTDCRSAm = 57H (in case of 2^3 division) $(23/f_{SX})$[Note] × (87 + 1) = 18.33 ms@38.4 kHz

Note The clock of the timer counter A is set by the SMOTDTCAm1 and SMOTDTCAm0 bits.

Cautions 1. Do not change the setting of the SMOTDCRSAm register during counting (SMOTDTCEm = 1).

2. Using this register (SMOTDTCEm = 1) with the default (00H) is prohibited. Set any value other than 00H.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(6) SMOTD compare register Bm (SMOTDCRSBm)

The SMOTDCRSBm register is used to set the high pulse width of the PWM waveform of the sampling clock output from the SMOmj pin.

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

Reset signal generation clears this register to 00H.

Figure 14-8. Format of SMOTD Compare Register Bm (SMOTDCRSBm)

Address: F02B1H (SMOTDCRSB0), F02B9H (SMOTDCRSB1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SMOTDCRSBm	SMOTDCRSBm[7:0]							

SMOTDCRSBm[7:0]	Setting of compare value for generating the sampling clock output high pulse width
00H (default) • • • FFH	PWM waveform whose high pulse width is (1/f_{SX}) × (SMOTDCRSBm + 1) is output from the SMOmj pin. Setting example: SMOTDCRSBm = 00H (1/f_{SX})[Note] × (0 + 1) = 26 μs@38.4 kHz SMOTDCRSBm = 01H (1/f_{SX})[Note] × (1 + 1) = 52 μs@38.4 kHz SMOTDCRSBm = 99H (1/f_{SX})[Note] × (153 + 1) = 4.01 ms@38.4 kHz

Note The clock of the timer counter B is set by the SMOTDTCBm1 and SMOTDTCBm0 bits.

Cautions 1. Do not change the setting of the SMOTDCRSBm register during counting (SMOTDTCEm = 1).

2. Set this register to satisfy SMOTDCRSAm > SMOTDCRSBm.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(7) SMOTD clock select register m (SMOTDTCSm)

The SMOTDTCSm register is used to select the count clock for the SMOTD timer counter Am (SMOTDTMSAm) and SMOTD timer counter Bm (SMOTDTMSBm).

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

Reset signal generation clears this register to 00H.

Figure 14-9. Format of SMOTD Clock Select Register m (SMOTDTCSm)

Address: F02B2H (SMOTDTCS0), F02BAH (SMOTDTCS1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SMOTDTCSm	0	0	SMOTDTCBm1	SMOTDTCBm0	0	0	SMOTDTCAm1	SMOTDTCAm0

SMOTDTCBm1 SMOTDTCBm0	SMOTD timer counter Bm (SMOTDTMSBm) clock select bits
00B (default)	fsx (30.5 μs/26 μs)
01B	fsx/2^2 (122 μs/104 μs)
10B	fsx/2^4 (488 μs/416.7 μs)
11B	fsx/2^6 (1.95 ms/1.67 ms)

SMOTDTCAm1 SMOTDTCAm0	SMOTD timer counter Am (SMOTDTMSAm) clock select bits
00B (default)	fsx (30.5 μs/26 μs)
01B	fsx/2^3 (244 μs/208 μs)
10B	fsx/2^{10} (31.3 ms/26.7 ms)
11B	fsx/2^{14} (500 ms/426.7 ms)

Cautions 1. Set bits 7, 6, 3, and 2 to the default value.

2. Stop counting (SMOTDTCEm = 0) before setting the register.

Remarks 1. The values in parentheses indicate the settings for f_{SX} = 32.768 kHz or 38.4 kHz (f_{SX}: sub clock frequency).

2. m: Unit number (m = 0, 1)

(8) SMOTD control register m (SMOTDCRm)

The SMOTDCRm register is used to select the polarity of the sampling clock signal output from the SMOmj pin and enable the counting operation of the sampling output timer.

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

Reset signal generation clears this register to 00H.

Figure 14-10. Format of SMOTD Control Register m (SMOTDCRm)

Address: F02B3H (SMOTDCR0), F02BBH (SMOTDCR1) After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	<0>
SMOTD CRm	0	0	0	SMOTDPOLm	0	0	0	SMOTDTCEm

SMOTDPOLm	SMOmj pin polarity select bit
0 (default)	Active-high (default: low level output)
1	Active-low (default: high level output)

Cautions 1. The setting of this bit applies to all the sampling clock signals output from the SMO0 to SMO2 pins.

2. Set this bit while counting is stopped (SMOTDTCEm = 0) and sampling output is disabled (SMOTDOEm2 to SMOTDOEm0 = 000B).

SMOTDTCEm	Timer counting operation enable
0 (default)	Stops counting.
1	Enables counting.

Cautions 1. Set the SMOTDTCEm bit to 1 after setting the SMOTDCRSAm, SMOTDCRSBm, SMOTDTCSm, SMOTDSMSm, and SMOTDSMDm registers and SMOTDPOLm bit.

2. Following a change to the setting after setting the SMOTDTCEm bit, wait for at least 3 cycles of the subsystem clock.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(9) SMOTD sampling level setting register m (SMOTDSMSm)

The SMOTDSMSm register is used to set the conditions for generating the sampling detector detection interrupt (INTSMPmi).

When the level of the sampling input to the SMP0 to SMP5 pins match the active level set with SMOTDSMSm, the sampling detector detection interrupt (INTSMPmi) can be generated.

The input to the SMP0 to SMP5 pins is sampled on the falling edge of the sampling clock signal.

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

Reset signal generation clears this register to 00H.

Figure 14-11. Format of SMOTD Sampling Level Setting Register m (SMOTDSMSm)

Address: F02B4H (SMOTDSMS0), F02BCH (SMOTDSMS1) After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
SMOTDSMSm	0	0	SMOTDSMSm5	SMOTDSMSm4	SMOTDSMSm3	SMOTDSMSm2	SMOTDSMSm1	SMOTDSMSm0

SMOTDSMSmi	Sampling signal active level setting bit
0	An interrupt request is generated when the low level on SMPi is detected on the falling edge of the sampling clock signal.
1	An interrupt request is generated when the high level on SMPi is detected on the falling edge of the sampling clock signal.

Cautions 1 Do not change the setting of the SMOTDSMSm register during counting (SMOTDTCEm = 1).

2. Set bits 7 and 6 to the default value.

Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5)

(10) SMOTD sampling pin status register m (SMOTDSMDm)

The SMOTDSMDm register is used to detect the status of the SMP0 to SMP5 pins on the falling edge of the PWM output waveform output to the sampling clock pin (SMOmj).

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

Reset signal generation clears this register to 00H.

Figure 14-12. Format of SMOTD Sampling Pin Status Register m (SMOTDSMDm)

Address: F02B5H (SMOTDSMD0), F02BDH (SMOTDSMD1) After reset: 00H R

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
SMOTDSMDm	0	0	SMOTDSMDm5	SMOTDSMDm4	SMOTDSMDm3	SMOTDSMDm2	SMOTDSMDm1	SMOTDSMDm0

SMOTDSMDmi	SMPi pin status
0 (default)	Low level
1	High level

Caution Read the SMOTDSMDm register after the INTSMPmi interrupt is generated.

Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5), j: Output channel number (j = 0 to 2)

(11) SMOTD output control register m (SMOTDOEm)

The SMOTDOEm register is used to disable or enable output of the sampling clock from the SMOmj pin.

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

Reset signal generation clears this register to 00H.

Figure 14-13. Format of SMOTD Output Control Register m (SMOTDOEm)

Address: F02B6H (SMOTDOE0), F02BEH (SMOTDOE1) After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
SMOTDOEm	0	0	0	0	0	SMOTDOEm2	SMOTDOEm1	SMOTDOEm0

SMOTDOEmj	SMOmj pin output control bit
0 (default)	Output disabled (port mode).
1	Output enabled (sampling mode).

Cautions 1. Do not change the setting of the SMOTDOEm register during counting (SMOTDTCEm = 1).

2. Set bits 7 through 3 to the default value.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

(12) Port mode registers 0, 3, 5, 7 to 9 (PM0, PM3, PM5, PM7 to PM9)

These registers specify input or output mode for the ports 0, 3, 5, and 7 to 9 in 1-bit units.

To use the port pin (such as P85/SMP0/SEG41/SO30/TXD3, P90/SMP1/COM0/(SEG31)) on which a sampling input pin function is multiplexed for sampling input, set the corresponding bit in the port mode register (PMxx) to 1. At this time, the bit in the port register (Pxx) may be 0 or 1.

Example: When P85/SMP0/SEG41/SO30/TXD3 is to be used for sampling input:
Set the PM85 bit of port mode register 8 to 1.
Set the P85 bit of port register 8 to 0 or 1.
Set the PFSEG41 bit of LCD port function register 6 to 0.

To use the port pin (such as P50/SMO00/SEG32, P51/SMO01/SEG33/TxD4) on which a sampling output pin function is multiplexed for sampling output, set the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P50/SMO00/SEG32 is to be used for sampling output:
Set the PM50 bit of port mode register 5 to 0.
Set the P50 bit of port register 5 to 0.
Set the PFSEG32 bit of LCD port function register 5 to 0.

The PM0, PM3, PM5, and PM7 to PM9 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-14. Format of Port Mode Registers (PM0, PM3, PM5, PM7 to PM9)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	1	1

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

Address: FFF28H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM8	1	1	PM85	PM84	PM83	PM82	PM81	PM80

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

PMmn	Selection of I/O mode for Pmn pin (m = 0, 3, 5, 7 to 9; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operation of Sampling Output Timer Detector

Figure 14-15 shows the sampling clock output timing and **Figure 14-16** shows the sampling detection timing.

Figure 14-15. SMO0 to SMO2 Output Timing

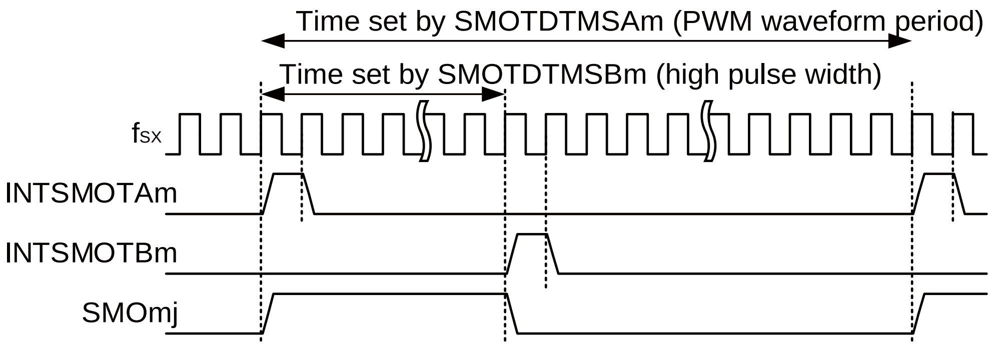
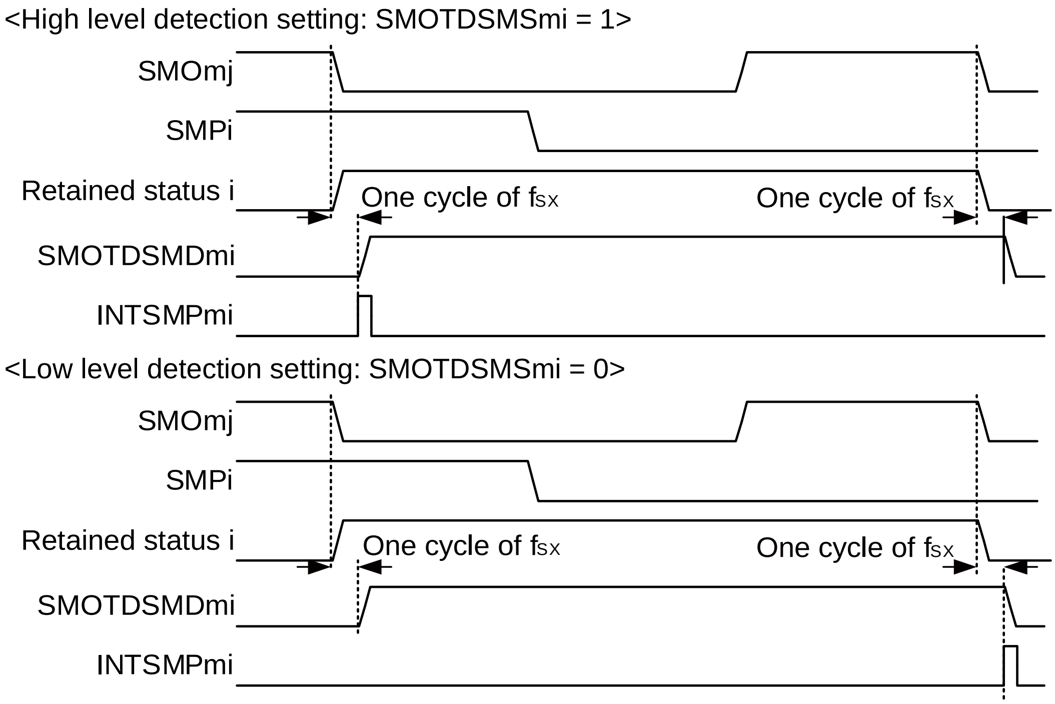

Time set by SMOTDTMSAm (PWM waveform period)
Time set by SMOTDTMSBm (high pulse width)
fSX
INTSMOTAm
INTSMOTBm
SMOmj

Figure 14-16. Sampling Detection Timing

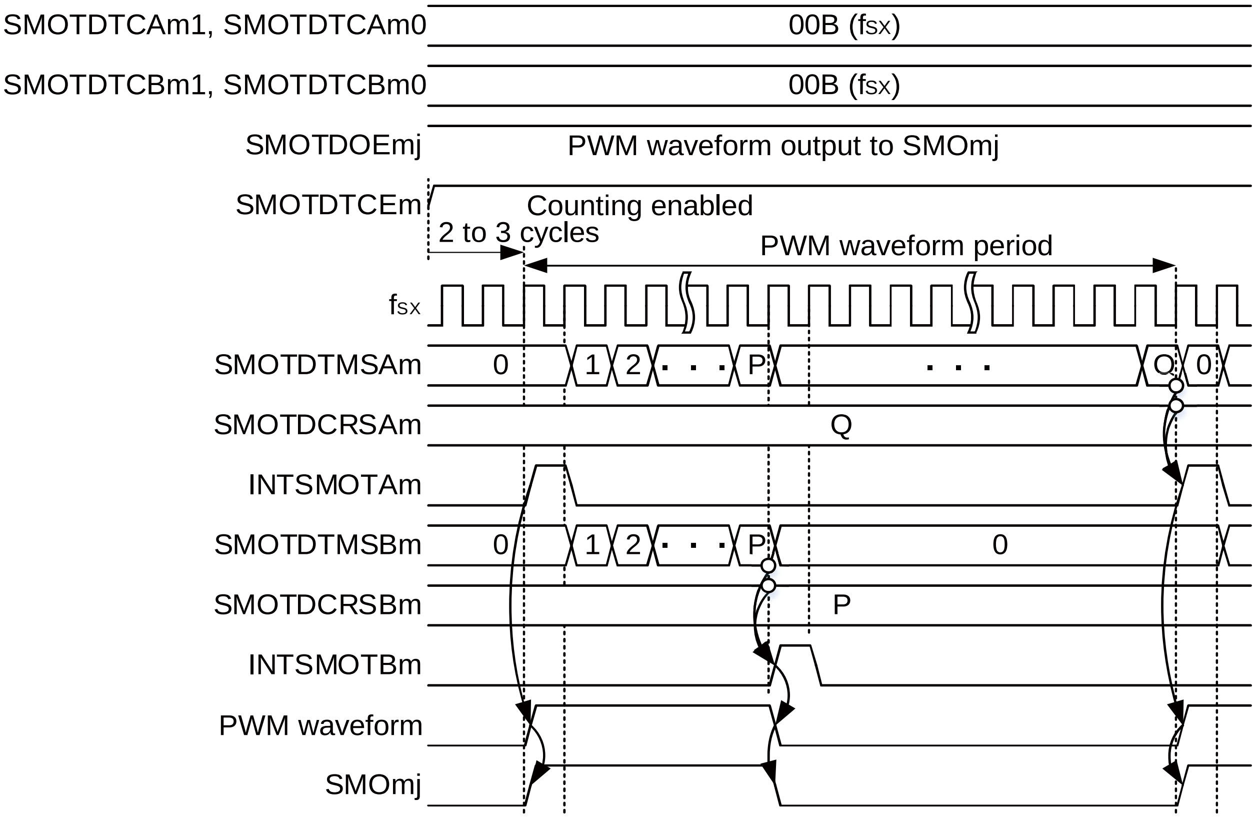


Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5), j: Output channel number (j = 0 to 2)

14.4.1 Sampling Clock Output Function

The sampling clock output function outputs the PWM waveform. The PWM waveform is generated by setting the period and high pulse width. The PWM waveform period is set by the SMOTDTCAm1 and SMOTDTCAm0 bits and SMOTDCRSAm register. The high pulse width of the PWM waveform is set by the SMOTDTCBm1 and SMOTDTCBm0 bits and SMOTDCRSBm register. The generated PWM waveform is output to the SMO0 to SMO2 pins according to the setting of the PWM waveform output (SMOTDOEmj bit). On the falling edge of the PWM waveform, the sampling output timer compare match interrupt (INTSMOTBm) is output. On the rising edge of the PWM waveform, the sampling output timer interval interrupt (INTSMOTAm) is output. The polarity of the value to be output to SMOmj can be selected by the SMOTDPOLm bit. **Figure 14-17** shows the timing of the sampling clock output (without division).

Figure 14-17. Sampling Clock Output (without division)

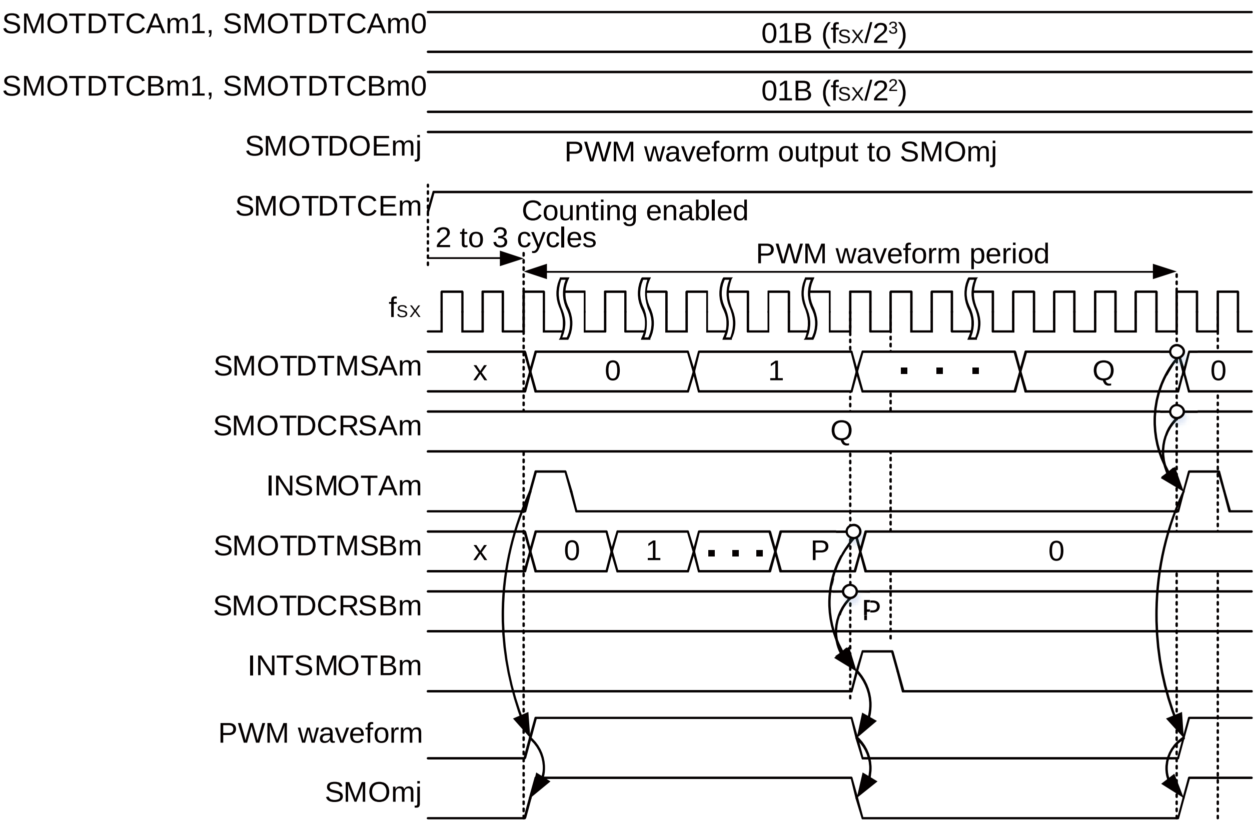


Remarks 1. The above figure applies to the case in which SMOTDPOLm = 0 (active-high).

2. m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

The sampling output timer detector has a function of dividing the PWM waveform. **Figure 14-18** shows the timing of the sampling clock output (with division).

Figure 14-18. Sampling Clock Output (with division)



Remarks 1. The above figure applies to the case in which SMOTDPOLm = 0 (active-high).

2. m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

Table 14-3. PWM Waveform of Sampling Clock Output (f_{SX} = 38.4 kHz)

SMOTD TCAm1	SMOTD TCAm0	SMOTD TCBm1	SMOTD TCBm0	High Pulse Width		PWM Waveform Period	
0	0	0	0	$(b + 1)(1/f_{SX})$	26.0 μs to 6.7 ms	$(a + 1)(1/f_{SX})$	52.0 μs to 6.7 ms
0	1	0	0	$(b + 1)(1/f_{SX})$	26.0 μs to 6.7 ms	$(a + 1)(2^{3}/f_{SX})$	416.7 μs to 53.3 ms
		0	1	$(b + 1)(2^{2}/f_{SX})$	104.2 μs to 26.7 ms		
1	0	0	0	$(b + 1)(1/f_{SX})$	26.0 μs to 6.7 ms	$(a + 1)(2^{10}/f_{SX})$	53.3 ms to 6.8 s
		0	1	$(b + 1)(2^{2}/f_{SX})$	104.2 μs to 26.7 ms		
		1	0	$(b + 1)(2^{4}/f_{SX})$	416.7 μs to 106.7 ms		
		1	1	$(b + 1)(2^{6}/f_{SX})$	1.7 ms to 426.7 ms		
1	1	0	0	$(b + 1)(1/f_{SX})$	26.0 μs to 6.7 ms	$(a + 1)(2^{14}/f_{SX})$	853.3 ms to 109.2 s
		0	1	$(b + 1)(2^{2}/f_{SX})$	104.2 μs to 26.7 ms		
		1	0	$(b + 1)(2^{4}/f_{SX})$	416.7 μs to 106.7 ms		
		1	1	$(b + 1)(2^{6}/f_{SX})$	1.7 ms to 426.7 ms		

Cautions 1. "a" represents the SMOTDCRSAm[7:0] value, and "b" represents the SMOTDCRSBm[7:0] value.

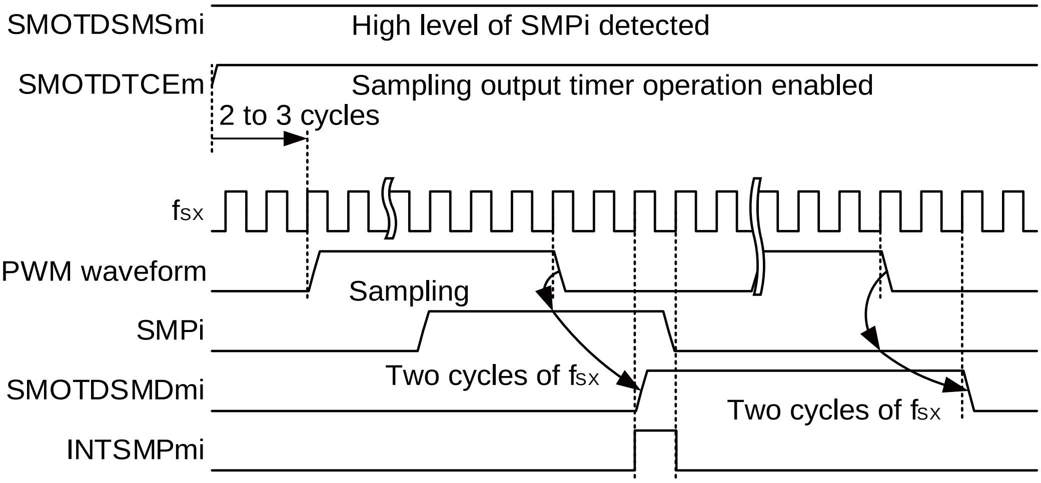
2. Combinations of SMOTDTCAm1, SMOTDTCAm0, SMOTDTCBm1, and SMOTDTCBm0 settings not shown in Table 14-3 are prohibited. Set the bits to satisfy PWM period > high pulse width.

Remark m: Unit number (m = 0, 1)

14.4.2 Sampling Detector Function

The sampling detector function stores the sampling signal (SMPi) in the SMOTDSMDmi bit on the falling edge of the PWM waveform provided by the sampling clock output function. Here, a delay of two cycles of fSX is produced relative to the falling edge of the PWM waveform. If the sampling signal (SMPi) matches the condition set with the SMOTDSMSmi bit, the sampling detector detection interrupt (INTSMPmi) is output simultaneously with the storage of the sampling signal in the SMOTDSMDmi bit.

Figure 14-19. Sampling Clock Output Function

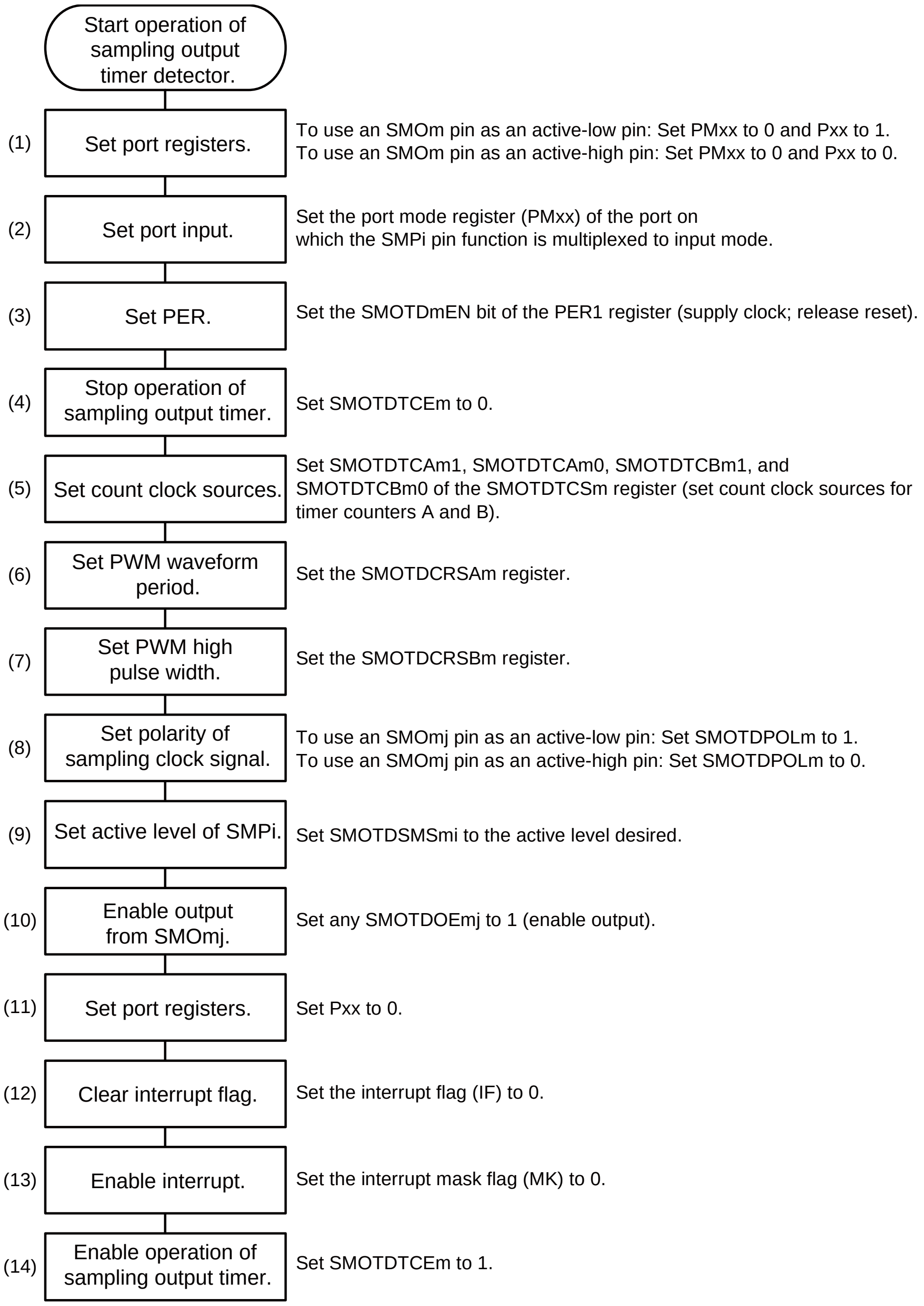


Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5)

14.4.3 Setting the Operation of the Sampling Output Timer Detector

Figure 14-20 shows the procedure for starting the operation of the sampling output timer detector.

Figure 14-20. Procedure for Starting Operation


Start operation of sampling output timer detector.
(1)
Set port registers.
(2)
Set port input.
(3)
Set PER.
(4)
Stop operation of sampling output timer.
(5)
Set count clock sources.
(6)
Set PWM waveform period.
(7)
Set PWM high pulse width.
(8)
Set polarity of sampling clock signal.
(9)
Set active level of SMPi.
(10)
Enable output from SMOmj.
(11)
Set port registers.
(12)
Clear interrupt flag.
(13)
Enable interrupt.
(14)
Enable operation of sampling output timer.

(1) To use an SMOm pin as an active-low pin: Set PMxx to 0 and Pxx to 1.
To use an SMOm pin as an active-high pin: Set PMxx to 0 and Pxx to 0.

(2) Set the port mode register (PMxx) of the port on which the SMPi pin function is multiplexed to input mode.

(3) Set the SMOTDmEN bit of the PER1 register (supply clock; release reset).

(4) Set SMOTDTCEm to 0.

(5) Set SMOTDTCAm1, SMOTDTCAm0, SMOTDTCBm1, and SMOTDTCBm0 of the SMOTDTCSm register (set count clock sources for timer counters A and B).

(6) Set the SMOTDCRSAm register.

(7) Set the SMOTDCRSBm register.

(8) To use an SMOmj pin as an active-low pin: Set SMOTDPOLm to 1.
To use an SMOmj pin as an active-high pin: Set SMOTDPOLm to 0.

(9) Set SMOTDSMSmi to the active level desired.

(10) Set any SMOTDOEmj to 1 (enable output).

(11) Set Pxx to 0.

(12) Set the interrupt flag (IF) to 0.

(13) Set the interrupt mask flag (MK) to 0.

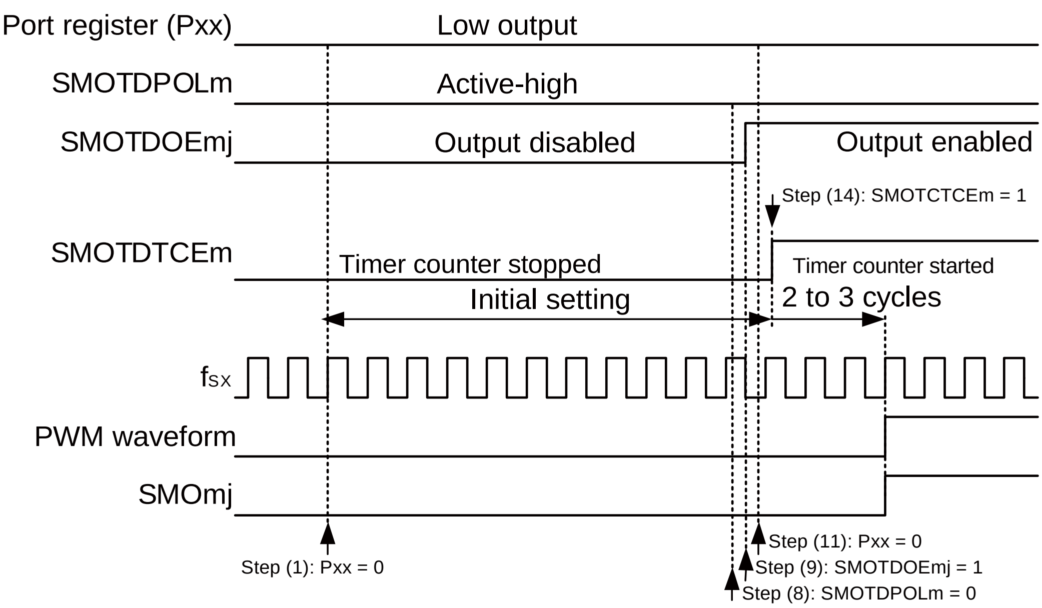
(14) Set SMOTDTCEm to 1.

Caution **For setting the Pxx and PMxx registers, refer to (12) in 14.3 Registers Controlling the Sampling Output Timer Detector.**

Remark m: Unit number (m = 0, 1), i: Input channel number (i = 0 to 5), j: Output channel number (j = 0 to 2)

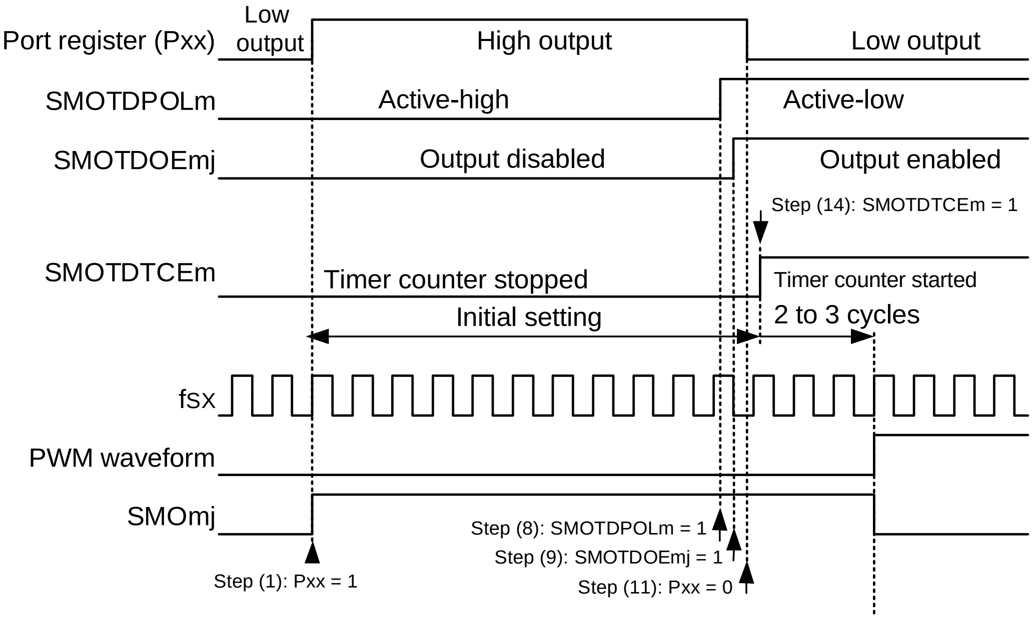
Figure 14-21 and **Figure 14-22** show the timing of starting the operation when an SMOmj pin is used as an active-low pin and an active-high pin, respectively.

Figure 14-21. Timing of Stating Operation When SMOmj Pin is Active High (when SMOTDPOLm = 0)



Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

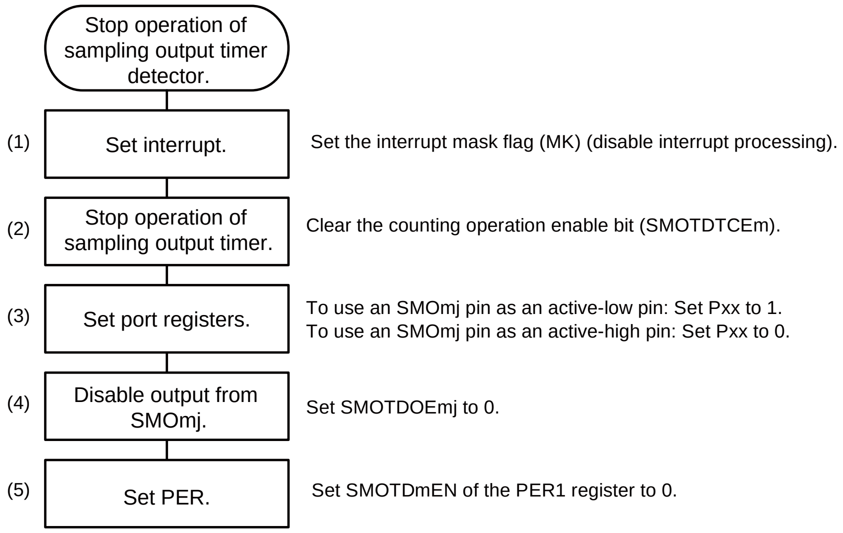
Figure 14-22. Timing of Stating Operation When SMOmj Pin is Active Low (when SMOTDPOLm = 1)



Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

Figure 14-23 shows the procedure for stopping the operation of the sampling output timer detector

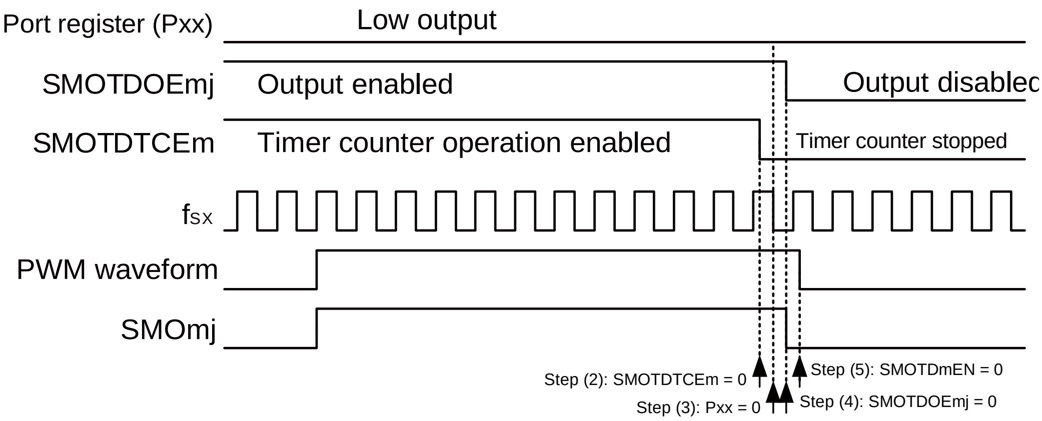
Figure 14-23. Procedure for Stopping Operation


Stop operation of sampling output timer detector.
(1) Set interrupt.
Set the interrupt mask flag (MK) (disable interrupt processing).
(2) Stop operation of sampling output timer.
Clear the counting operation enable bit (SMOTDTCEm).
(3) Set port registers.
To use an SMOmj pin as an active-low pin: Set Pxx to 1.
To use an SMOmj pin as an active-high pin: Set Pxx to 0.
(4) Disable output from SMOmj.
Set SMOTDOEmj to 0.
(5) Set PER.
Set SMOTDmEN of the PER1 register to 0.

Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

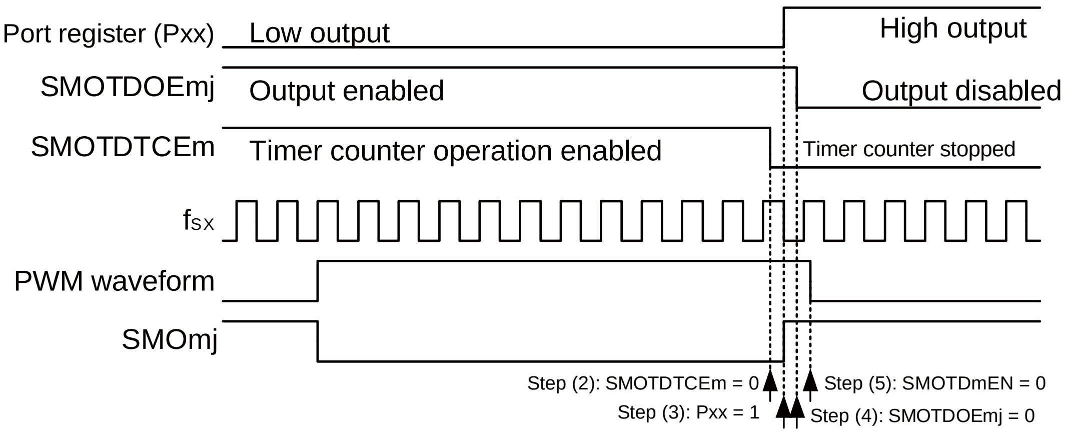
The sampling output timer stops three cycles of the subsystem clock after the setting to stop it is made. The interrupt may thus be generated before it stops completely. Executing step (5) of the procedure for stopping operation, it stops immediately. **Figure 14-24** shows the timing of stopping the operation when an SMOmj pin is used as an active-high pin and **Figure 14-25** shows the timing of stopping the operation when an SMOmj pin is used as an active-low pin.

Figure 14-24. Timing of Stopping Operation When SMOmj Pin is Active High (when SMOTDPOLm = 0)



Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

Figure 14-25. Timing of Stopping Operation When SMOmj Pin is Active High (when SMOTDPOLm = 1)



Remark m: Unit number (m = 0, 1), j: Output channel number (j = 0 to 2)

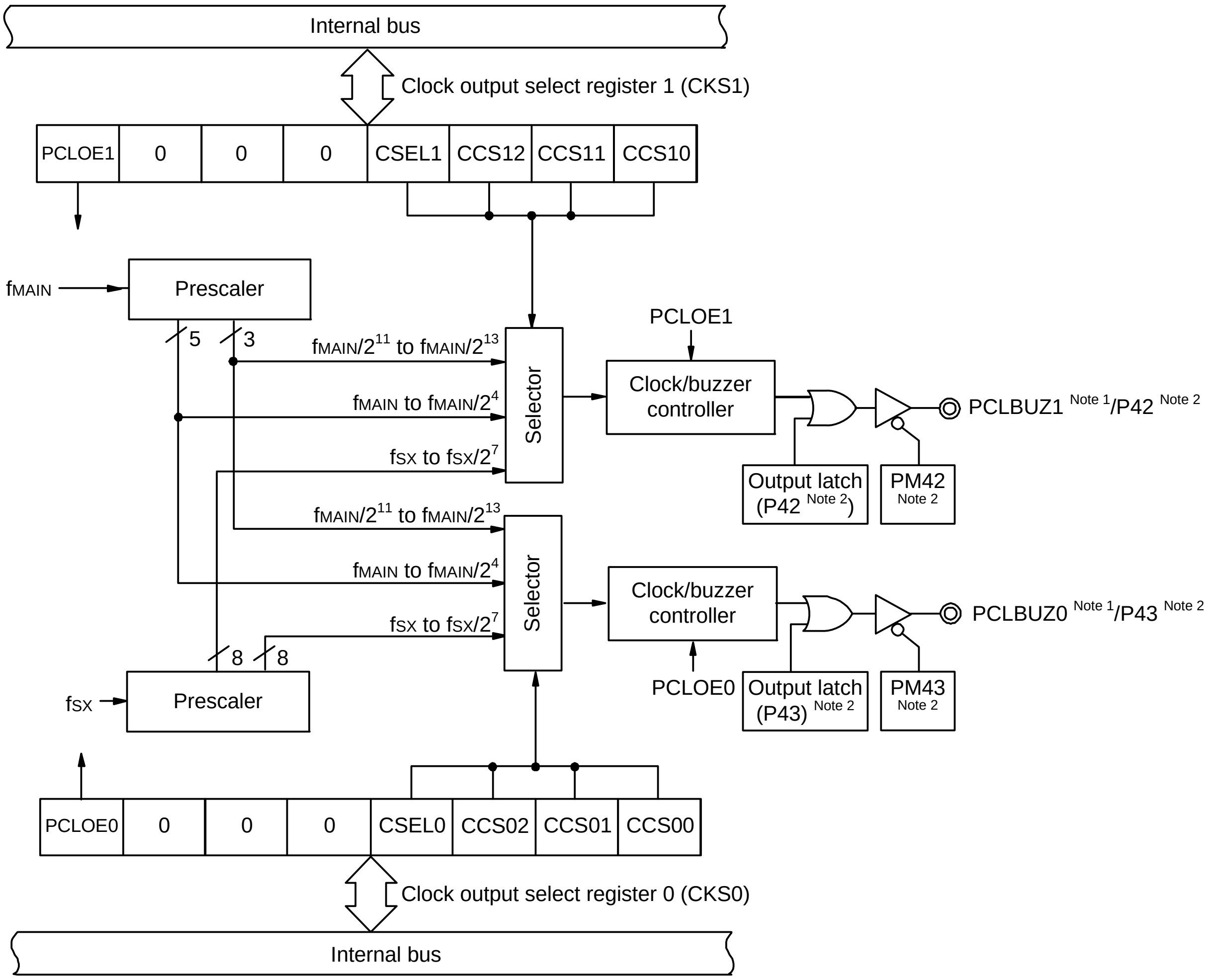
CHAPTER 15 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

15.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for clock output for supply to peripheral ICs.
Buzzer output is a function to output a square wave of buzzer frequency.
One pin can be used to output a clock or buzzer sound.
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).
Figure 15-1 shows the block diagram of clock output/buzzer output controller.

Remark n = 0, 1

Figure 15-1. Block Diagram of Clock Output/Buzzer Output Controller



Notes 1. For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **43.4 AC Characteristics**.

2. The port mode register (PMxx) and port register (Pxx) to be set depend on the product and the setting of the peripheral I/O redirection register 0 (PIOR0).
For details, see **4.5 Register Settings When Using Alternate Function.**

15.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 15-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Subsystem clock supply option control register (OSMC) Port mode registers 3, 4 (PM3, PM4) Port registers 3, 4 (P3, P4)

15.3 Registers Controlling Clock Output/Buzzer Output Controller

The following register is used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Subsystem clock supply option control register (OSMC)
- Port mode registers 3, 4 (PM3, PM4)
- Port registers 3, 4 (P3, P4)

15.3.1 Clock output select registers n (CKSn)

This register set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-2. Format of Clock Output Select Registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H 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 disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection					
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz	f_{MAIN} = 32 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz	Setting prohibited[Note]	Setting prohibited[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz	12 MHz	16 MHz
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz	8 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz	4 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz	2 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz	15.63 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz	7.81 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz	3.91 kHz
1	0	0	0	f_{SX}	32.768 kHz/38.4 kHz				
1	0	0	1	$f_{SX}/2$	16.384 kHz/19.2 kHz				
1	0	1	0	$f_{SX}/2^2$	8.192 kHz/9.6 kHz				
1	0	1	1	$f_{SX}/2^3$	4.096 kHz/4.8 kHz				
1	1	0	0	$f_{SX}/2^4$	2.048 kHz/2.4 kHz				
1	1	0	1	$f_{SX}/2^5$	1.024 kHz/1.2 kHz				
1	1	1	0	$f_{SX}/2^6$	512 Hz/600 Hz				
1	1	1	1	$f_{SX}/2^7$	256 Hz/300 Hz				

Note Use the output clock within a range of 16 MHz. See **43.4 AC Characteristics** for details.

Caution Change the output clock after disabling clock output (PCLOEn = 0).

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency
f_{SX}: Sub clock

15.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P43/TI00/TO00/PCLBUZ0/INTP7, P42/INTP6/TI01/TO01/PCLBUZ1) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P43/TI00/TO00/PCLBUZ0/INTP7 is to be used for clock or buzzer output
Set the PM43 bit of port mode register 4 to 0.
Set the P43 bit of port register 4 to 0.
Make the setting for channel 0 of the timer array unit to not be in use.

15.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock/buzzer selected by clock output select register 0 (CKS0).
The PCLBUZ1 pin outputs a clock/buzzer selected by clock output select register 1 (CKS1).

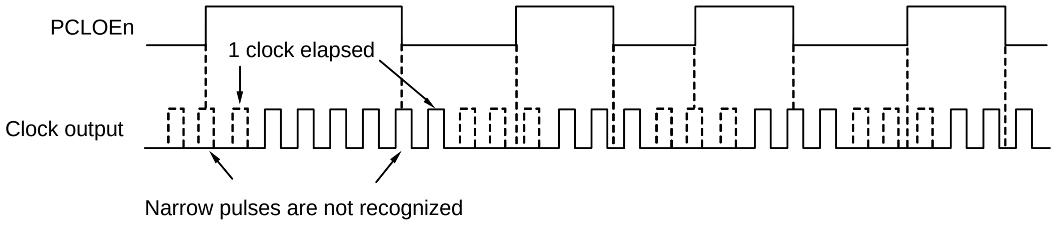
15.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedures.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Px) which correspond to the port which has a pin used as the PCLBUZ0 pin.
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).
<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remarks 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. **Figure 15-3** shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.
2. n = 0, 1

Figure 15-3. Timing of Outputting Clock from PCLBUZn Pin



15.5 Cautions of Clock Output/buzzer Output Controller

- When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.
- Setting the WUTMMCK0 bit in the subsystem clock supply option control register (OSMC) to 1 disables operation of the clock output/buzzer output controller.

CHAPTER 16 WATCHDOG TIMER

16.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).

The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

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 **CHAPTER 30 RESET FUNCTION**.

When 75% of the overflow time + 1/2 f_{IL} is reached, an interval interrupt can be generated.

16.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 16-1. Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

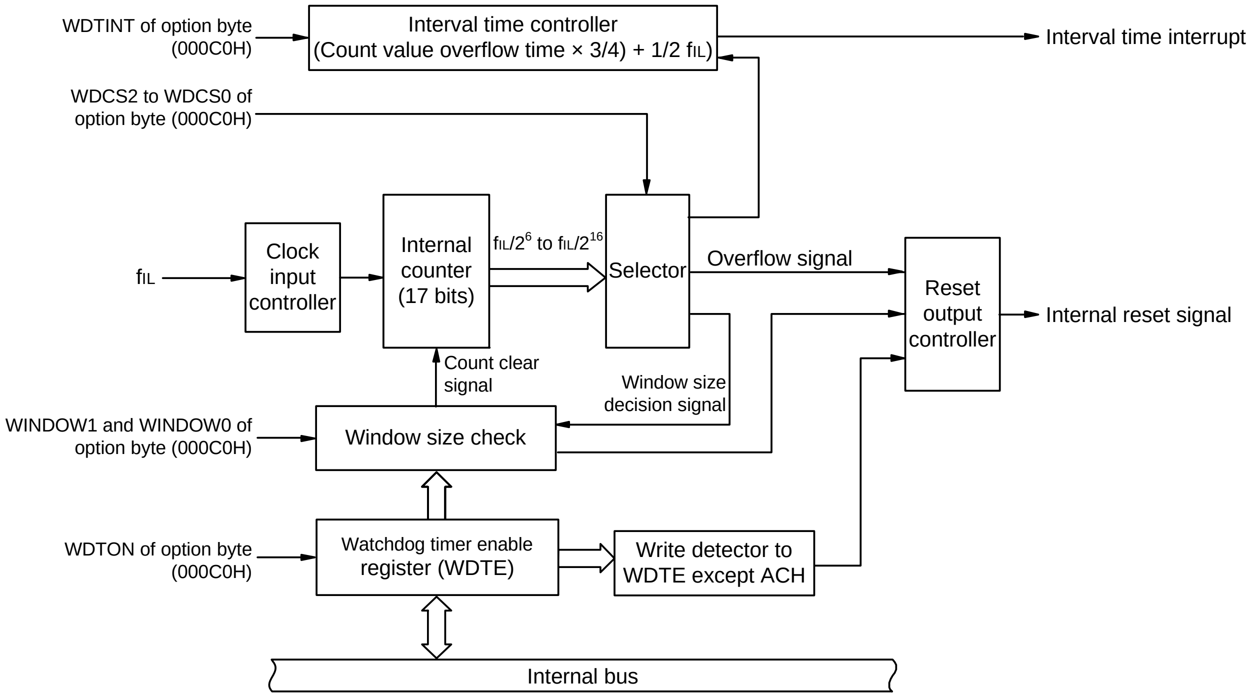
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 16-2. 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 **CHAPTER 37 OPTION BYTE**.

Figure 16-1. Block Diagram of Watchdog Timer


WDTINT of option byte (000C0H)
Interval time controller (Count value overflow time × 3/4) + 1/2 fIL)
Interval time interrupt
WDCS2 to WDCS0 of option byte (000C0H)
fIL
Clock input controller
Internal counter (17 bits)
fIL/2^6 to fIL/2^16
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size decision signal
WINDOW1 and WINDOW0 of option byte (000C0H)
Window size check
WDTON of option byte (000C0H)
Watchdog timer enable register (WDTE)
Write detector to WDTE except ACH
Internal bus

Remark fIL: Low-speed on-chip oscillator clock

16.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

16.3.1 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.
Reset signal generation sets this register to 9AH or 1AH [Note].

Figure 16-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 1AH/9AH [Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions
1. **If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.**
2. **If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.**
3. **The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).**

16.4 Operation of Watchdog Timer

16.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 37 OPTION BYTE**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **16.4.2 Setting overflow time of watchdog timer** and **CHAPTER 37 OPTION BYTE**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **16.4.3 Setting window open period of watchdog timer** and **CHAPTER 37 OPTION BYTE**).

2. After a reset release, the watchdog timer starts counting.
3. 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.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated. An internal reset signal is 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

Cautions 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. After “ACH” is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.

3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP 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.

16.4.2 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 16-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms) Note
1	1	0	$2^{14}/f_{IL}$ (949.79 ms) Note
1	1	1	$2^{16}/f_{IL}$ (3799.18 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 overflow time of the watchdog timer is set to $2^{13}/f_{IL}$, $2^{14}/f_{IL}$, or $2^{16}/f_{IL}$,
- the interval interrupt is in use (the setting of the WDTINT bit of the relevant option byte is 1), and
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.

1. Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
2. Clear the watchdog timer counter.
3. Wait for at least 80 μs.
4. Clear the WDTIIF bit of the interrupt request flag register 0 (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

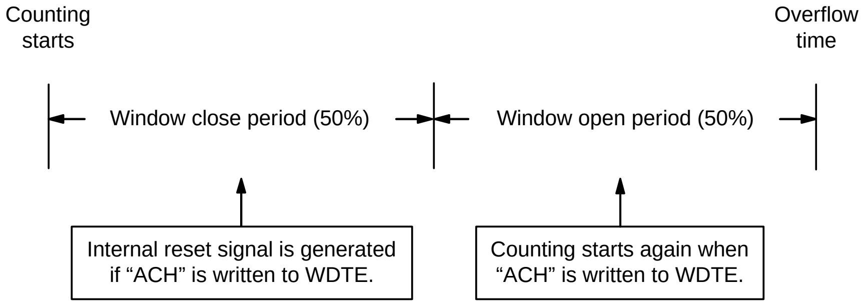
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

16.4.3 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%


Counting starts
Overflow time
Window close period (50%)
Window open period (50%)
Internal reset signal is generated if "ACH" is written to WDTE.
Counting starts again when "ACH" is written to WDTE.

Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 16-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75% Note
1	1	100%

Note When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 – 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 – 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

16.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% of the overflow time + 1/2 f_{IL} is reached.

Table 16-5. 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 overflow time + 1/2 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.

CHAPTER 17 12-BIT A/D CONVERTER

In this chapter, “PCLK” is used to refer to CPU/peripheral hardware clock (f_{CLK}).

17.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. Up to 6 channel analog inputs, temperature sensor output, and internal reference voltage (V_{BGR}) are selectable for conversion.

The 12-bit A/D converter converts a maximum of 6 selected channels of analog inputs, temperature sensor output, and internal reference voltage (V_{BGR}), which have been selected, into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 6 arbitrarily selected channels are converted only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 6 arbitrarily selected channels are continuously converted in ascending channel order.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.

It is prohibited to simultaneously select both temperature sensor output and internal reference voltage (V_{BGR}). Perform A/D conversion independently for the temperature sensor output or the internal reference voltage (V_{BGR}).

The external pin input (AV_{REFP}), voltage reference voltage output (VREFOUT), or analog reference voltage (AV_{DD}) is selectable as the reference voltage on the high-potential side. The external pin input (AV_{REFM}) or the analog reference voltage (AV_{SS1}) is selectable as the reference voltage on the low-potential side.

Table 17-1 lists the specifications of the 12-bit A/D converter and **Table 17-2** lists the functions of the 12-bit A/D converter. **Figure 17-1** shows a block diagram of the 12-bit A/D converter.

Table 17-1. Specifications of 12-Bit A/D Converter

Item	Description
Number of units	One unit
Input channels	Up to 6 channels
Extended analog function	Temperature sensor output, internal reference voltage (V_{BGR})
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time [Note 4]	2.25 µs per channel (fastest conversion time) (when A/D conversion clock ADCLK = 32 MHz)
A/D conversion clock	Peripheral hardware clock PCLK [Note 1] and A/D conversion clock ADCLK [Note 1] can be set so that the frequency division ratio should be one of the following. PCLK to ADCLK frequency division ratio = 1:1, 2:1, 4:1, 8:1
Data registers	• 6 registers for analog input • One register for temperature sensor output • One register for internal reference voltage (V_{BGR}) • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits [Note 2] in the A/D data registers in A/D-converted value addition mode.
Operating modes [Note 3]	• Single scan mode: A/D conversion is performed only once on the analog inputs of up to 6 channels arbitrarily selected. A/D conversion is performed only once on the temperature sensor output. A/D conversion is performed only once on the internal reference voltage (V_{BGR}). • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 6 channels arbitrarily selected.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the event link controller (ELC).
Functions	• Channel-dedicated sample-and-hold function (3 channels) • Variable sampling state count (selectable per channel) • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Automatic clear function of A/D data registers
Interrupt sources	• A/D scan end interrupt request (INTAD) can be generated on completion of single scan. • The INTAD interrupt can activate the data transfer controller (DTC).
Event link function	• Scan can be started by a trigger output by the ELC.
Reference voltage	• AV_{REFP}/VREFOUT or AV_{DD} is selectable as the reference voltage on the high-potential side. • AV_{REFM} or AV_{SS1} is selectable as the reference voltage on the low-potential side.

Notes 1. The frequency divisor for the peripheral hardware clock PCLK and A/D conversion clock ADCLK is set in the ADCKS register. Note that the ADCLK cannot be set to a frequency below 1 MHz.

2. The number of extended bits during addition differs depending on the A/D conversion accuracy and the addition count.
2-bit extension: A/D conversion accuracy = 1-time to 4-time conversion (addition zero to three times)
4-bit extension: A/D conversion accuracy = 16-time conversion in 12-bit mode (addition 15 times)

3. When the temperature sensor or internal reference voltage (V_{BGR}) is selected, do not use the continuous scan mode.

4. For the conversion time, refer to A/D conversion processing time (t_{CONV}) in **Table 17-9 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK)**.

Table 17-2. Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation
Analog input channels			ANI0 to ANI5, temperature sensor output, internal reference voltage (V_{BGR})
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Synchronous trigger	ELC trigger	Enabled
Interrupt			INTAD interrupt
Setting of clock supply stop function			PER0.ADCEN bit
Reset control			PRR0.ADCERES bit

Figure 17-1. Block Diagram of 12-Bit A/D Converter

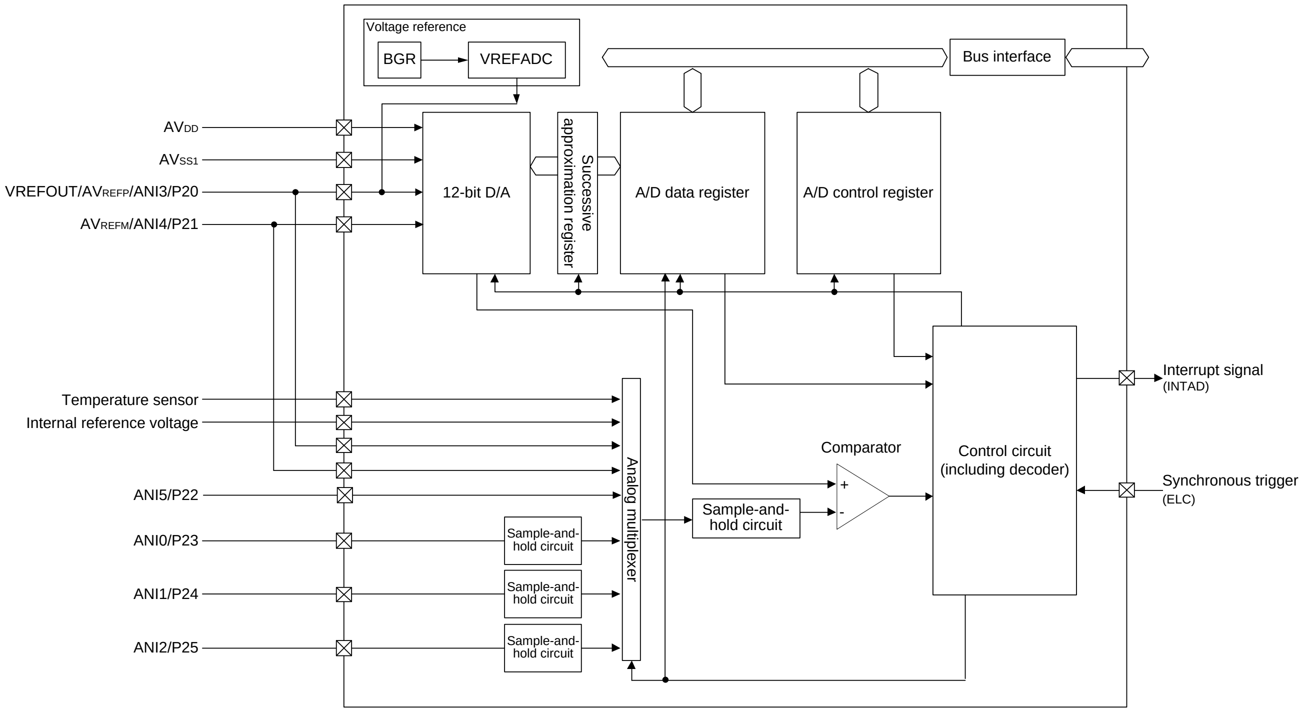


Table 17-3 lists the input pins of the 12-bit A/D converter.

Table 17-3. Input and Output Pins of 12-Bit A/D Converter

Pin Name	I/O	Function	Channel-Dedicated Sample-and-Hold Circuits
AV_{DD}	—	Analog block power supply pin	—
AV_{SS1}	—	Analog block ground pin	—
AV_{REFP}/VREFOUT	I/O	Reference power supply pin	—
AV_{REFM}	Input	Reference power supply ground pin	—
ANI0 to ANI2	Input	Analog input pins 0 to 2	Built-in
ANI3 to ANI5	Input	Analog input pins 3 to 5	—

17.2 Register Descriptions

Table 17-4. Registers of the 12-bit A/D Converter

Register Name	Symbol
Peripheral enable register 0	PER0
Peripheral reset control register 0	PRR0
A/D data registers y (y = 0 to 5)	ADDRy
A/D temperature sensor data register	ADTSDR
A/D internal reference voltage data register	ADOCDR
A/D self-diagnosis data register	ADRD
A/D control register	ADCSR
A/D channel select register A0	ADANSA0
A/D-converted value addition/average function channel select register 0	ADADS0
A/D-converted value addition/average count select register	ADADC
A/D control extended register	ADCER
A/D conversion start trigger select register	ADSTRGR
A/D conversion extended input control register	ADEXICR
A/D sampling state register n (n = 0 to 5, T, O)	ADSSTRn
A/D sample-and-hold circuit control register	ADSHCR
A/D high-potential/low-potential reference voltage control register	ADHVREFCNT
A/D conversion clock control register	ADCKS
Voltage reference control register	VREFAMPCNT

17.2.1 Peripheral enable registers 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the 12-bit A/D converter is to be used, be sure to set bit 5 (ADCEN) to 1.

The PER0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

ADCEN	Control over supply of the input clock for the 12-bit A/D converter and temperature sensor 2
0	Stops input clock supply. • SFR used by the 12-bit A/D converter and temperature sensor 2 cannot be written. The read value is 00H. However, the SFR is not initialized. [Note]
1	Enables input clock supply. • SFR used by the 12-bit A/D converter and temperature sensor 2 can be read and written.

Note To initialize the 12-bit A/D converter and the SFR used by the 12-bit A/D converter, use bit 5 (ADCRES) of PRR0.

Cautions 1. When setting the 12-bit A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1.

If ADCEN = 0, the values of the registers which control the 12-bit A/D converter are cleared to their initial values and writing to them is ignored.

- **A/D data register y (y = 0 to 5) (ADDRy)**
- **A/D temperature sensor data register (ADTSDR)**
- **A/D internal reference voltage data register (ADOCDR)**
- **A/D self-diagnosis data register (ADRD)**
- **A/D control register (ADCSR)**
- **A/D channel select register A0 (ADANSA0)**
- **A/D-converted value addition/average function channel select register 0 (ADADS0)**
- **A/D-converted value addition/average count select register (ADADC)**
- **A/D control extended register (ADCER)**
- **A/D conversion start trigger select register (ADSTRGR)**
- **A/D conversion extended input control register (ADEXICR)**
- **A/D sampling state register n (n = 0 to 5, T, O) (ADSSTRn)**
- **A/D sample-and-hold circuit control register (ADSHCR)**
- **A/D high-potential/low-potential reference voltage control register (ADHVREFCNT)**
- **A/D conversion clock control register (ADCKS)**
- **Voltage reference control register (VREFAMPCNT)**

2. Be sure to clear bit 7 to 0.

17.2.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the 12-bit A/D converter in the reset state, be sure to set bit 5 (ADCRES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17-3. Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

ADCRES	Control resetting of the 12-bit A/D converter and temperature sensor 2
0	The 12-bit A/D converter and temperature sensor 2 are released from the reset state.
1	The 12-bit A/D converter and temperature sensor 2 are in the reset state.

Caution Wait for 1 μs or longer to start A/D conversion after release from the reset state.

17.2.3 A/D data registers y (ADDRy) A/D temperature sensor data register (ADTSDR) A/D internal reference voltage data register (ADOCDR)

ADDRy (y = 0 to 5) are 16-bit read-only registers which store the A/D conversion results.
The ADDR0 to ADDR5 registers correspond to the ANI0 to ANI5 pin input voltage.
ADTSDR is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.
ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage (V_{BGR}).

The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.
  A/D-converted value average mode can be set only when twice or four times is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (A/D-converted value addition mode and 1-time to 4-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (A/D-converted value addition mode and 16-time conversion selected)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

Figure 17-4. Format of A/D Data Register y (ADDRy), A/D Temperature Sensor Data Register (ADTSDR), and A/D Internal Reference Voltage Data Register (ADOCDR)

Address: ADDR0: F0620h, ADDR1: F0622h, ADDR2: F0624h, ADDR3: F0626h, ADDR4: F0628h, ADDR5: F062Ah,
ADTSDR: F061Ah, ADOCDR: F061Ch

After reset: 0000h

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRy																
ADTSDR																
ADOCDR																

Remark y = 0 to 5

17.2.4 A/D self-diagnosis data register (ADRD)

Figure 17-5. Format of A/D Self-diagnosis Data Register (ADRD)

Address: F061Eh After reset: 0000h

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADRD																

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter’s self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see **17.2.9 A/D control extended register (ADCER)**.

The data formats for each given condition are shown below.

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

Table 17-5. Self-diagnosis Status Description [Note]

Bits 15 and 14 for Flush-right Format Setting Bits 1 and 0 for Flush-left Format Setting	Self-diagnosis Status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of reference power supply × 1/2 has been executed.
11b	Self-diagnosis using the voltage of reference power supply has been executed.

Note For details of self-diagnosis, see **17.2.9 A/D control extended register (ADCER)**.

17.2.5 A/D control register (ADCSR)

The ADCSR register sets A/D conversion start trigger, enables/disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

This register can be set by 16-bit memory manipulation instructions.

Figure 17-6. Format of A/D Control Register (ADCSR) (1/2)

Address: F0600h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCSR	ADST	ADCS[1:0]		ADIE	0	ADHSC	TRGE	0	0	0	0	0	0	0	0	0

ADST	A/D conversion start bit
0	Stops A/D conversion process.
1	Starts A/D conversion process.

The ADST bit starts or stops A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.

[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.TRGE bit being set to 1.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, the temperature sensor output, or the internal reference voltage (V_{BGR}) is completed in single scan mode.

ADCS[1:0]	Scan mode select bit
00	Single scan mode
01	Setting prohibited
10	Continuous scan mode
11	Setting prohibited

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 6 channels selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.

In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 6 channels selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel.

If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.

When selecting the temperature sensor output or internal reference voltage (V_{BGR}), select single scan mode, and deselect all the channels selected with the ADANSA0 register before performing A/D conversion. When A/D conversion of the selected temperature sensor output or internal reference voltage (V_{BGR}) is completed, A/D conversion is stopped.

The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Figure 17-6. Format of A/D Control Register (ADCSR) (2/2)

Address: F0600h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCSR	ADST	ADCS[1:0]		ADIE	0	ADHSC	TRGE	0	0	0	0	0	0	0	0	0

ADIE	Scan end interrupt enable bit
0	Disables INTAD interrupt generation upon scan completion.
1	Enables INTAD interrupt generation upon scan completion.
The ADIE bit enables or disables the A/D scan end interrupt (INTAD) in scans.	

ADHSC	A/D conversion select bit
0	High-speed conversion
1	Normal conversion
The ADHSC bit sets the operating mode of A/D conversion. Note that high-speed conversion is not available when V_{BGR} is selected. When modifying this bit, set the 12-bit converter to the standby state. For the procedure for modifying the ADHSC bit, see **17.8.9 ADHSC bit rewriting procedure**.	

TRGE	Trigger start enable bit
0	Disables A/D conversion to be started by trigger.
1	Enables A/D conversion to be started by trigger.
The TRGE bit enables or disables A/D conversion by the synchronous trigger.	

17.2.6 A/D channel select register A0 (ADANSA0)

The ADANSA0 register selects analog input channels for A/D conversion among ANI0 to ANI5.

This register can be set by 16-bit memory manipulation instructions.

Figure 17-7. Format of A/D Channel Select Register A0 (ADANSA0)

Address: F0604h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADANSA0	0	0	0	0	0	0	0	0	0	0	ANSA0[5:0]					

ANSA0[5:0]	A/D conversion channel select bit
0	ANI0 to ANI5 are not subjected to conversion.
1	ANI0 to ANI5 are subjected to conversion.
The ANSA0[5:0] bits select analog input channels for A/D conversion among ANI0 to ANI5. The channels to be selected and the number of channels can be arbitrarily set. The ANSA0[0] bit corresponds to ANI0 and the ANSA0[5] bit corresponds to ANI5. When performing A/D conversion of the temperature sensor output or internal reference voltage (V_{BGR}), do not select analog input channels. The setting value of this register should be 0000h. The ANSA0[5:0] bits should be set while the ADCSR.ADST bit is 0.	

17.2.7 A/D-converted value addition/average function channel select register 0 (ADADS0)

The ADADS0 register selects the channels 0 to 5 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

This register can be set by 16-bit memory manipulation instructions.

Figure 17-8. Format of A/D-converted Value Addition/Average Function Channel Select Register 0 (ADADS0)

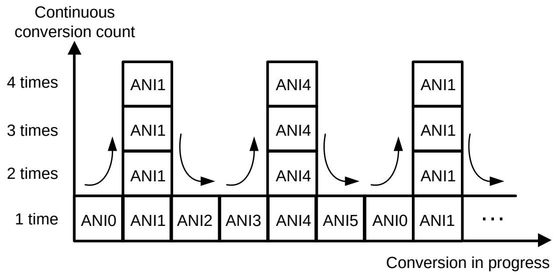
Address: F0608h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADADS0	0	0	0	0	0	0	0	0	0	0	ADS0[5:0]					

ADS0[5:0]	A/D-converted value addition/average channel select
0	A/D-converted value addition/average mode for ANI0 to ANI5 is not selected.
1	A/D-converted value addition/average mode for ANI0 to ANI5 is selected.
When the ADS0[n] bit of the number that is the same as that of A/D-converted channel selected by the ANSA0[n] bits (n = 0 to 5) in ADANSA0 is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register. The ADS0[5:0] bits should be set while the ADCSR.ADST bit is 0.	

Figure 17-9 shows a scanning operation sequence in which both the ADS0[1] and ADS0[4] bits are set to 1. It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011b), and the channels ANI0 to ANI5 are selected (ADANSA0.ANSA0[5:0] = 3Fh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with ANI0. The ANI1 conversion is performed successively four times (addition three times), and the added (integrated) value is returned to A/D data register 1. After that the ANI2 conversion is started. The ANI4 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 4. After conversion of ANI5, the conversion operation is once again performed in the same sequence from ANI0.

Figure 17-9. Scan Conversion Sequence with ADADC.ADC[2:0] = 011b, ADADC.AVEE = 0, ADS0[1] = 1, ADS0[4] = 1



17.2.8 A/D-converted value addition/average count select register (ADADC)

The ADADC register sets the addition count for A/D conversion of the channel, temperature sensor output, or internal reference voltage (V_{BGR}) for which A/D-converted value addition or average mode is selected, and selects either addition or average mode.

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

Figure 17-10. Format of A/D-converted Value Addition/Average Count Select Register (ADADC)

Address: F060Ch After reset: 0000h R/W

Symbol	<7>	6	5	4	3	2	1	0
ADADC	AVEE	0	0	0	0	ADC[2:0]		

AVEE	Average mode enable bit
0	Addition mode is selected.
1	Average mode is selected.
The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D conversion and A/D-converted value addition/average mode is selected, temperature sensor output, and internal reference voltage (V_{BGR}). When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC = 000b), three times (ADADC.ADC = 010b), or 16 times (ADADC.ADC = 101b). The mean value of 1- time, 3-time, and 16-time conversion cannot be obtained. The AVEE bit should be set while the ADCSR.ADST bit is 0.	

ADC[2:0]	Addition count select bit
000	1-time conversion (no addition; same as normal conversion) Note
001	2-time conversion (addition once)
010	3-time conversion (addition twice) Note
011	4-time conversion (addition three times)
101	16-time conversion (addition 15 times) Note
Other than above	Setting prohibited
The ADC[2:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, and to A/D conversion of temperature sensor output and internal reference voltage (V_{BGR}). When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.	

Note Only 2-time or 4-time conversion is selected when the AVEE bit is 1. When average mode is selected (ADADC.AVEE bit = 1), do not set 1-time conversion (ADADC.ADC[2:0] = 000b), 3-time conversion (ADADC.ADC[2:0] = 010b), or 16-time conversion (ADADC.ADC[2:0] = 101b).

17.2.9 A/D control extended register (ADCER)

The ADCER register sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

This register can be set by 16-bit memory manipulation instructions.

Figure 17-11. Format of A/D Control Extended Register (ADCER) (1/2)

Address: F060Eh After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCER	ADRFMT	0	0	0	DIAGM	DIAGLD	DIAGVAL[1:0]		0	0	ACE	0	0	0	0	0

ADRFMT	A/D data register format select bit
0	Flush-right is selected for the A/D data register format.
1	Flush-left is selected for the A/D data register format.
The ADRFMT bit specifies flush-right or flush-left for the data to be stored in the ADDRy, ADRD, ADTSDR, or ADOCDR register. The ADRFMT bit should be set while the ADST bit is 0.	

DIAGM	Self-diagnosis enable bit
0	Disables self-diagnosis of 12-bit A/D converter.
1	Enables self-diagnosis of 12-bit A/D converter.
The DIAGM bit enables or disables self-diagnosis. Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. The DIAGM bit should be set while the ADST bit is 0.	

DIAGLD	Self-diagnosis mode select bit
0	Rotation mode for self-diagnosis voltage
1	Fixed mode for self-diagnosis voltage
The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value. The DIAGLD bit should be set while the ADST bit is 0.	

Figure 17-11. Format of A/D Control Extended Register (ADCER) (2/2)

Address: F060Eh After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCER	ADRFMT	0	0	0	DIAGM	DIAGLD	DIAGVAL[1:0]		0	0	ACE	0	0	0	0	0

DIAGVAL [1:0]	Self-diagnosis conversion voltage select bit
00	Setting prohibited in self-diagnosis voltage fixed mode
01	Uses the voltage of 0 V for self-diagnosis.
10	Uses the voltage of reference power supply × 1/2 for self-diagnosis.
11	Uses the voltage of reference power supply for self-diagnosis.
These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit. Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.	

ACE	A/D data register automatic clearing enable bit
0	Disables automatic clearing.
1	Enables automatic clearing.
The ACE bit enables or disables automatic clearing (all “0”) of ADDRy, ADRD, ADTSDR, or ADOCDR after any of these registers have been read by the CPU or DTC. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.	

For details on the format of each data register, see **17.2.3 A/D data registers y (ADDRy) A/D temperature sensor data register (ADTSDR) A/D internal reference voltage data register (ADOCDR)** and **17.2.4 A/D self-diagnosis data register (ADRD)**.

17.2.10 A/D conversion start trigger select register (ADSTRGR)

The ADSTRGR register selects the A/D conversion start trigger.
This register can be set by 16-bit memory manipulation instructions.

Figure 17-12. Format of A/D Conversion Start Trigger Select Register (ADSTRGR)

Address: F0610h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADSTRGR	0	0	TRSA[5:0]						0	0	0	0	0	0	0	0

<table>
<tr><th>TRSA[5:0]</th><th>A/D conversion start trigger select bit [Note]</th></tr>
<tr><td>110000</td><td>Event output signal from event link controller (ELCTRG0)</td></tr>
<tr><td>111111</td><td>Trigger source deselection</td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td></tr>
<tr><td colspan="2">The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode.
• When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1.
• Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit and the TRSA[5:0] bits.</td></tr>
</table>

Note The issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. See **17.3.4 Analog input sampling time and scan conversion time** for details.

Table 17-6 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 17-6. Selection of A/D Activation Sources by the TRSA[5:0] Bits

Peripheral Function	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			1	1	1	1	1	1
ELC	ELCTRG0	Event output signal from the event link controller	1	1	0	0	0	0

17.2.11 A/D conversion extended input control register (ADEXICR)

The ADEXICR register specifies the settings of A/D conversion of the temperature sensor output and internal reference voltage (V_{BGR}).

This register can be set by 16-bit memory manipulation instructions.

Figure 17-13. Format of A/D Conversion Extended Input Control Register (ADEXICR) (1/2)

Address: F0612h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADEXICR	0	0	0	0	0	0	OCSA	TSSA	0	0	0	0	0	0	OCSAD	TSSAD

OCSA	Internal reference voltage (V_{BGR}) A/D conversion select bit
0	A/D conversion of internal reference voltage (V_{BGR}) is not performed.
1	A/D conversion of internal reference voltage (V_{BGR}) is performed.
This bit selects A/D conversion of the internal reference voltage (V_{BGR}) in single scan mode. When A/D conversion of the internal reference voltage (V_{BGR}) is to be performed, set all the bits in the ADANSA0 register and the TSSA bit should be set to all 0 in single scan mode. The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage (V_{BGR}), discharge the A/D converter before sampling. The sampling time should be 5 μs or longer. Sampling starts after discharging is completed during A/D conversion of the internal reference voltage (V_{BGR}), so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.	

TSSA	Temperature sensor output A/D conversion select bit
0	A/D conversion of temperature sensor output is not performed.
1	A/D conversion of temperature sensor output is performed.
This bit selects A/D conversion of the temperature sensor output in single scan mode. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANSA0 register and the OCSA bit should all be set to 0 in single scan mode. The TSSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the temperature sensor output, discharge the A/D converter before sampling. The sampling time should be 5 μs or longer. Sampling starts after discharging is completed during A/D conversion of the temperature sensor output, an auto-discharging period of 15 ADCLK cycles is inserted before sampling.	

OCSAD	Internal reference voltage (V_{BGR}) A/D-converted value addition/average mode select bit
0	Internal reference voltage (V_{BGR}) A/D-converted value addition/average mode is not selected.
1	Internal reference voltage (V_{BGR}) A/D-converted value addition/average mode is selected.
When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage (V_{BGR}) is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR. The OCSAD bit should be set while the ADCSR.ADST bit is 0.	

Figure 17-13. Format of A/D Conversion Extended Input Control Register (ADEXICR) (2/2)

Address: F0612h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADEXICR	0	0	0	0	0	0	OCSA	TSSA	0	0	0	0	0	0	OCSAD	TSSAD

TSSAD	Temperature sensor output A/D-converted value addition/average mode select
0	Temperature sensor output A/D-converted value addition/average mode is not selected.
1	Temperature sensor output A/D-converted value addition/average mode is selected.
When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D temperature sensor data register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is stored in the A/D temperature sensor data register (ADTSDR). The TSSAD bit should be set while the ADCSR.ADST bit is 0.	

17.2.12 A/D sampling state register n (ADSSTRn) (n = 0 to 5, T, O)

The ADSSTRn register sets the sampling time for analog input.

If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 24 MHz, one state is 41.67 ns.

The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. Set a value greater than 05h in this register.

This register can be set by 8-bit memory manipulation instructions.

Figure 17-14. Format of A/D Sampling State Register n (ADSSTRn) (n = 0 to 5, T, O)

Address: ADSSTRT: F06DEh, ADSSTRO: F06DFh, ADSSTR0: F06E0h, ADSSTR1: F06E1h, ADSSTR2: F06E2h,
ADSSTR3: F06E3h, ADSSTR4: F06E4h, ADSSTR5: F06E5h

After reset: 0Dh R/W

Symbol	7	6	5	4	3	2	1	0
ADSSTRn								

Table 17-7 shows the relationship between the A/D sampling state register and the relevant channels.

For details, refer to **17.3.4 Analog input sampling time and scan conversion time**.

Table 17-7. Relationship between A/D Sampling State Register and Relevant Channels

Register Name	Channels
ADSSTR0	ANI0 [Note 2]
ADSSTR1	ANI1 [Note 2]
ADSSTR2	ANI2 [Note 2]
ADSSTR3	ANI3 [Note 2]
ADSSTR4	ANI4 [Note 2]
ADSSTR5	ANI5 [Note 2]
ADSSTRT	Temperature sensor output [Notes 1, 2]
ADSSTRO	Internal reference voltage (V_{BGR}) [Notes 1, 2]

Notes 1. When performing A/D conversion of the temperature sensor output or internal reference voltage (V_{BGR}), the sampling time should be 5 μs or longer.

2. To perform the A/D conversion of the analog input channel, the sampling time must be set to 1.26 μs or longer.

17.2.13 A/D sample-and-hold circuit control register (ADSHCR)

The ADSHCR register is used to control the channel-dedicated sample-and-hold circuits.
This register can be set by 16-bit memory manipulation instructions.

Figure 17-15. Format of A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address: F0666h After reset: 0026h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADSHCR	0	0	0	0	0	SHANS[2:0]			SSTSH[7:0]							

SSTSH[7:0]	Channel-dedicated sample-and-hold circuit sampling time setting bit
04H … FFH	Set the sampling time (4 to 255 states).
Other than above	Setting prohibited

SHANS[0]	ANI0 channel-dedicated sample-and-hold circuit bypass select bit
0	Bypass the channel-dedicated sample-and-hold circuits.
1	Use the channel-dedicated sample-and-hold circuits.

SHANS[1]	ANI1 channel-dedicated sample-and-hold circuit bypass select bit
0	Bypass the channel-dedicated sample-and-hold circuits.
1	Use the channel-dedicated sample-and-hold circuits.

SHANS[2]	ANI2 channel-dedicated sample-and-hold circuit bypass select bit
0	Bypass the channel-dedicated sample-and-hold circuits.
1	Use the channel-dedicated sample-and-hold circuits.

17.2.14 A/D high-potential/low-potential reference voltage control register (ADHVREFCNT)

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

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

Figure 17-16. Format of A/D High-potential/Low-potential Reference Voltage Control Register (ADHVREFCNT)

Address: F068Ah After reset: 0000h R/W

Symbol	<7>	6	5	4	3	2	1	0
ADHVREFCNT	ADSLP	0	0	LVSEL	0	0	HVSEL[1:0]	

ADSLP	Sleep bit
0	Normal operation
1	Standby state
This bit is used to transition the 12-bit A/D converter to the standby state. Set the ADSLP bit to 1 only when modifying the ADCSR.ADHSC bit. In other cases, setting the ADSLP bit to 1 is prohibited. After the ADSLP bit is set to 1, wait at least 5 μs before clearing this bit to 0. Furthermore, after the ADSLP bit is cleared to 0, wait at least 1 μs and then start the A/D conversion. For the ADHSC bit rewriting procedure, see **17.8.9 ADHSC bit rewriting procedure**.	

LVSEL	Low-potential reference voltage select bit
0	AV_{SS1} is selected as the low-potential reference voltage.
1	AV_{REFM} is selected as the low-potential reference voltage.
This bit is used to set the low-potential reference voltage. AV_{SS1} or AV_{REFM} is selectable as the low-potential reference voltage.	

HVSEL[1:0]	High-potential reference voltage select bit
00	AV_{DD} is selected as the high-potential reference voltage.
01	AV_{REFP} or VREFOUT is selected as the high-potential reference voltage [Note].
10	Setting prohibited
11	Discharges the internal reference voltage (the high-potential reference voltage is not selected).
These bits are used to set the high-potential reference voltage. AV_{DD} or AV_{REFP}/VREFOUT is selectable as the high-potential reference voltage. To select the voltage reference output (VREFOUT) (HVSEL = 01b and VREFEN = 1), discharge the voltage source (HVSEL = 11b) first.	

Note VREFOUT is selected when the voltage reference is operated, and AV_{REFP} is selected other than when the voltage reference is operated. For the procedure to the start voltage reference operation, refer to **17.6 Selecting Reference Voltage**.

17.2.15 A/D conversion clock control register (ADCKS)

The ADCKS register sets the divisor for the A/D conversion clock (ADCLK) and peripheral hardware clock (PCLK). Set this register before starting A/D conversion.

This register can be set by 8-bit memory manipulation instructions.

Figure 17-17. Format of A/D Conversion Clock Control Register (ADCKS)

Address: F0079h After reset: 0000h R/W

Symbol	7	6	5	4	3	2	1	0
ADCKS	0	0	0	0	0	0	ADCKS[1:0]	

ADCKS[1:0]	A/D conversion clock select bit
00	System clock not divided (f1)
01	System clock divided by 2 (f2)
10	System clock divided by 4 (f4)
11	System clock divided by 8 (f8)

Caution Settings such that the frequency of the A/D conversion clock is below 1 MHz are prohibited.

17.2.16 Voltage reference control register (VREFAMPCNT)

The VREFAMPCNT register is used to control the voltage reference (VREFADC) and output voltage (VREFOUT).
This register can be set by 8-bit memory manipulation instructions.
Reset signal generation clears this register to 00H.

Figure 17-18. Format of Voltage Reference Control Register (VREFAMPCNT)

Address: F0075H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
VREFAMPCNT	0	0	0	BGREN	VREFADCEN	VREFADCG[1:0]		OLDETEN

OLDETEN	OLDET enable
0	Disable detection of overcurrents.
1	Enable detection of overcurrents.

VREFADCG1	VREFADCG0	VREFOUT output voltage control
0	x	1.5 V
1	0	2.0 V
1	1	2.5 V

x: Don’t care

VREFADCEN	VREFADCG enable
0	Disable the voltage reference output and select AV_{REFP} as the high-potential reference voltage.
1	Enable the voltage reference output and select VREFOUT as the high-potential reference voltage.

BGREN	BGR enable
0	The power supply for the BGR block is turned off.
1	The power supply for the BGR block is turned on.

The voltage reference output voltage is output to the VREFOUT pin. When the voltage reference (VREFADC) is in use, do not apply a voltage to the AV_{REFP}/VREFOUT pin. To stabilize the voltage reference output, connect the AV_{REFP}/VREFOUT pin to the AV_{REFM} pin via a capacitor (1 μF).

As shown in **Table 17-8**, the voltage reference output is controlled by the combination of the BGREN, VREFADCEN, VREFADCG[1:0], and OLDETEN bits.

Table 17-8. List of Voltage Reference Output Voltage Control State

State	BGREN	VREFADCEN	VREFADCG[1]	VREFADCG[0]	OLDETEN	Output voltage (VREFOUT)
After release from the reset state	0	0	0	0	0	Hi-Z
Only the power supply for the BGR block is turned on	1	0	x	x	x	Hi-Z
VREFOUT 1.5 V output	1	1	0	x	0 or 1	1.5 V
VREFOUT 2.0 V output	1	1	1	0	0 or 1	2.0 V
VREFOUT 2.5 V output	1	1	1	1	0 or 1	2.5 V

x: Don’t care

17.3 Operation

17.3.1 Scanning operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

A scan conversion is performed in two operating modes: single scan mode and continuous scan mode.

Scan conversion is in either of two conversion modes: high-speed and normal.

In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software.

In single scan mode and continuous scan mode, A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.

When any of ANI0 to ANI2 channels is set as a channel-dedicated sample-and-hold circuit by the S12AD1.ADSHCR.SHANS[2:0] bits, the target analog input is sampled and held before the first A/D conversion of each scan.

For A/D conversion of the temperature sensor output or internal reference voltage (V_{BGR}), proceed with conversion in the single scan mode and without selecting any other channels.

Caution While the ADCSR.ADST bit is “1” (scanning is in progress), the software trigger and synchronization trigger input that are the start conditions for A/D conversion are ineffective regardless of the scan mode.

17.3.2 Single scan mode

17.3.2.1 Basic operation (without channel-dedicated sample-and-hold circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

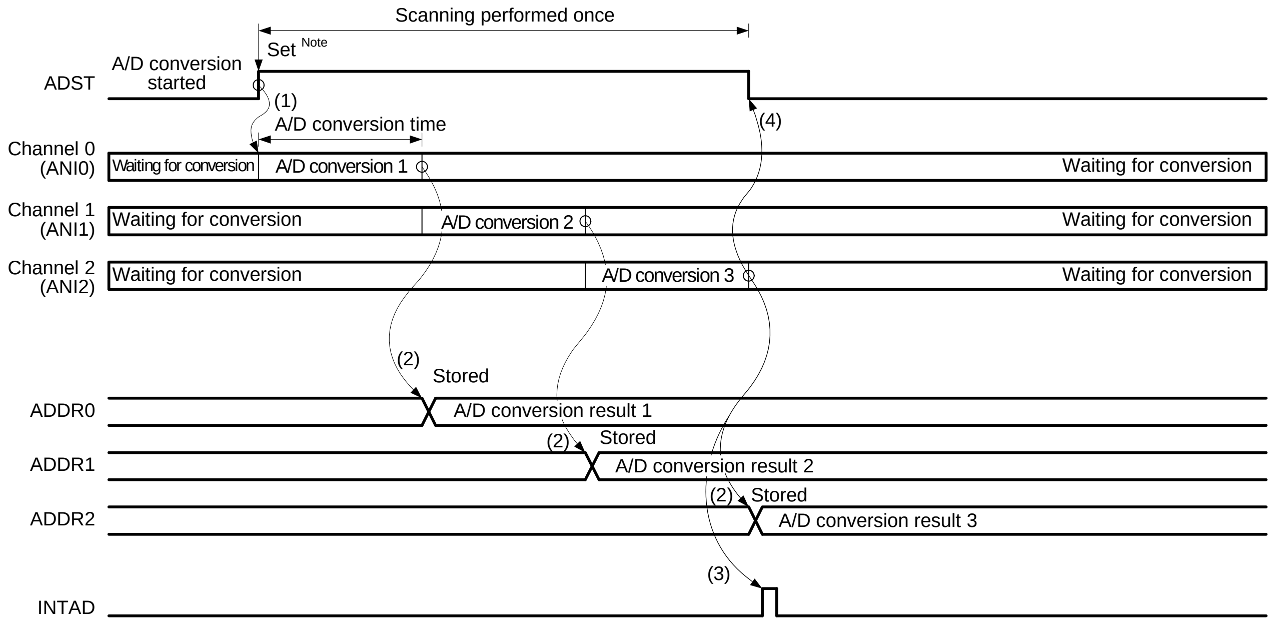
(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input, A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

(2) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled).

(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

Figure 17-19. Example of Operation in Single Scan Mode (Basic Operation: ANI0, ANI1, and ANI2 Selected)



Note ↓indicates the instruction is executed by software.

17.3.2.2 Basic operation (with channel-dedicated sample-and-hold circuits)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.

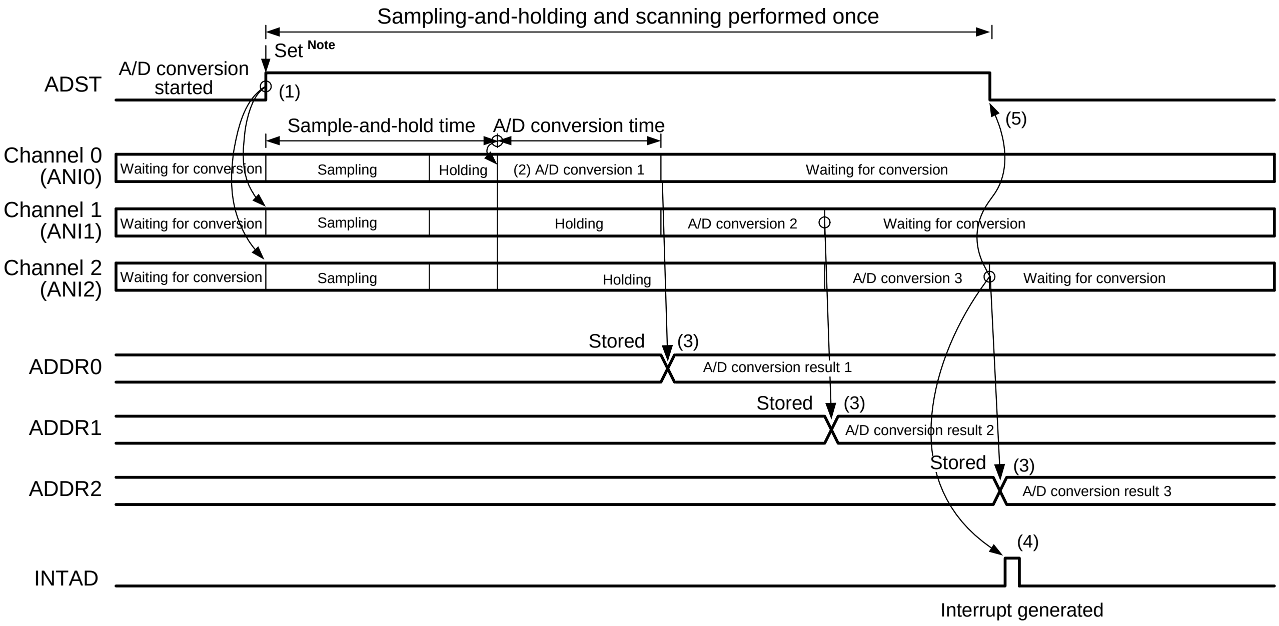
(2) After sample-and-hold operation, A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(5) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

Figure 17-20. Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used, ANI0, ANI1, and ANI2 Selected)



Note ↓ indicates the instruction is executed by software.

17.3.2.3 Channel selection and self-diagnosis (without channel-dedicated sample-and-hold circuits)

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.

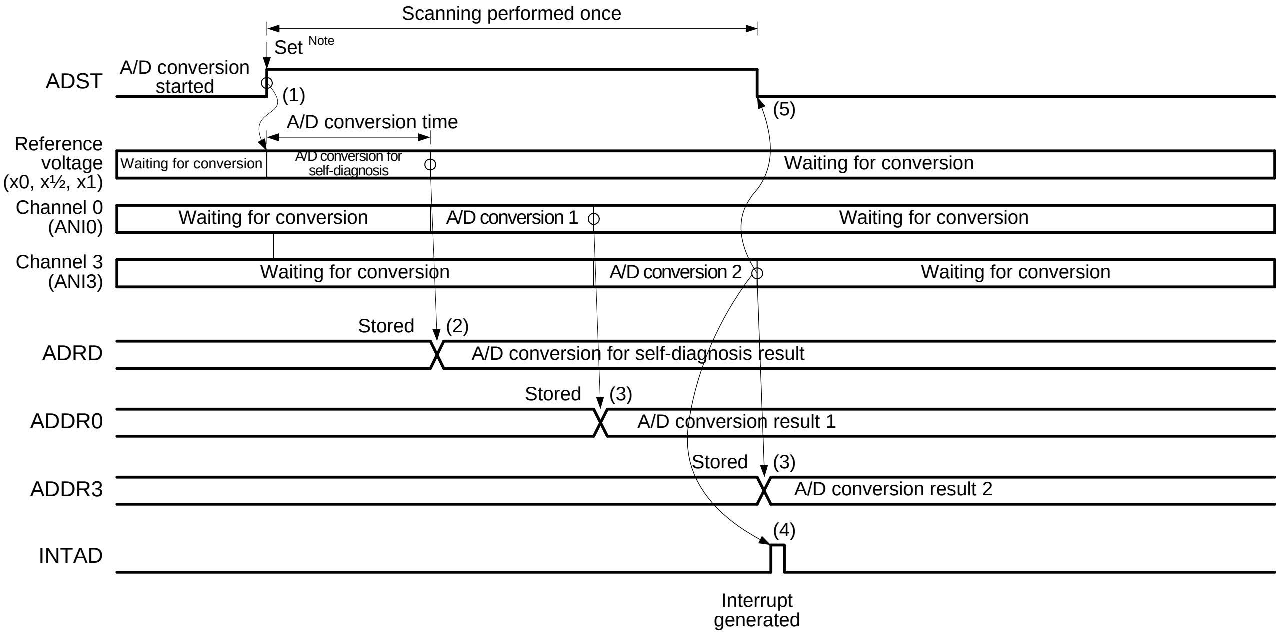
(2) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(4) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled).

(5) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

Figure 17-21. Example of Operation in Single Scan Mode (Basic Operation: ANI0 and ANI3 Selected + Self-Diagnosis)



Note ↓ indicates the instruction is executed by software.

17.3.2.4 Channel selection and self-diagnosis (with channel-dedicated sample-and-hold circuits)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.

(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.

(3) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

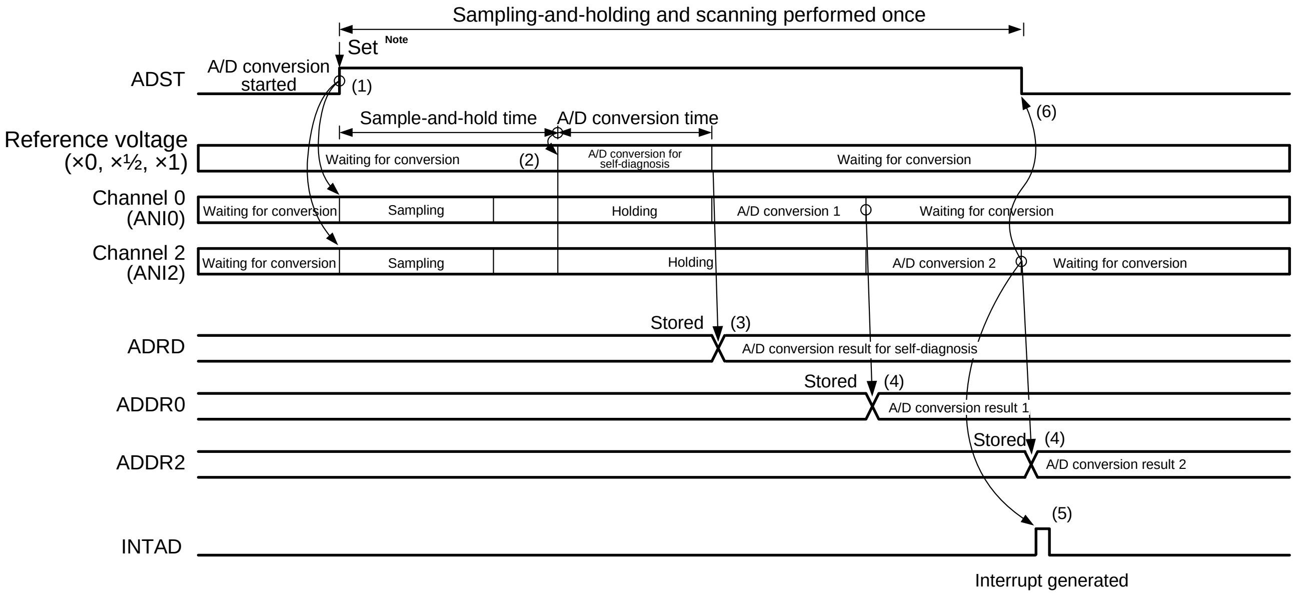
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(6) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

Figure 17-22. Example of Operation in Continuous Scan Mode

(Channel-Dedicated Sample-and-Hold Circuits Used, ANI0 and ANI2 Selected + Self-Diagnosis)



Note ↓ indicates the instruction is executed by software.

17.3.2.5 A/D conversion of temperature sensor output/internal reference voltage (V_{BGR})

A/D conversion of the temperature sensor output and internal reference voltage (V_{BGR}) is performed in single scan mode as below.

All channels should be deselected (by setting the ADANSA0 register bit to all 0). When selecting A/D conversion of the temperature sensor output, the A/D conversion select bit for the internal reference voltage (V_{BGR}) (ADEXICR.OCSA) should be set to 0 (deselected). When selecting A/D conversion of the internal reference voltage (V_{BGR}), the A/D conversion select bit for the temperature sensor output (ADEXICR.TSSA) should be set to 0 (deselected).

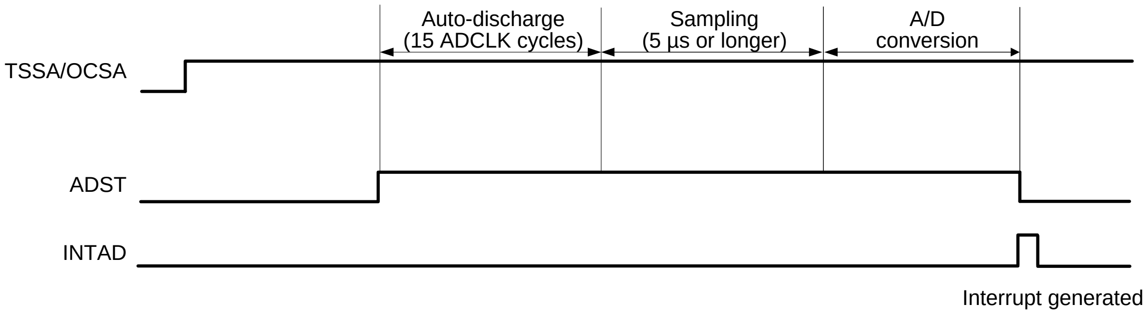
(1) Set the sampling time to 5 μs or longer.

(2) After switching to A/D conversion of the internal reference voltage (V_{BGR}) or the temperature sensor output, start A/D conversion by setting the ADST bit to 1.

(3) When A/D conversion is completed, the conversion result is stored into the corresponding A/D temperature sensor data register (ADTSDR) or A/D internal reference voltage data register (ADOCDR). If the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled), an INTAD interrupt request is generated.

(4) The ADST bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

Figure 17-23. Example of Operation in Single Scan Mode (Temperature Sensor Output or Internal Reference Voltage (V_{BGR}) Selected)



17.3.3 Continuous scan mode

17.3.3.1 Basic operation (without channel-dedicated sample-and-hold circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input, A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

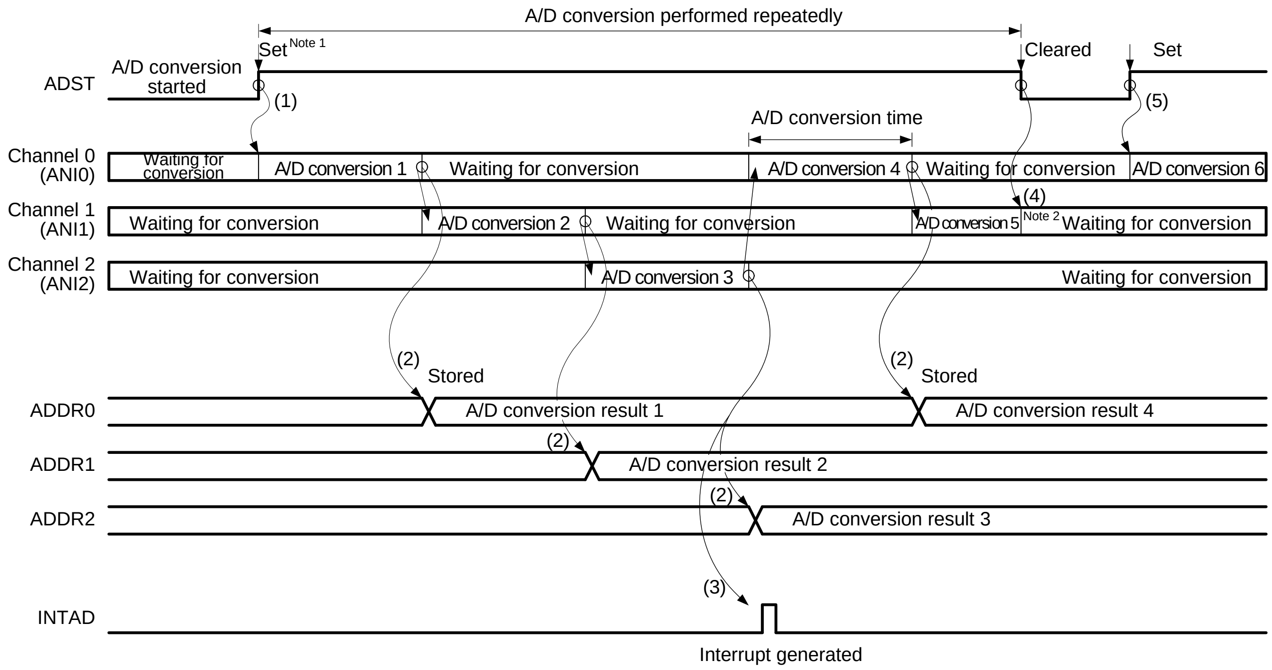
(2) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

(4) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

Figure 17-24. Example of Operation in Continuous Scan Mode (Basic Operation: ANI0, ANI1, and ANI2 Selected)



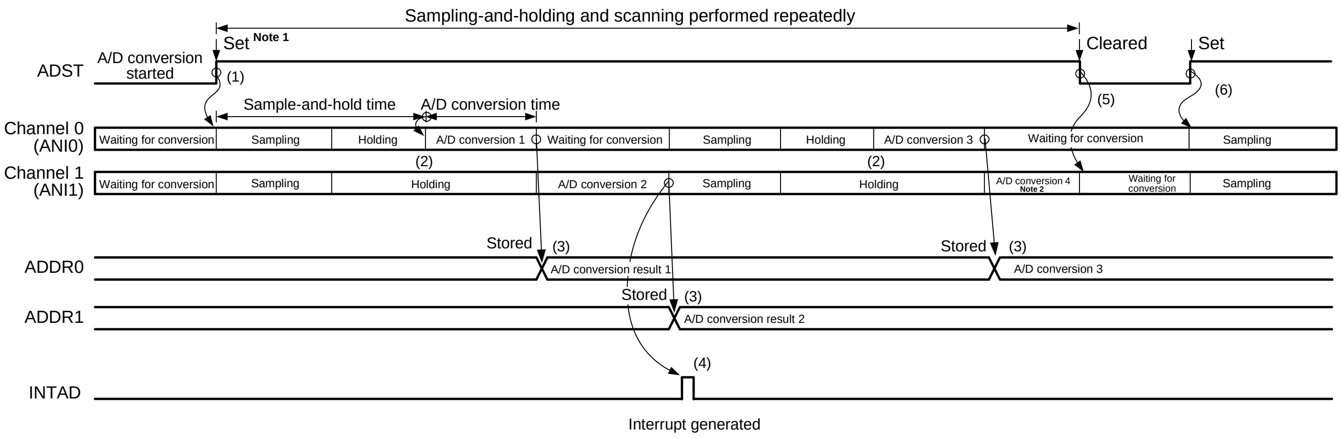
Notes 1. ↓ indicates the instruction is executed by software.
2. The converted data of A/D conversion 5 is ignored.

17.3.3.2 Basic operation (with channel-dedicated sample-and-hold circuits)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed repeatedly on the analog input of all the selected channels as below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:0] bits. In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Figure 17-25. Example of Operation in Continuous Scan Mode
(Channel-Dedicated Sample-and-Hold Circuits Used)



Notes 1. ↓ indicates the instruction is executed by software.
2. The converted data of A/D conversion 4 is ignored.

17.3.3.3 Channel selection and self-diagnosis (without channel-dedicated sample-and-hold circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage AV_{REFP} supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below. In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input, A/D conversion for self-diagnosis is started first.

(2) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

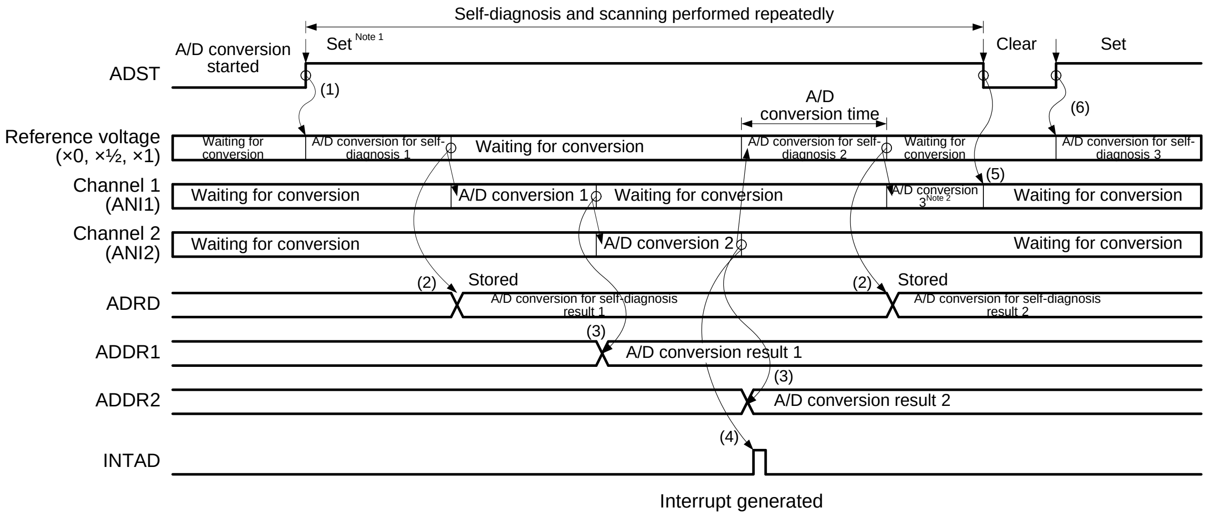
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).

(4) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.

(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(6) When the ADCSR.ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

Figure 17-26. Example of Operation in Continuous Scan Mode
(Basic Operation; ANI1 and ANI2 Selected + Self-Diagnosis)



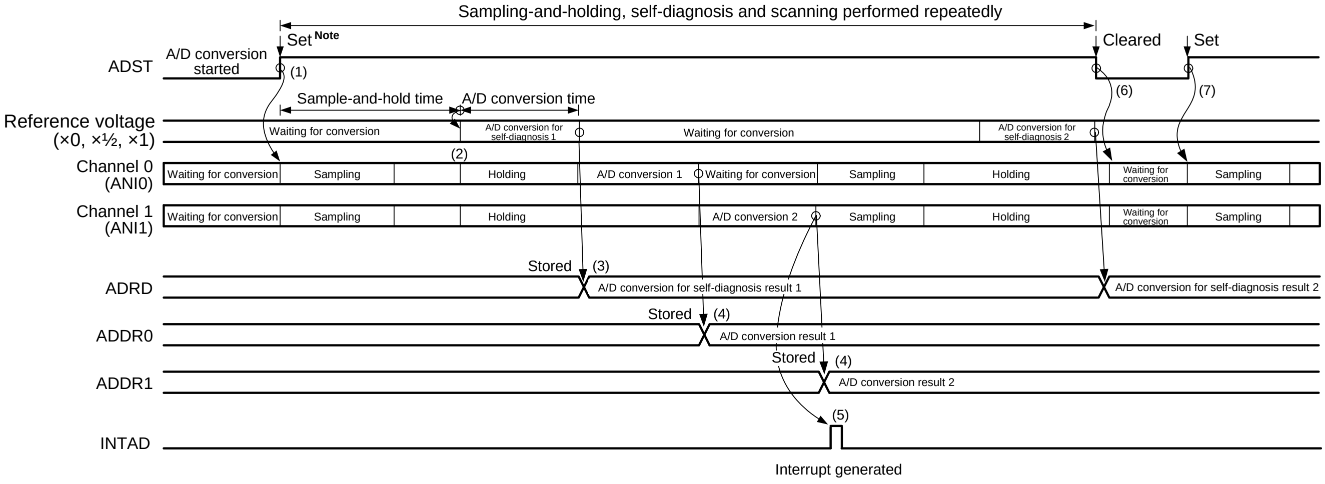
Notes 1. ↓ indicates the instruction is executed by software.
2. The converted data of A/D conversion 3 is ignored.

17.3.3.4 Channel selection and self-diagnosis (with channel-dedicated sample-and-hold circuits)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed repeatedly on the analog input of the selected channels. In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.
(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.
(3) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANIn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(6) The ADCSR.ADST bit is not automatically cleared and steps 2 to 5 are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(7) When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

Figure 17-27. Example of Operation in Continuous Scan Mode
(Channel-Dedicated Sample-and-Hold Circuits Used, ANI0 and ANI1 Selected + Self-Diagnosis)



Note ↓ indicates the instruction is executed by software.

17.3.4 Analog input sampling time and scan conversion time

Scan conversion can be activated either by software or synchronous trigger input. After the start-of-scanning-delay time (t_{D}) has elapsed, processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.

Figure 17-28 shows the timing of scan conversion in response to a software trigger or synchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_{D}), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH}) **Note 1**, self-diagnosis A/D conversion processing time (t_{DIAG}) **Note 2**, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED}) **Note 3**, and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states during high-speed conversion operation, and 41 ADCLK states during normal conversion operation. **Table 17-9** shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$t_{SCAN} = t_{D} + t_{SPLSH} + t_{DIAG} + (t_{CONV} \times n)$ **Note 4** $+ t_{ED}$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}.

The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $t_{SPLSH} + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)$ **Note 4** $+ t_{SHED}$

Notes 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.

2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

3. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).

4. $t_{CONV} \times n$ applies when the sampling time (t_{SPL}) for all selected channels is the same. When the sampling times of individual channels are not the same, it is the total of the sampling times for each of the channels and the time for conversion by successive approximation (t_{SAM}).

Table 17-9. Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK)

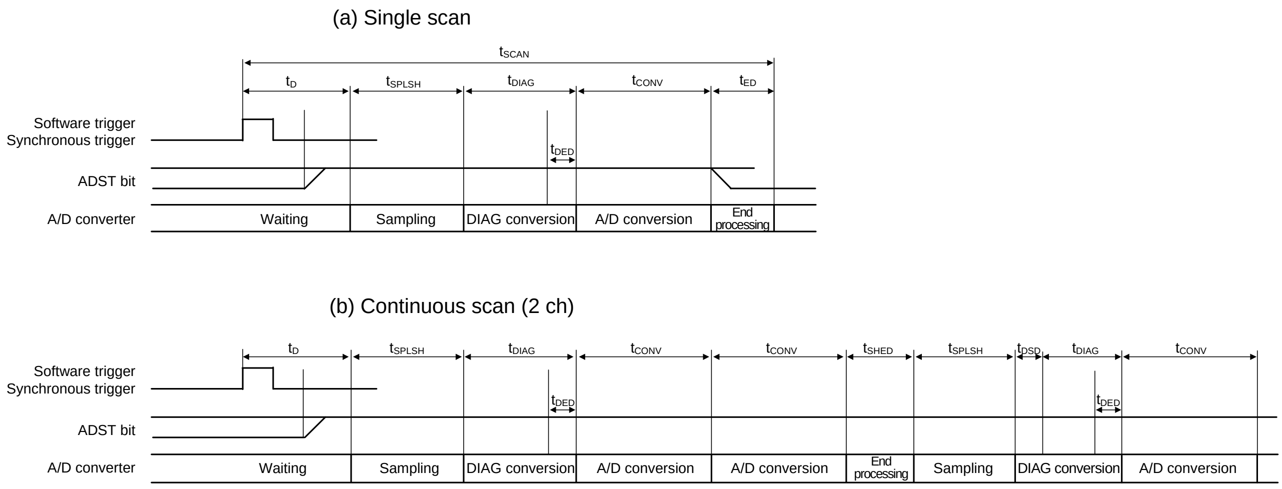
Item			Symbol		Type/Conditions		Unit
					Synchronous Trigger	Software Trigger	
Scan start processing time Notes 1, 2	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.	t_D		2 PCLK + 6 ADCLK	6 ADCLK	Cycle
	Other than above				2 PCLK + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time Note 1	Sampling time		t_{SPLSH}	t_{SH}	The setting of ADSHCR.SSTSH[7:0] (initial value = 1Ah) × ADCLK		
	Wait time between sampling and A/D conversion			t_W	13 ADCLK		
Self-diagnosis conversion processing time Note 1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Dh) × ADCLK Note 3		
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation)		
					41 ADCLK (during normal conversion operation)		
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.			t_{DED}	2 ADCLK		
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.			t_{DSD}	2 ADCLK		
A/D conversion processing time Note 1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 5, T, O) (initial value = 0Dh) × ADCLK Note 3		
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	32 ADCLK (during high-speed conversion operation)		
					41 ADCLK (during normal conversion operation)		
Channel-dedicated sample-and-hold end processing time			t_{SHED}		3 ADCLK		
Scan end processing time Note 1			t_{ED}		1 PCLK + 3 ADCLK		

Notes 1. For t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}, see **Figure 17-28**.

2. This is the maximum time required from software writing or trigger input to A/D conversion start.

3. The required sampling time (μs) is specified according to the voltage conditions. See **43.6.1 12-bit A/D converter characteristics**.

Figure 17-28. Scan Conversion Timing (Activated by Software or Synchronous Trigger)


(a) Single scan

(b) Continuous scan (2 ch)

17.3.5 Usage example of A/D data register automatic clearing function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR) to 0000h when the A/D data registers are read by the CPU or DTC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU or DTC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register.

Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

17.3.6 A/D-converted value addition/average mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. Using an average can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average mode can be specified when A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage (V_{BGR}) is selected.

17.3.7 Starting A/D conversion with synchronous trigger from peripheral function

The A/D conversion can be started by a synchronous trigger from the event link controller. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] bits.

17.4 Interrupt Sources and DTC Transfer Requests

17.4.1 Interrupt requests

The 12-bit A/D converter can send scan end interrupt requests INTAD to the CPU.

Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an INTAD interrupt, respectively.

In addition, the DTC can be activated when an INTAD interrupt is generated. Using an INTAD interrupt to allow the DTC to read the converted data enables continuous conversion without burden on software. For details on DTC settings, see **CHAPTER 25 DATA TRANSFER CONTROLLER (DTC)**.

17.5 Event Link Function

17.5.1 12-bit A/D converter operation by event from the ELC

The 12-bit A/D converter can be started by the predetermined event by setting ELSELRn of the ELC.

17.5.2 Note on 12-bit A/D converter when an event is input from the ELC

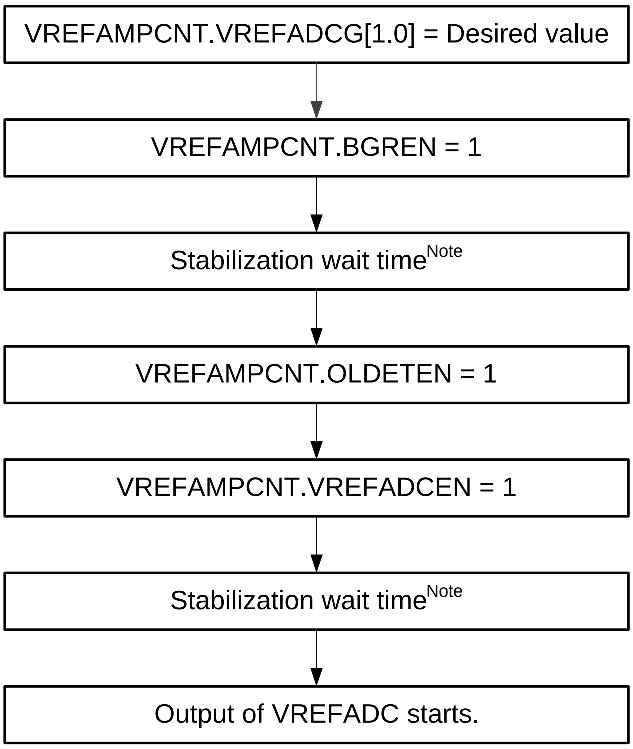
If an event occurs during A/D conversion, the event is disabled.

17.6 Selecting Reference Voltage

The high-potential reference voltage of the A/D converter can be selected from among the external pin input (AV_{REFP}), voltage reference voltage output (VREFOUT), and analog reference voltage (AV_{DD}). The low-potential reference voltage can be selected as either the external pin input (AV_{REFM}) or the analog reference voltage (AV_{SS1}). Make the settings before starting A/D conversion. For details of this setting, see **17.2.14 A/D high-potential/low-potential reference voltage control register (ADHVREFCNT)** and **17.2.16 Voltage reference control register (VREFAMPCNT)**.

Figure 17-29 shows the flow for starting the voltage reference (VREFADC).

Figure 17-29. Flowchart for Starting the Voltage Reference (VREFADC)


VREFAMPCNT.VREFADCG[1.0] = Desired value
VREFAMPCNT.BGREN = 1
Stabilization wait timeNote
VREFAMPCNT.OLDETEN = 1
VREFAMPCNT.VREFADCEN = 1
Stabilization wait timeNote
Output of VREFADC starts.

Note For details on the stabilization wait time, refer to **CHAPTER 43 ELECTRICAL SPECIFICATIONS**.

17.7 Allowable Impedance of Signal Source

To achieve high-speed conversion of 3 μs, the analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 0.5 kΩ or less. Since the load of the input pins creates an equivalent low-pass filter, the tracking of analog signals with large differential coefficients may become impossible.

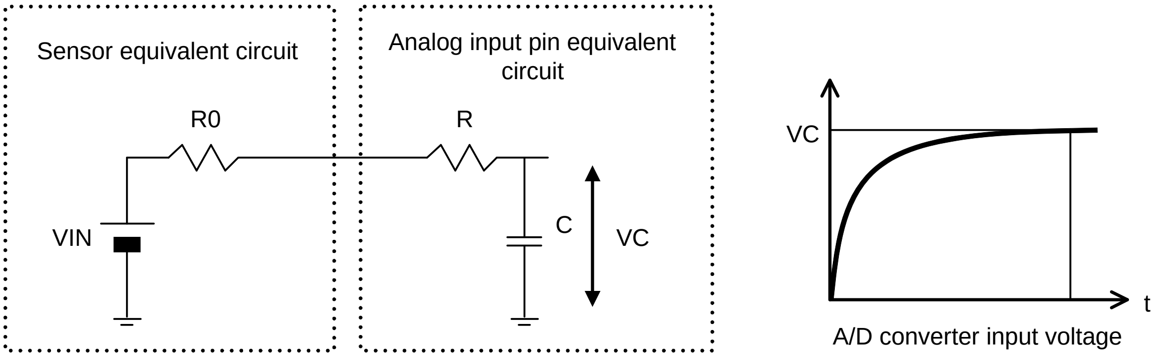
Insert a low-impedance buffer in cases of the conversion of high-speed analog signals or of multiple signals in scan mode and so on.

Furthermore, if conversion of the signal on a single pin is to proceed in single-scan mode, even in the case that an external capacitor with a large value is provided, switching of the analog multiplexer from input signal to input signal will affect the current.

Figure 17-30 shows an equivalent circuit of an analog input pin and an external sensor.

To perform A/D conversion accurately, charging of the internal capacitor C shown in **Figure 17-30** must be completed within the specified period of time. This specified period is referred to as sampling time.

Figure 17-30. Equivalent Circuit of Analog Input Pin and External Sensor


Sensor equivalent circuit
R0
VIN
Analog input pin equivalent circuit
R
C
VC
VC
t
A/D converter input voltage

17.8 Usage Notes

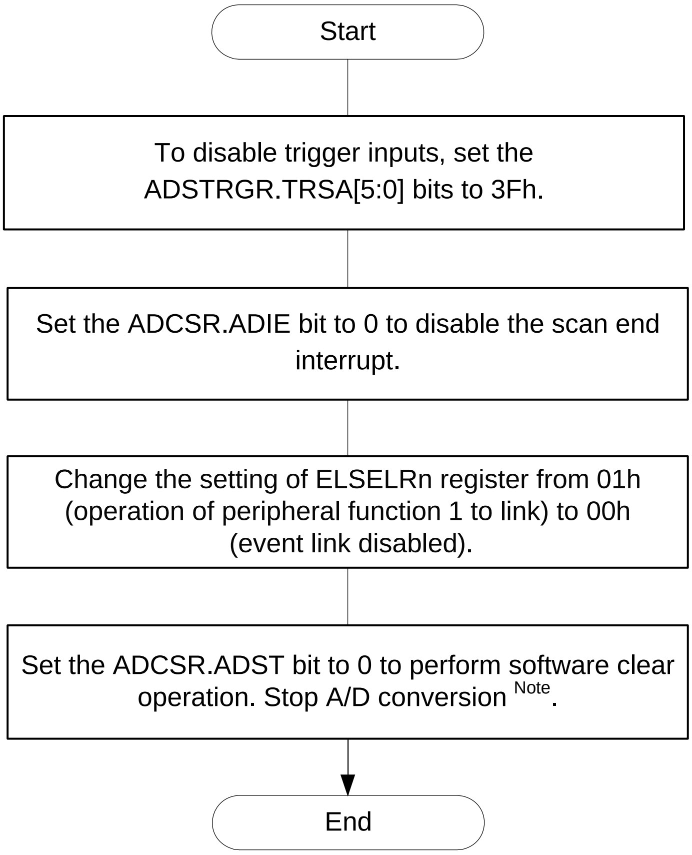
17.8.1 Notes on reading data registers

The A/D data registers, A/D temperature sensor data register, A/D internal reference voltage data register, and A/D self-diagnosis data register should be read in word units.

17.8.2 Procedure for stopping A/D conversion

To stop A/D conversion when a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in **Figure 17-31**.

Figure 17-31. Procedure for Clear Operation by Software through the ADCSR.ADST Bit


Start
To disable trigger inputs, set the ADSTRGR.TRSA[5:0] bits to 3Fh.
Set the ADCSR.ADIE bit to 0 to disable the scan end interrupt.
Change the setting of ELSELRn register from 01h (operation of peripheral function 1 to link) to 00h (event link disabled).
Set the ADCSR.ADST bit to 0 to perform software clear operation. Stop A/D conversion Note.
End

Note From software clearing to stopping of scanning takes two clock cycles of ADCLK.
Accordingly, wait for at least two clock cycles of ADCLK before making any of the following settings after software clearing.

- Enabling the scan end interrupt
- Selecting operation of peripheral function 1 to link
- Using software to starting A/D conversion
- Enabling trigger input

Remark n = 00 to 29

17.8.3 Point for caution on mode and status bits

Initialize or re-set the voltage state for self-diagnosis as necessary.

- Set ADCER.DIAGLD to 1 and select the voltage state for self-diagnosis by using ADCER.DIAGVAL[1:0] to re-set the voltage state for self-diagnosis.

17.8.4 A/D conversion restarting timing and termination timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

17.8.5 Point for caution on processing for two consecutive scan end interrupts

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

17.8.6 Clock supply stop function setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting peripheral enable register 0 (PER0).

The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the clock supply stop state.

After the clock supply stop state is released, wait for 1 μs to start A/D conversion.

17.8.7 Notes on entering low power consumption states

Be sure to stop A/D conversion before stopping supply of the clock signals or placing the chip in STOP mode. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.

Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in **Figure 17-31**.

After that, wait for three clock cycles of ADCLK before stopping supply of the clock signals or placing the chip in STOP mode.

17.8.8 Notes on canceling STOP mode

After release from STOP mode, wait for 1 μs before starting A/D conversion after the oscillation stabilization time has elapsed. For details on release from STOP mode, see **(2) Release from STOP mode** in **29.3.2 STOP mode**.

17.8.9 ADHSC bit rewriting procedure

Before rewriting the A/D conversion select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the 12-bit A/D converter must be in the standby state. Carry out steps 1 to 3 below to modify the ADCSR.ADHSC bit. After the sleep bit (ADHVREFCNT.ADSLP) is cleared to 0, wait for at least 1 μs and then start A/D conversion.

ADHSC Bit Rewriting Procedure:

1. Set the sleep bit (ADHVREFCNT.ADSLP) to 1.
2. Wait for at least 0.2 μs, and then modify the A/D conversion select bit (ADCSR.ADHSC).
3. Wait for at least 4.8 μs, and then clear the sleep bit (ADHVREFCNT.ADSLP) to 0.

Cautions 1. Only set the ADHVREFCNT.ADSLP bit to 1 when the value of the ADCSR.ADHSC bit is to be changed. In other cases, setting the ADSLP bit to 1 is prohibited.

2. Do not reset the sleep bit (ADHVREFCNT.ADSLP) while the A/D conversion select bit (ADCSR.ADHSC) is 1. After the A/D conversion select bit (ADCSR.ADHSC) is cleared to 0 or the operating mode is transitioned to clock supply stop mode, reset the sleep bit according to the ADHVREFCNT.ADSLP bit rewriting procedure.

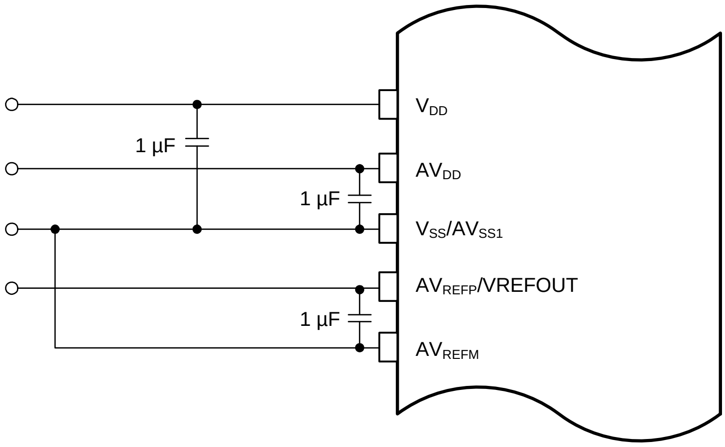
17.8.10 Voltage range of analog power supply pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins ANIn: $AV_{REFM} \leq V_{AI1} \leq AV_{REFP}$
  Reference voltage range applied to the AV_{REFP} pin: $AV_{REFP} \leq AV_{DD}$
  Voltage applied to analog input pins ANIn (n = 0 to 5): $AV_{SS1} \leq V_{AI1} \leq AV_{DD}$

- Relationship between power supply pin pairs ($AV_{DD} – AV_{SS1}$, $AV_{REFP} – AV_{REFM}$, $V_{DD} – V_{SS}$)
  The following condition should be satisfied: $AV_{SS1} = V_{SS}$. When performing A/D conversion of analog input pin ANIn (n = 0 to 5), the following condition should be satisfied: $AV_{DD} = V_{DD}$. A 1-μF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in **Figure 17-32**, and connection should be made so that the following conditions are satisfied at the supply side.
  $AV_{REFM} = AV_{SS1} = V_{SS}$
  When the 12-bit A/D converter is not used, the following conditions should be satisfied.
  $AV_{REFP} = AV_{DD} = V_{DD}$ and $AV_{REFM} = AV_{SS1} = V_{SS}$

Figure 17-32. Power Supply Pin Connection Example



17.8.11 Notes on board design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (ANI0 to ANI5), reference power supply pin (AV_{REFP}/VREFOUT), reference ground pin (AV_{REFM}), and analog power supply (AV_{DD}) should be separated from digital circuits using the analog ground (AV_{SS1}). The analog ground (AV_{SS1}) should be connected to a stable digital ground (V_{SS}) on the board (single-point ground plane connection).

17.8.12 Notes on noise prevention

(i) To prevent the analog input pins (ANI0 to ANI5) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AV_{DD} and AV_{SS1} and between AV_{REFP}/VREFOUT and AV_{REFM}, and a protection circuit should be connected to protect the analog input pins (ANI0 to ANI5) as shown in **Figure 17-33**.

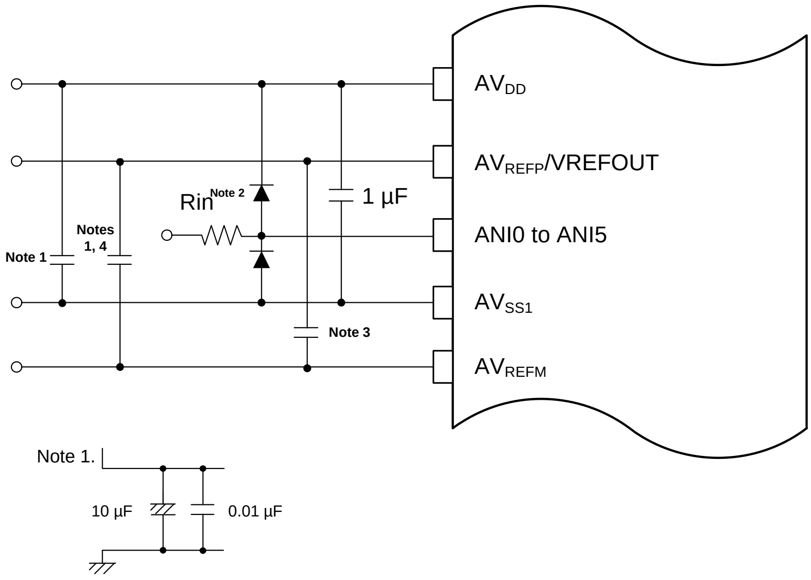
(ii) Do not switch these pins with other pins during conversion.

(iii) The accuracy is improved if the HALT mode is set immediately after the start of conversion.

(iv) When A/D conversion of the signal on any channel (ANI0 to ANI5) is selected, do not change the levels output on P20 to P25 during conversion, since doing so may decrease the accuracy.

(v) When a pin adjacent to a pin on which A/D conversion is in progress is used as a digital I/O port pin, the result of A/D conversion may differ from the accurate value due to noise coupling. Take care to avoid pulses which change dramatically, like digital signals, being input or output through adjacent pins during A/D conversion.

Figure 17-33. Sample Protection Circuit for Analog Inputs



Notes 1. The values indicated in the figure are reference values.

2. Rin: Signal source impedance

3. When AV_{REFP} is selected as the high-potential reference voltage for the A/D converter, connect the AV_{REFP} pin to the AV_{REFM} pin through a 10-µF capacitor.
When VREFOUT is selected as the high-potential reference voltage for the A/D converter, connect the AV_{REFP} pin to the AV_{REFM} pin through a 1-µF capacitor.
We recommend placing the capacitor adjacent to the reference pins of the A/D converter.

4. When AV_{REFP} is selected as the high-potential reference voltage for the A/D converter, only connect the capacitor to the AV_{REFM} pin.

CHAPTER 18 TEMPERATURE SENSOR 2

18.1 Functions of Temperature Sensor

The RL78/I1C (512 KB) has an on-chip temperature sensor. Temperature can be measured by measuring the output voltage from the temperature sensor using the 12-bit A/D converter. The mode of the temperature sensor can be switched to one of the following three modes by setting the temperature control register.

- High-temperature range mode: Mode 1, 0°C to 90°C (Output Image Diagram Mode 1)
- Normal-temperature range mode: Mode 2, –20°C to 70°C (Output Image Diagram Mode 2)
- Low-temperature range mode: Mode 3, –40°C to 50°C (Output Image Diagram Mode 3)

Temperature sensor may be used in HS (high-speed main) mode.

Figure 18-1 shows a block diagram of temperature sensor.

Figure 18-1. Block Diagram

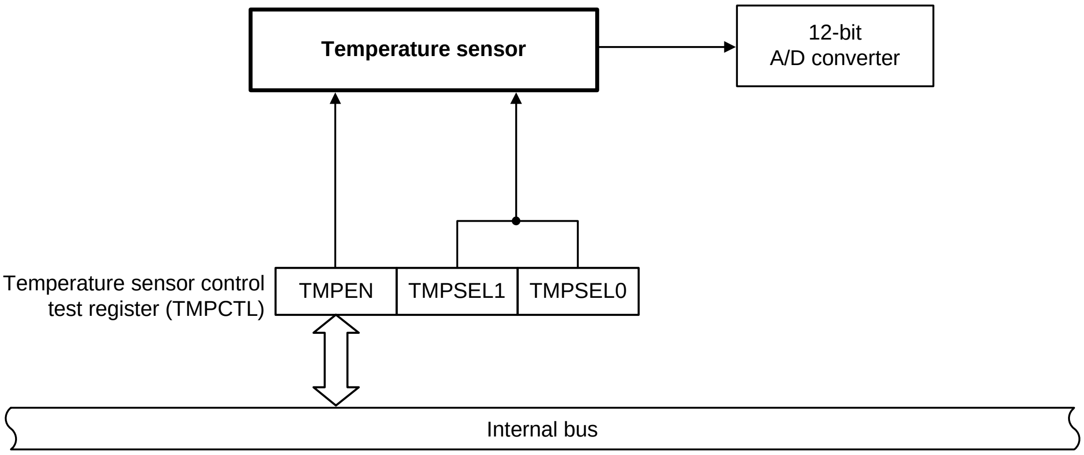
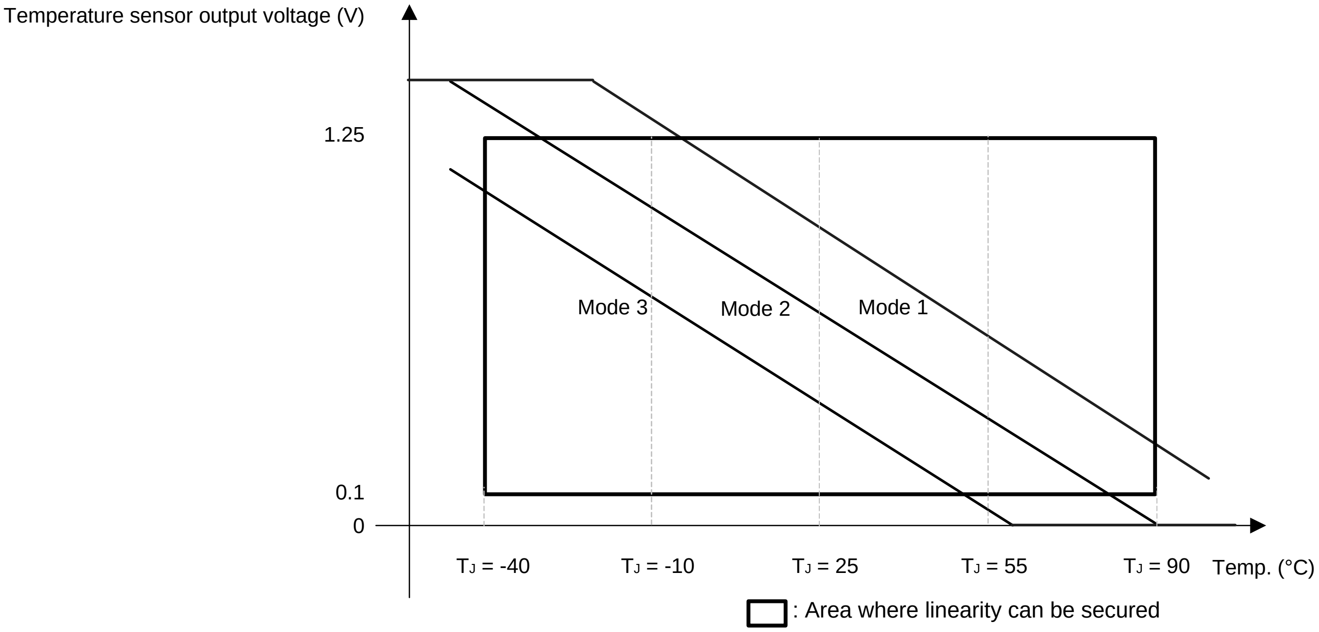


Figure 18-2. Output Image Diagram



18.2 Registers

Table 18-1 shows the register used for the temperature sensor.

Table 18-1. Register

Item	Configuration
Control registers	Temperature sensor control test register (TMPCTL) Peripheral enable register 0 (PER0) Peripheral reset control register 0 (PRR0)

18.2.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the temperature sensor 2 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 an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 18-3. Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

ADCEN	Control over supply of the input clock for the 12-bit A/D converter and temperature sensor 2
0	Stops input clock supply. ● SFR used by the 12-bit A/D converter and temperature sensor 2 cannot be written. The read value is 00H. However, the SFR is not initialized. Note
1	Enables input clock supply. ● SFR used by the 12-bit A/D converter and temperature sensor 2 can be read and written.

Note To initialize the 12-bit A/D converter, temperature sensor 2, and the SFR used by the 12-bit A/D converter and temperature sensor 2, use bit 5 (ADCRES) of PRR0.

18.2.2 Temperature sensor control test register (TMPCTL)

The TMPCTL register is used to stop or start operation of the temperature sensor, and select the mode of the temperature sensor.

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

Reset generation clears this register to 00H.

Figure 18-4. Format of Temperature sensor control test register (TMPCTL)

Address: F03B0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TMPCTL	TMPEN[Note 1]	0	0	0	0	0	TMPSEL1[Note 2]	TMPSEL0[Note 2]

TMPEN	Temperature sensor operation control
0	Temperature sensor stops operation
1	Temperature sensor starts operation

TMPSEL1	TMPSEL0	Temperature sensor operation selection
0	0	Normal-temperature range (Mode 2)
0	1	High-temperature range (Mode 1)
1	0	Low-temperature range (Mode 3)
Other than above		Setting prohibited

Notes 1. After setting the TMPEN bit to 1, a 50 μs operation stabilization wait time is necessary.

2. After changing bits TMPSEL1-TMPSEL0, a 15 μs mode switch stabilization wait time is necessary.

Caution Be sure to clear bits 6 to 2 to "0".

18.2.3 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.

To place the temperature sensor in the reset state, be sure to set bit 5 (ADCRES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 18-5. Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

ADCRES	Control resetting of the 12-bit A/D converter and temperature sensor 2
0	The 12-bit A/D converter and temperature sensor 2 are released from the reset state.
1	The 12-bit A/D converter and temperature sensor 2 are in the reset state.

18.3 Setting Procedures

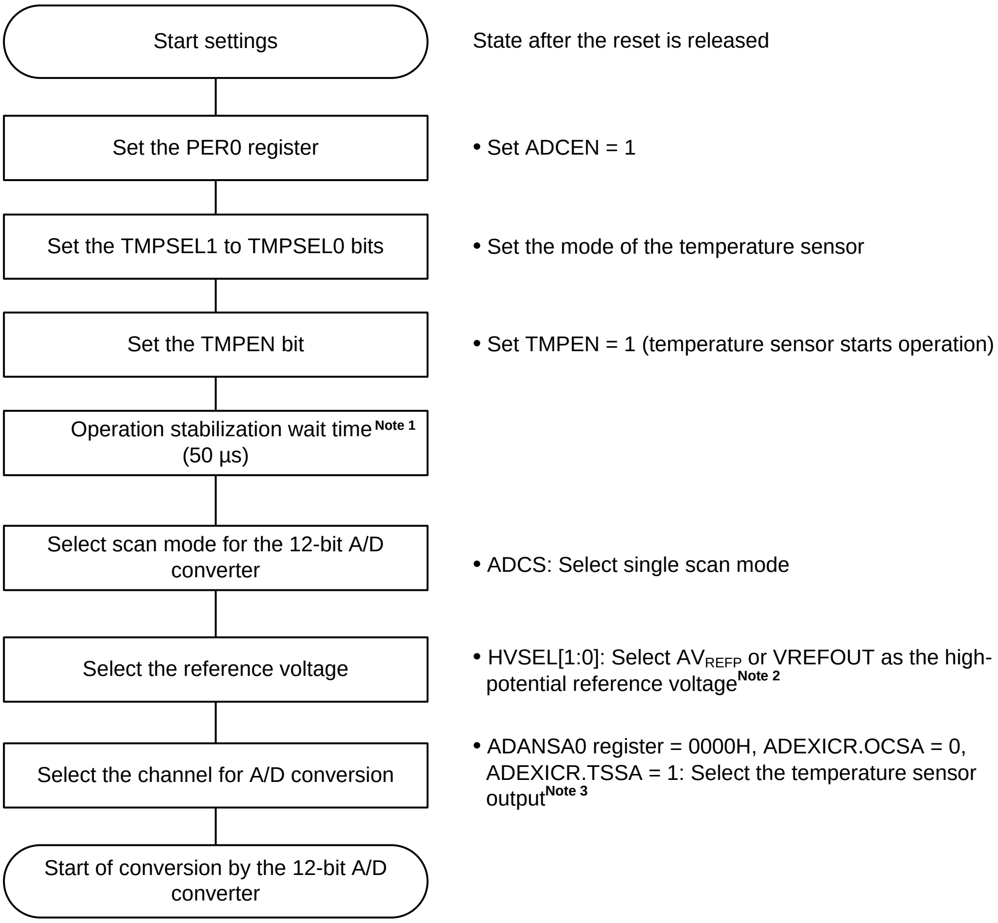
The procedures for setting the temperature sensor are shown below.

18.3.1 Starting operation of the temperature sensor

Figure 18-6 shows the setting flowchart when starting operation of temperature sensor.

For the procedure for setting the 12-bit A/D converter, see **17.3.2.5 A/D conversion of temperature sensor output/internal reference voltage (V_{BGR})**.

Figure 18-6. Setting Flowchart When Starting Operation of Temperature Sensor


Start settings
State after the reset is released
Set the PER0 register
• Set ADCEN = 1
Set the TMPSEL1 to TMPSEL0 bits
• Set the mode of the temperature sensor
Set the TMPEN bit
• Set TMPEN = 1 (temperature sensor starts operation)
Operation stabilization wait time Note 1 (50 µs)
Select scan mode for the 12-bit A/D converter
• ADCS: Select single scan mode
Select the reference voltage
• HVSEL[1:0]: Select AVREFP or VREFOUT as the high-potential reference voltage Note 2
Select the channel for A/D conversion
• ADANSA0 register = 0000H, ADEXICR.OCSA = 0, ADEXICR.TSSA = 1: Select the temperature sensor output Note 3
Start of conversion by the 12-bit A/D converter

Notes 1. Operation stabilization wait time is required until the 12-bit A/D converter starts conversion.

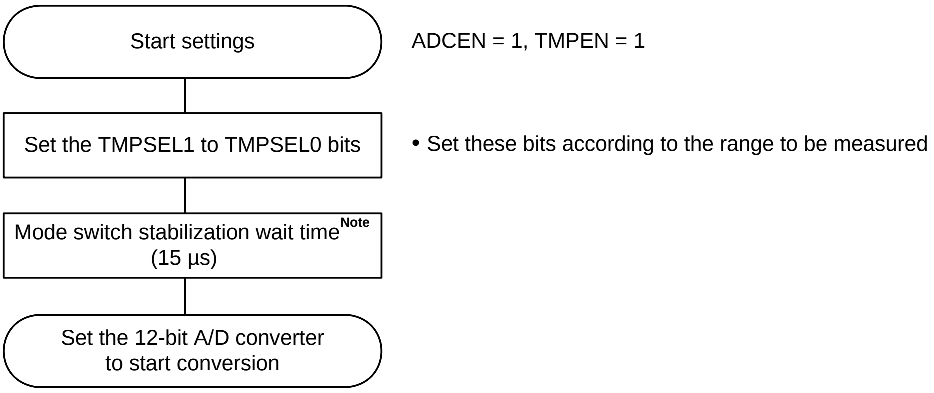
2. VREFOUT is selected when the voltage reference is operating. AV_{REFP} is selected other than when the voltage reference is operating. For the procedure to start the voltage reference, see **17.6 Selecting Reference Voltage**.

3. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANSA0 register and the OCSA bit should all be set to 0 in single scan mode.

18.3.2 Switching modes

Figure 18-7 shows the setting flowchart when switching mode of temperature sensor.

Figure 18-7. Setting Flowchart When Switching Mode of Temperature Sensor


Start settings
ADCEN = 1, TMPEN = 1
Set the TMPSEL1 to TMPSEL0 bits
• Set these bits according to the range to be measured
Mode switch stabilization wait time^Note (15 µs)
Set the 12-bit A/D converter to start conversion

Note Mode switch stabilization wait time is required until the 12-bit A/D converter starts conversion.

CHAPTER 19 24-BIT ΔΣ A/D CONVERTER

The 24-bit ΔΣ A/D converter has a 24-bit resolution when converting an analog input signal to digital values.

19.1 Functions of 24-bit ΔΣ A/D Converter

The 24-bit ΔΣ A/D converter has the following functions:

O S/N+D ratio: 80 dB min. (when pre-amplifier gain of ×1 is selected)
O 24-bit resolution (conversion result register: 24 bits)
O 3 channels (current channel: 2 channels voltage channel: 1 channel) (80-pin product)
O 4 channels (current channel: 2 channels voltage channel: 2 channels) (100-pin product)
O Analog input: 8 (positive, negative input/channel)
O ΔΣ conversion mode
O Pre-amplifier gain selectable: ×1, ×2, ×4, ×8, ×16, or ×32[Note] (channels 0, 1, 2, and 3)
O Operating voltage: AV_{DD} = 2.4 to 5.5 V, AV_{SS0} = 0 V
O Analog input voltage: ±0.500 V (when pre-amplifier gain of ×1 is selected)
±0.250 V (when pre-amplifier gain of ×2 is selected)
±0.125 V (when pre-amplifier gain of ×4 is selected)
±62.5 mV (when pre-amplifier gain of ×8 is selected)
±31.25 mV (when pre-amplifier gain of ×16 is selected)
±15.625 mV (when pre-amplifier gain of ×32[Note] is selected)
O Reference voltage generation (0.8 V (TYP.) can be output)
O Sampling frequency: 3906.25 Hz (4 kHz sampling mode)/1953.125 Hz (2 kHz sampling mode)
O HPF cutoff frequency: 0.607 Hz, 1.214 Hz, 2.429 Hz, or 4.857 Hz can be selected
O Operating clock: High-speed system clock (f_{MX}) (only 12 MHz crystal resonator can be used)
High-speed on-chip oscillator ((f_{HOCO}/2) (f_{HOCO} = 24 MHz))

Note The pre-amplifier gain of ×32 is obtained by multiplying the ×16 gain by two through a low-pass filter (LPF). Therefore, when the bypass function for the low-pass filter is in use, the pre-amplifier gain of ×32 is not selectable.

Cautions 1. When using the high-speed system clock (f_{MX}) by setting DSADCK in the PCKC register to 1, supply 12 MHz.

2. The 24-bit ΔΣ A/D converter cannot be used in the LP (low-power main) or LV (low-voltage main) mode.

Table 19-1 lists the configuration of 24-bit ΔΣ A/D converter. **Figures 19-1** and **19-2** show the block diagram of 24-bit ΔΣ A/D converter, respectively.

Table 19-1. Configuration of 24-bit ΔΣ A/D Converter

Item	Configuration
Analog input	3 channels and 6 inputs (80-pin product) 4 channels and 8 inputs (100-pin product)
Internal units	Pre-amplifier block ΔΣ A/D converter Reference voltage generation Phase adjustment circuits (PHC0, PHC1, PHC2, PHC3) Digital filter (DF) block ● Decimation filters (DECs) ● Low-pass filter (LPF) High-pass filter (HPF)

Figure 19-1. Block Diagram of 24-bit ΔΣ A/D Converter (100-pin Product)

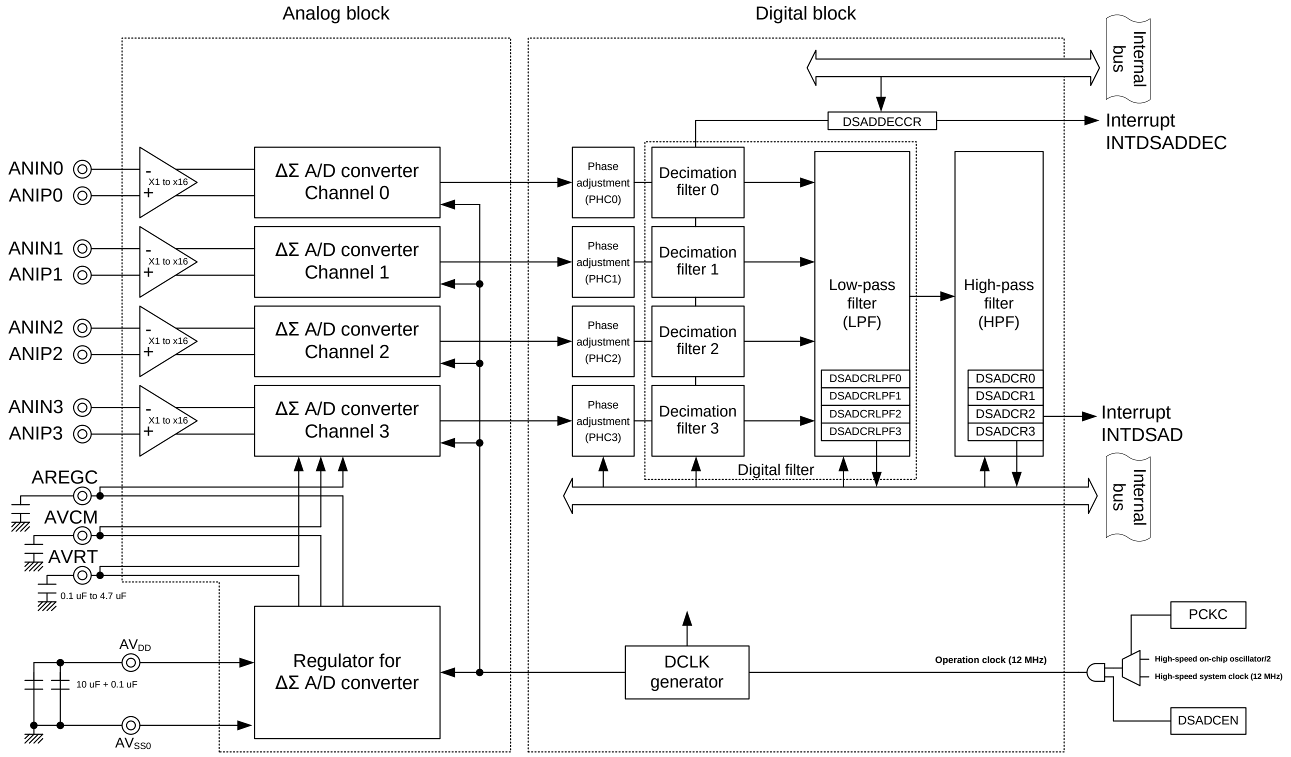
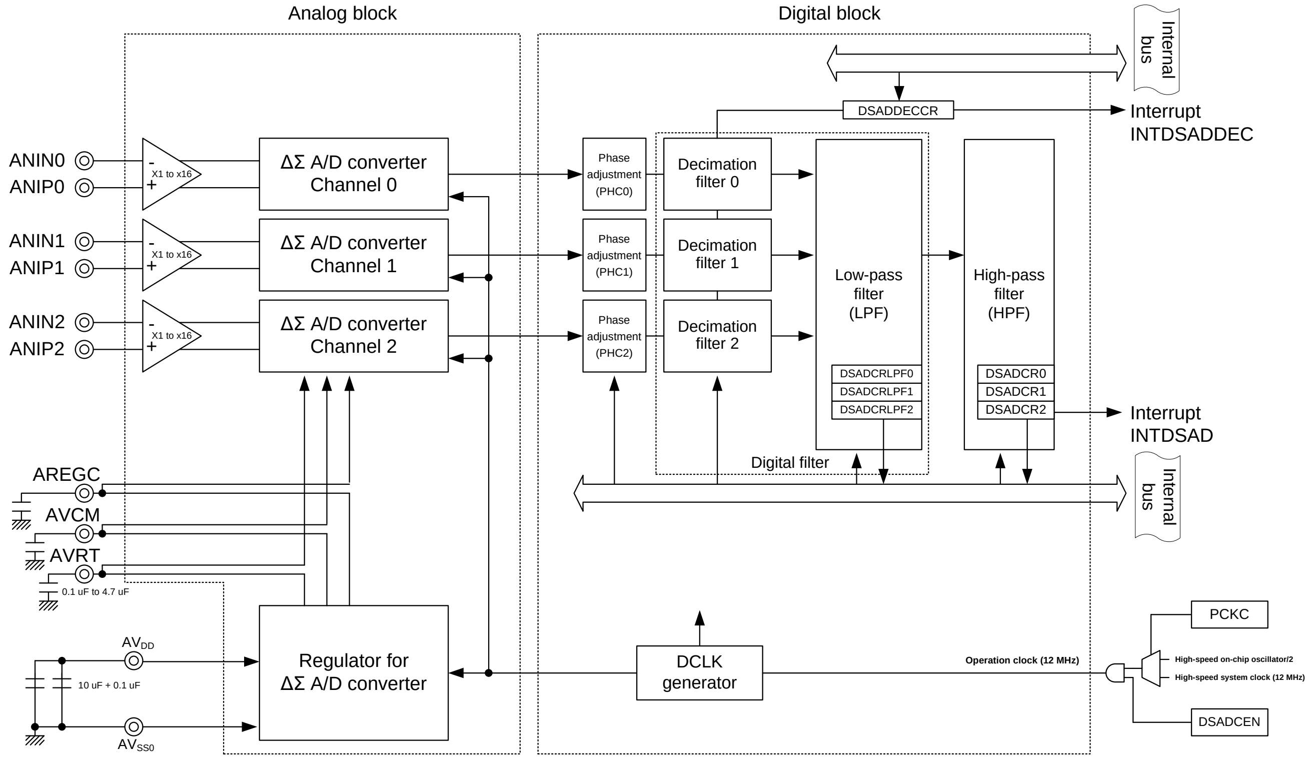


Figure 19-2. Block Diagram of 24-bit ΔΣ A/D Converter (80-pin Product)



19.1.1 I/O pins

Table 19-2 lists the I/O pins for the 24-bit ΔΣ A/D converter.

Table 19-2. Pin Configuration

Name	Symbol	I/O	Function
Analog input positive pin 0 to analog input positive pin 3	ANIPn	Input	Analog input pin for ΔΣ A/D converter (positive input)Notes 1, 3
Analog input negative pin 0 to analog input negative pin 3	ANINn	Input	Analog input pin for ΔΣ A/D converter (negative input)Notes 1, 3
ΔΣ A/D converter power supply voltage pin	AREGC	–	ΔΣ A/D converter power supply voltage
Common voltage pin	AVCM	–	Common voltage
Reference voltage pin	AVRT	–	Reference voltage
Analog power supply pin	AV_{DD}	–	Analog power supplyNote 2
Analog GND	AV_{SS0}	–	Analog GND pin

Notes 1. One channel inputs two signals. The ANINn pin is the negative input, while the ANIPn pin is the positive input.

2. Connect capacitors of 10 µF + 0.1 µF as stabilization capacitance between the AV_{DD} and AV_{SS0} pins.

3. Consider the sensor delay when selecting the pin for a single phase two-wire meter.

Remark n = 0 to 3 for 100-pin product, n = 0 to 2 for 80-pin product

19.1.2 Pre-amplifier

This unit amplifies an analog input signal to be input to the ANINn and ANIPn pins.

The gain can be set to ×1, ×2, ×4, ×8, ×16, or ×32Note using the register settings.

Note The pre-amplifier gain of ×32 is obtained by multiplying the ×16 gain by two through a low-pass filter (LPF). Therefore, when the bypass function for the low-pass filter is in use, the pre-amplifier gain of ×32 is not selectable.

Remark n = 0 to 3 for 100-pin product, n = 0 to 2 for 80-pin product

19.1.3 ΔΣ A/D converter

Four ΔΣ A/D converter circuits are provided so that a total of four channels of analog inputs can be converted into 2-bit digital signals. These four ΔΣ A/D converter circuits operate synchronously. Each 2-bit digital value is passed through the phase adjustment circuit, the digital filter, and the high-pass filter, and then stored into the conversion result registers (DSADCR0 to DSADCR3) as the conversion result of each channel. Each time conversion by all four channels is completed, an interrupt request signal is generated to inform the CPU that the result of conversion can be read. The sampling frequency (fs) can be selected as 3906.25 Hz or 1953.125 Hz. The maximum retention time and over-sampling frequency depend on the sampling frequency as shown in the table below. Make sure that reading of the ΔΣ A/D converter conversion result registers is completed within their maximum retention time.

Sampling Frequency (fs)	Maximum Retention Time	Over-sampling Frequency
3906.25 Hz (4 kHz sampling mode)	192 µs	1.5 MHz
1953.125 Hz (2 kHz sampling mode)	384 µs	750 kHz

19.1.4 Reference voltage generator

An internal reference voltage source (band-gap reference circuit) is provided and a reference voltage is output from the reference voltage output pin AVRT. Connect a capacitor of 0.47 µF as external capacitance.

19.1.5 Phase adjustment circuits (PHCn)

This circuit adjusts the phase of input analog signals. The phase between analog signals is adjusted in steps (one step = 384 fs) up to 1151 steps.

Phase shifts between input analog signals occur due to external components (such as current sensors). Use the DSADPHCn register to correct such phase shifts in advance, because these shifts can decrease the precision of power calculations.

A step for correcting phase shifts can be adjusted in 0.0144° units if the line frequency is 60 Hz, or in 0.0120° units if the line frequency is 50 Hz.

Four phase adjustment circuits (PHC0 to PHC3) are provided in the RL78/I1C (512 KB) and each phase can be adjusted for input signals.

Remark n = 0 to 3 for 100-pin product, n = 0 to 2 for 80-pin product

19.1.6 Digital filter (DF) block

The digital filter (DF) block consists of multiple decimation filters (DECs) and a low-pass filter (LPF).

This block eliminates high harmonic signals in outputs from the ΔΣ A/D converter channels and thins out the data rate to 1/384.

19.1.7 Decimation filters (DECs)

The decimation filters thin out the data rate to 1/32. In the RL78/I1C (512 KB), the output values of the respective decimation filters are readable. A decimation filter output end interrupt (INTDSADDEC) is generated at the times of output from the decimation filters.

The output rates are 1.5 MHz ÷ 32 = 46.875 kHz when DSADFR = 0 and 0.75 MHz ÷ 32 = 23.4375 kHz when DSADFR = 1.

19.1.8 Low-pass filter (LPF)

The low-pass filter eliminates high harmonic signals in outputs from the ΔΣ A/D converter channels and, in combination with the individual decimation filters, thins out the data rate to 1/384. In the RL78/I1C (512 KB), the output values of the low-pass filter can be read even if the high-pass filter is inserted.

The output rates are 1.5 MHz ÷ 384 = 3906.25 Hz when DSADFR = 0 and 0.75 MHz ÷ 384 = 1953.125 Hz when DSADFR = 1.

19.1.9 High-pass filter (HPF)

The high-pass filter eliminates the DC component included in the input signal and the DC offset generated by the analog circuit.

Whether the high-pass filter is inserted or not can be selected for each channel.

19.2 Registers

Table 19-3 lists the registers used for the 24-bit ΔΣ A/D converter.

Table 19-3. Registers (1/2)

Item	Configuration
Control registers	ΔΣ A/D converter mode register (DSADMR)
	ΔΣ A/D converter gain control register 0 (DSADGCR0)
	ΔΣ A/D converter gain control register 1 (DSADGCR1)
	ΔΣ A/D converter HPF control register (DSADHPFCR)
	ΔΣ A/D converter decimation filter control register (DSADDECCR)
	ΔΣ A/D converter phase control register 0 (DSADPHCR0)
	ΔΣ A/D converter phase control register 1 (DSADPHCR1)
	ΔΣ A/D converter phase control register 2 (DSADPHCR2)
	ΔΣ A/D converter phase control register 3 (DSADPHCR3)
Registers	ΔΣ A/D converter conversion result register 0L (DSADCR0L)
	ΔΣ A/D converter conversion result register 0M (DSADCR0M)
	ΔΣ A/D converter conversion result register 0H (DSADCR0H)
	ΔΣ A/D converter conversion result register 1L (DSADCR1L)
	ΔΣ A/D converter conversion result register 1M (DSADCR1M)
	ΔΣ A/D converter conversion result register 1H (DSADCR1H)
	ΔΣ A/D converter conversion result register 2L (DSADCR2L)
	ΔΣ A/D converter conversion result register 2M (DSADCR2M)
	ΔΣ A/D converter conversion result register 2H (DSADCR2H)
	ΔΣ A/D converter conversion result register 3L (DSADCR3L)
	ΔΣ A/D converter conversion result register 3M (DSADCR3M)
	ΔΣ A/D converter conversion result register 3H (DSADCR3H)
	ΔΣ A/D converter conversion result register (LPF) 0L (DSADCRLPF0L)
	ΔΣ A/D converter conversion result register (LPF) 0M (DSADCRLPF0M)
	ΔΣ A/D converter conversion result register (LPF) 0H (DSADCRLPF0H)
	ΔΣ A/D converter conversion result register (LPF) 1L (DSADCRLPF1L)
	ΔΣ A/D converter conversion result register (LPF) 1M (DSADCRLPF1M)
	ΔΣ A/D converter conversion result register (LPF) 1H (DSADCRLPF1H)
	ΔΣ A/D converter conversion result register (LPF) 2L (DSADCRLPF2L)
	ΔΣ A/D converter conversion result register (LPF) 2M (DSADCRLPF2M)
	ΔΣ A/D converter conversion result register (LPF) 2H (DSADCRLPF2H)
	ΔΣ A/D converter conversion result register (LPF) 3L (DSADCRLPF3L)
	ΔΣ A/D converter conversion result register (LPF) 3M (DSADCRLPF3M)
	ΔΣ A/D converter conversion result register (LPF) 3H (DSADCRLPF3H)
	ΔΣ A/D converter conversion result register 0 (DSADCR0)
	ΔΣ A/D converter conversion result register 1 (DSADCR1)
	ΔΣ A/D converter conversion result register 2 (DSADCR2)
	ΔΣ A/D converter conversion result register 3 (DSADCR3)
	ΔΣ A/D converter conversion result register (LPF) 0 (DSADCRLPF0)
	ΔΣ A/D converter conversion result register (LPF) 1 (DSADCRLPF1)
	ΔΣ A/D converter conversion result register (LPF) 2 (DSADCRLPF2)
	ΔΣ A/D converter conversion result register (LPF) 3 (DSADCRLPF3)

Table 19-3. Registers (2/2)

Item	Configuration
Control registers	Peripheral enable register 1 (PER1)
	Peripheral clock control register (PCKC)
	Peripheral reset control register 1 (PRR1)

19.2.1 ΔΣ A/D converter mode register (DSADMR)

The DSADMR register is used to set the operating mode of the ΔΣ A/D converter. This register is used to select the sampling period and the resolution of the ΔΣ A/D converter, and control powering on each channel and enabling its operation.

The DSADMR register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 19-3. Format of ΔΣ A/D Converter Mode Register (DSADMR)

Address: F0740H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSADMR	DSADFR	DSADTYP	0	0	DSADPON3	DSADPON2	DSADPON1	DSADPON0	0	0	0	0	DSADCE3	DSADCE2	DSADCE1	DSADCE0

DSADFR	Sampling frequency selection
0	3906.25 Hz
1	1953.125 Hz
This bit is used to select the sampling frequency.	

DSADTYP	Resolution selection when reading ΔΣ A/D converter conversion result register
0	24-bit resolution
1	16-bit resolution
When DSADTYP = 0: The lower 16 bits in the ΔΣ A/D converter conversion result register can be read by reading the ΔΣ A/D converter conversion result register (DSADCRn). Read DSADCRnH as the higher 8 bits. When DSADTYP = 1: The higher 16 bits in the ΔΣ A/D converter conversion result register can be read by reading the ΔΣ A/D converter conversion result register (DSADCRn).	

DSADPONn	ΔΣ A/D converter power-on control (analog block) of channel n
0	Power down
1	Power on

DSADCEn	ΔΣ A/D converter operation enable (analog and digital blocks) of channel n
0	Electric charge reset
1	Normal operation
This bit is used to enable conversion operation of the ΔΣ A/D converter. The charge of the analog block and the conversion result of the digital block are reset. To reset the charge of the ΔΣ A/D converter normally, first set the DSADCEn bit from 1 to 0, and then wait for at least 1.4 μs before performing conversion again.	

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

Cautions 1. When a clock faster than 12 MHz is selected as the CPU clock (f_{CLK}), do not write to the DSADMR register successively. When writing to this register successively, allow at least one cycle of f_{CLK} between writes. Three cycles is required until the ΔΣ A/D converter is powered down after the DSADPONn bit is set to 0. When setting the DSADPONn bit to 1 again, be sure to allow at least three cycles of f_{CLK} before powering on the ΔΣ A/D converter.

2. Be sure to clear bits 13, 12, and 7 to 4 to “0”.

Remark n = 0 to 3

19.2.2 ΔΣ A/D converter gain control register 0 (DSADGCR0)

The DSADGCR0 register is used to select the gain of the programmable gain amplifier of channels 0 and 1.

DSADGCR0 can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19-4. Format of ΔΣ A/D Converter Gain Control Register 0 (DSADGCR0)

Address: F0742H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSADGCR0	0	DSADGAIN12	DSADGAIN11	DSADGAIN10	0	DSADGAIN02	DSADGAIN01	DSADGAIN00

DSADGAIN12	DSADGAIN11	DSADGAIN10	Selection of programmable amplifier gain of channel 1
Bit 6	Bit 5	Bit 4	
0	0	0	PGA gain: ×1
0	0	1	PGA gain: ×2
0	1	0	PGA gain: ×4
0	1	1	PGA gain: ×8
1	0	0	PGA gain: ×16
1	0	1	PGA gain: ×32[Note]
Other than above			Setting prohibited
These bits are used to control the PGA gain. The gain can be set in the range of ×1 to ×32.			

DSADGAIN02	DSADGAIN01	DSADGAIN00	Selection of programmable amplifier gain of channel 0
Bit 2	Bit 1	Bit 0	
0	0	0	PGA gain: ×1
0	0	1	PGA gain: ×2
0	1	0	PGA gain: ×4
0	1	1	PGA gain: ×8
1	0	0	PGA gain: ×16
1	0	1	PGA gain: ×32[Note]
Other than above			Setting prohibited
These bits are used to control the PGA gain. The gain can be set in the range of ×1 to ×32.			

Note The programmable amplifier gain of ×32 is obtained by multiplying the ×16 gain by two through a low-pass filter (LPF). Therefore, when the bypass function for the low-pass filter is in use, the programmable amplifier gain of ×32 is not selectable.

Caution Be sure to clear bits 7 and 3 to "0".

19.2.3 ΔΣ A/D converter gain control register 1 (DSADGCR1)

The DSADGCR1 register is used to select the gain of the programmable gain amplifier of channels 2 and 3.

DSADGCR1 can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19-5. Format of ΔΣ A/D Converter Gain Control Register 1 (DSADGCR1)

Address: F0743H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSADGCR1	0	DSADGAIN32	DSADGAIN31	DSADGAIN30	0	DSADGAIN22	DSADGAIN21	DSADGAIN20

DSADGAIN32	DSADGAIN31	DSADGAIN30	Selection of programmable amplifier gain of channel 3
Bit 6	Bit 5	Bit 4	
0	0	0	PGA gain: ×1
0	0	1	PGA gain: ×2
0	1	0	PGA gain: ×4
0	1	1	PGA gain: ×8
1	0	0	PGA gain: ×16
1	0	1	PGA gain: ×32[Note]
Other than above			Setting prohibited
These bits are used to control the PGA gain. The gain can be set in the range of ×1 to ×32.			

DSADGAIN22	DSADGAIN21	DSADGAIN20	Selection of programmable amplifier gain of channel 2
Bit 2	Bit 1	Bit 0	
0	0	0	PGA gain: ×1
0	0	1	PGA gain: ×2
0	1	0	PGA gain: ×4
0	1	1	PGA gain: ×8
1	0	0	PGA gain: ×16
1	0	1	PGA gain: ×32[Note]
Other than above			Setting prohibited
These bits are used to control the PGA gain. The gain can be set in the range of ×1 to ×32.			

Note The programmable amplifier gain of ×32 is obtained by multiplying the ×16 gain by two through a low-pass filter (LPF). Therefore, when the bypass function for the low-pass filter is in use, the programmable amplifier gain of ×32 is not selectable.

Caution Be sure to clear bits 7 and 3 to “0”.

19.2.4 ΔΣ A/D converter HPF control register (DSADHPFCR)

The DSADHPFCR register is used to select the cutoff frequency of the high pass filter and disable or enable the high-pass filter for each channel.

DSADHPFCR can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19-6. Format of ΔΣ A/D Converter HPF Control Register (DSADHPFCR)

Address: F0745H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSADHPFCR	DSADCOF1	DSADCOF0	0	0	DSADTHR3	DSADTHR2	DSADTHR1	DSADTHR0

DSADCOF1	DSADCOF0	Selection of cutoff frequency of high-pass filter
Bit 7	Bit 6	
0	0	0.607 Hz
0	1	1.214 Hz
1	0	2.429 Hz
1	1	4.857 Hz

DSADTHR3	High-pass filter disable of channel 3
0	High-pass filter used
1	High-pass filter not used

DSADTHR2	High-pass filter disable of channel 2
0	High-pass filter used
1	High-pass filter not used

DSADTHR1	High-pass filter disable of channel 1
0	High-pass filter used
1	High-pass filter not used

DSADTHR0	High-pass filter disable of channel 0
0	High-pass filter used
1	High-pass filter not used

Cautions 1. Be sure to clear bits 5 and 4 to “0”.

2. Writing to the DSADTHRn bit shall be completed when the following condition is satisfied:

- **DSADCEn = 0 (Conversion is being stopped)**

Remark The high-pass filter convergence time can be changed by changing the high-pass filter cut-off frequency. The convergence time decreases as the cut-off frequency increases.

To initialize the high-pass filter, use the DSADRES bit of the peripheral reset control register (PRR1).

19.2.5 ΔΣ A/D converter decimation filter control register (DSADDECCR)

The DSADDECCR register controls generation of the decimation filter output end interrupt.

Setting the DSADDECEN bit to 1 enables generation of the decimation filter output end interrupt (INTDSADDEC).

The DSADDECSEL bit is used to select output values after filtering by using the decimation filter alone or with the low-pass filter in reading from the ΔΣ A/D converter conversion result registers (LPF) n (DSADCRLPFnH, DSADCRLPFnM, DSADCRLPFnL) or the ΔΣ A/D converter conversion result registers (LPF) n (DSADCRLPFn).

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

Reset signal generation clears this register to 00H.

Figure 19-7. Format of ΔΣ A/D Converter Decimation Filter Control Register (DSADDECCR)

Address: F0746H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DSADDECCR	DSADDECEN	0	0	0	0	0	0	DSADDECSEL

DSADDECEN	Selection of enabling or disabling generation of the decimation filter output end interrupt
0	Disables generation of the decimation filter output end interrupt (INTDSADDEC).
1	Enables generation of the decimation filter output end interrupt (INTDSADDEC).

DSADDECSEL	Target for reading from DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL or DSADCRLPFn
0	Output values read from DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL or DSADCRLPFn are those for the given channel after filtering by the low-pass filter.
1	Output values read from DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL or DSADCRLPFn are those for the given channel after filtering by the decimation filter.

Remark n = 0 to 3

19.2.6 ΔΣ A/D converter phase control register n (DSADPHCRn) (n = 0, 1, 2, 3)

The DSADPHCRn register is used to select the channel for input to the phase adjustment n circuit and set the adjustment step.

DSADPHCRn can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 19-8. Format of ΔΣ A/D Converter Phase Control Register n (DSADPHCRn) (n = 0, 1, 2, 3)

Address: F0750H (DSADPHCR0), F0752H (DSADPHCR1), F0754H (DSADPHCR2), F0756H (DSADPHCR3)
After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSADPHCRn	0	0	0	0	0	DSADPHCn10	DSADPHCn9	DSADPHCn8	DSADPHCn7	DSADPHCn6	DSADPHCn5	DSADPHCn4	DSADPHCn3	DSADPHCn2	DSADPHCn1	DSADPHCn0

DSADPHCn10 to DSADPHCn0[Note]	ΔΣ A/D converter channel n phase adjustment
000H	Through (no phase adjustment)
001H	One step
...	...
47EH	1150 steps
47FH	1151 steps
These bits are used to adjust the phase of 2-bit ΔΣ A/D conversion data input from the analog block. The DSADPHCn10 to DSADPHCn0 bits are used to specify the phase adjustment (one step = 384 fs). Since the sampling frequency (3906.25 Hz) is included in the calculation of the adjustment value, the phase that can be adjusted by correcting one step is 1 [s]/(384 [fs] × 3906.25 [Hz]) = 0.6667 [µs]. Example: To adjust the phase by 100 µs for 2-bit signal input from the analog block, the register set value will be 96H since 100/0.6667 = 150 [steps].	

Note These bits cannot be set to a value of 480H or greater.

Caution Be sure to clear bits 15 to 11 to “0”.

19.2.7 ΔΣ A/D converter conversion result register n (DSADCRnL, DSADCRnM, DSADCRnH) (n = 0, 1, 2, 3)

The sets of DSADCRn (H/M/L) registers form 24-bit registers that are used to retain the results of conversion by the ΔΣ A/D converter of the corresponding channels.

The DSADCRnL, DSADCRnM, and DSADCRnH registers can be read individually by 8-bit memory manipulation instructions. The results of conversion from the ΔΣ A/D converter read from each of the registers depends on the setting of the DSADTYP bit in the ΔΣ A/D converter mode register (DSADMR).

Setting the DSADCEn bit in the ΔΣ A/D converter mode register (DSADMR) to 0 or generation of a reset signal clears the DSADCRnL, DSADCRnM, and DSADCRnH registers to 00H.

Figure 19-9. Format of ΔΣ A/D Converter Conversion Result Register n (DSADCRnL, DSADCRnM, DSADCRnH) (n = 0, 1, 2, 3)

Address: F0760H (DSADCR0L), F0761H (DSADCR0M), F0762H (DSADCR0H),
F0764H (DSADCR1L), F0765H (DSADCR1M), F0766H (DSADCR1H),
F0768H (DSADCR2L), F0769H (DSADCR2M), F076AH (DSADCR2H),
F076CH (DSADCR3L), F076DH (DSADCR3M), F076EH (DSADCR3H)

After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRnH	DSADCRnH[7:0]							

Symbol	7	6	5	4	3	2	1	0
DSADCRnM	DSADCRnM[7:0]							

Symbol	7	6	5	4	3	2	1	0
DSADCRnL	DSADCRnL[7:0]							

- When 24-bit resolution is set (DSADTYP in the DSADMR register = 0)

	DSADCRnH		DSADCRnM		DSADCRnL	
Bit	b23	b16	b15	b8	b7	b0
	ΔΣ A/D conversion result n [23:16]		ΔΣ A/D conversion result n [15:8]		ΔΣ A/D conversion result n [7:0]	

Bit	Symbol	Result of conversion by channel n
b7 to b0	DSADCRnL[7:0]	Bits 7 to 0 of the results of conversion by channel n
b15 to b8	DSADCRnM[7:0]	Bits 15 to 8 of the results of conversion by channel n
b23 to b16	DSADCRnH[7:0]	Bits 23 to 16 of the results of conversion by channel n

(**Caution** is listed on the next page.)

- When 16-bit resolution is set (DSADTYP in the DSADMR register = 1)

	DSADCRnH	DSADCRnM	DSADCRnL
Bit	b23 b16	b15 b8	b7 b0
	ΔΣ A/D conversion result n [23:16]	ΔΣ A/D conversion result n [23:16]	ΔΣ A/D conversion result n [15:8]

Bit	Symbol	Result of conversion by channel n
b7 to b0	DSADCRnL[7:0]	Bits 15 to 8 of the results of conversion by channel n
b15 to b8	DSADCRnM[7:0]	Bits 23 to 16 of the results of conversion by channel n
b23 to b16	DSADCRnH[7:0]	Bits 23 to 16 of the results of conversion by channel n

Caution Be sure to read the ΔΣ A/D converter conversion result registers within their maximum retention time after the ΔΣ A/D conversion end interrupt is generated.

19.2.8 ΔΣ A/D converter conversion result register n (DSADCRn) (n = 0, 1, 2, 3)

The DSADCRn register is used in access to the results of conversion of each channel by 16-bit memory manipulation instructions.

The DSADCRn registers can be read individually by 16-bit memory manipulation instructions. The results of conversion from the ΔΣ A/D converter read from the registers depends on the setting of the DSADTYP bit in the ΔΣ A/D converter mode register (DSADMR).

Setting the DSADCEn bit in the ΔΣ A/D converter mode register (DSADMR) to 0 or generation of a reset signal clears the DSADCRn registers to 0000H.

Figure 19-10. Format of ΔΣ A/D Converter Conversion Result Register n (DSADCRn) (n = 0, 1, 2, 3)

Address: F0760H (DSADCR0), F0764H (DSADCR1), F0768H (DSADCR2), F076CH (DSADCR3)
After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSADCRn	DSADCRn[15:0]															

- When 24-bit resolution is set (DSADTYP in the DSADMR register = 0)Note

Bit	Symbol	Result of conversion by channel n
b15 to b0	DSADCRn[15:0]	Bits 15 to 0 of the results of conversion by channel n

- When 16-bit resolution is set (DSADTYP in the DSADMR register = 1)Note

Bit	Symbol	Result of conversion by channel n
b15 to b0	DSADCRn[15:0]	Bits 23 to 8 of the results of conversion by channel n

Note Access to the results of conversion by the ΔΣ A/D converter through these registers depends on the setting of the DSADTYP bit in the DSADMR register.

- DSADTYP = 0: The 16 lower-order bits can be read from these registers.
  Read DSADCRnH for the 8 higher-order bits.
- DSADTYP = 1: The 16 higher-order bits can be read from these registers.

Caution Be sure to read the ΔΣ A/D converter conversion result registers within their maximum retention time after the ΔΣ A/D conversion end interrupt is generated.

19.2.9 ΔΣ A/D converter conversion result register (LPF) n (DSADCRLPFnL, DSADCRLPFnM, DSADCRLPFnH) (n = 0, 1, 2, 3)

The sets of DSADCRLPFn (H/M/L) registers form 24-bit registers that are used to retain the results of conversion after filtering by the low-pass filter or those (17-bit values) after filtering by the decimation filter of the corresponding channels. Operation of each of the DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL registers depends on the setting of the DSADDECSEL bit of the ΔΣ A/D converter decimation filter control register (DSADDECCR).

The DSADCRLPFnL, DSADCRLPFnM, and DSADCRLPFnH registers can be read individually by 8-bit memory manipulation instructions. The results of conversion (LPF) from the ΔΣ A/D converter read from each of the registers depends on the setting of the DSADTYP bit in the ΔΣ A/D converter mode register (DSADMR).

Setting the DSADCEn bit in the ΔΣ A/D converter mode register (DSADMR) to 0 or generation of a reset signal clears the DSADCRLPFnL, DSADCRLPFnM, and DSADCRLPFnH registers to 00H.

Figure 19-11. Format of ΔΣ A/D Converter Conversion Result Register (LPF) n (DSADCRLPFnL, DSADCRLPFnM, DSADCRLPFnH) (n = 0, 1, 2, 3)

Address: F0770H (DSADCRLPF0L), F0771H (DSADCRLPF0M), F0772H (DSADCRLPF0H),
F0774H (DSADCRLPF1L), F0775H (DSADCRLPF1M), F0776H (DSADCRLPF1H),
F0778H (DSADCRLPF2L), F0779H (DSADCRLPF2M), F077AH (DSADCRLPF2H),
F077CH (DSADCRLPF3L), F077DH (DSADCRLPF3M), F077EH (DSADCRLPF3H)

After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRLPFnH	DSADCRLPFnH [7:0]							

Symbol	7	6	5	4	3	2	1	0
DSADCRLPFnM	DSADCRLPFnM [7:0]							

Symbol	7	6	5	4	3	2	1	0
DSADCRLPFnL	DSADCRLPFnL [7:0]							

Caution Be sure to read the ΔΣ A/D converter conversion result register (LPF) n within their maximum retention time after the ΔΣ A/D conversion end interrupt is generated.

(1) Reference to results from the low-pass filter (the DSADDECSEL bit = 0)

When the value of the DSADDECSEL bit is 0, the combination of the DSADCRLPFn (H/M/L) registers acts as a 24-bit register to hold the results of conversion after filtering by using the low-pass filter on the results from each of the channels. In this case, the registers always hold the results of conversion while the high-pass filter is disabled regardless of the settings of the DSADTHRn bits (n = 0 to 3) in the ΔΣ A/D converter HPF control register (DSADHPFCR) to enable or disable the high-pass filter.

- When 24-bit resolution is set (the DSADTYP bit in the DSADMR register = 0)
  The DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL registers can be used to read 24-bit conversion results (LPF) from the ΔΣ A/D converter.

	DSADCRLPFnH	DSADCRLPFnM	DSADCRLPFnL
Bit	b23 b16	b15 b8	b7 b0
	ΔΣ A/D conversion result (LPF) n [23:16]	ΔΣ A/D conversion result (LPF) n [15:8]	ΔΣ A/D conversion result (LPF) n [7:0]

Bit	Symbol	Result of conversion (LPF) by channel n
b7 to b0	DSADCRLPFnL [7:0]	Bits 7 to 0 of the results of conversion (LPF) by channel n
b15 to b8	DSADCRLPFnM [7:0]	Bits 15 to 8 of the results of conversion (LPF) by channel n
b23 to b16	DSADCRLPFnH [7:0]	Bits 23 to 16 of the results of conversion (LPF) by channel n

- When 16-bit resolution is set (the DSADTYP bit in the DSADMR register = 1)
  The DSADCRLPFnM and DSADCRLPFnL registers can be used to read the 16 higher-order bits of the A/D conversion result register (LPF) n. Reading from an DSADCRLPFnH register also returns the values of bits 23 to 16 of the A/D conversion result register (LPF) n.

	DSADCRLPFnH	DSADCRLPFnM	DSADCRLPFnL
Bit	b23 b16	b15 b8	b7 b0
	ΔΣ A/D conversion result (LPF) n [23:16]	ΔΣ A/D conversion result (LPF) n [23:16]	ΔΣ A/D conversion result (LPF) n [15:8]

Bit	Symbol	Result of conversion (LPF) by channel n
b7 to b0	DSADCRLPFnL [7:0]	Bits 15 to 8 of the results of conversion (LPF) by channel n
b15 to b8	DSADCRLPFnM [7:0]	Bits 23 to 16 of the results of conversion (LPF) by channel n
b23 to b16	DSADCRLPFnH [7:0]	Bits 23 to 16 of the results of conversion (LPF) by channel n

(2) Reference to results from the decimation filter (the DSADDECSEL bit = 1)

When the value of the DSADDECSEL bit is 1, the combination of the DSADCRLPFn (H/M/L) registers acts as a 24-bit register to hold the results of conversion (17-bit values) after filtering by using the decimation filter on the results from each of the channels.

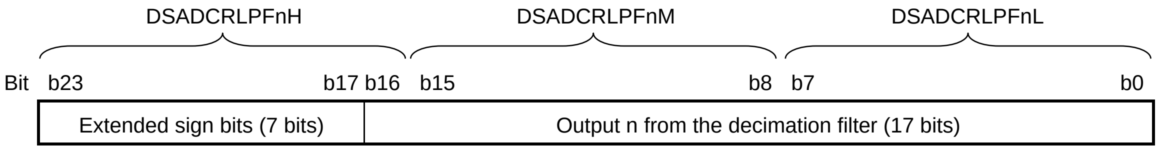
The DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL registers can be read individually by 8-bit memory manipulation instructions.

Caution Be sure to read out the results during the period from generation of a decimation filter output end interrupt (INTDSADDEC) to updating of the ΔΣ A/D converter conversion result register (LPF) n.

- Operation of each of the registers is common to the cases where the setting is for 24-bit resolution (the DSADTYP bit in the DSADMR register = 0) and for 16-bit resolution (the DSADTYP bit in the DSADMR register = 1).

  The DSADCRLPFnH, DSADCRLPFnM, and DSADCRLPFnL registers can be used to read the 17-bit results of conversion after filtering by using the decimation filter of each channel of the ΔΣ A/D converter. In this case, the values are flush right.

  The values obtained by extending the sign bit (the value of the 16th bit) are stored in 7 bits (bits 23 to 17).


DSADCRLPFnH
DSADCRLPFnM
DSADCRLPFnL
Bit
b23
b17 b16
b15
b8
b7
b0
Extended sign bits (7 bits)
Output n from the decimation filter (17 bits)

Bit	Symbol	Result of conversion by channel n
b7 to b0	DSADCRLPFnL [7:0]	Bits 7 to 0 of the output from the decimation filter
b15 to b8	DSADCRLPFnM [7:0]	Bits 15 to 8 of the output from the decimation filter
b23 to b16	DSADCRLPFnH [7:0]	Bit 16 of the output from the decimation filter Extended sign bits 0 to 6

19.2.10 ΔΣ A/D converter conversion result register (LPF) n (DSADCRLPFnL, DSADCRLPFnM, DSADCRLPFnH) (n = 0, 1, 2, 3)

The DSADCRLPFn register is used in access to the 24-bit registers that hold the results of conversion after filtering by using the low-pass filter and decimation filter or the 17-bit results of conversion after filtering by only using the decimation filter of each channel by 16-bit memory manipulation instructions. Operation of the DSADCRLPFn register depends on the setting of the DSADDECSEL bit of the ΔΣ A/D converter decimation filter control register (DSADDECCR).

The DSADCRLPFn registers can be read individually by 16-bit memory manipulation instructions. The results of conversion from the ΔΣ A/D converter read from the registers depends on the setting of the DSADTYP bit in the ΔΣ A/D converter mode register (DSADMR).

Setting the DSADCEn bit in the ΔΣ A/D converter mode register (DSADMR) to 0 or generation of a reset signal clears the DSADCRLPFn registers to 0000H.

Figure 19-12. Format of ΔΣ A/D Converter Conversion Result Register (LPF) n (DSADCRLPFn) (n = 0, 1, 2, 3)

Address: F0770H (DSADCRLPF0), F0774H (DSADCRLPF1), F0778H (DSADCRLPF2), F077CH (DSADCRLPF3)

After reset: 0000H R

Symbol	15 0
DSADCRLPFn	DSADCRLPFn [15:0]

Caution Be sure to read the ΔΣ A/D converter conversion result register (LPF) n within their maximum retention time after the ΔΣ A/D conversion end interrupt is generated.

(1) Reference to results from the low-pass filter (the DSADDECSEL bit = 0)

When the value of the DSADDECSEL bit is 0, the DSADCRLPFn registers act as registers for use in access to the 24-bit registers that hold the results of conversion after filtering by using the low-pass filter of each channel by 16-bit memory manipulation instructions.

In this case, the registers always hold the results of conversion while the high-pass filter is disabled regardless of the settings of the DSADTHRn bits (n = 0 to 3) in the ΔΣ A/D converter HPF control register (DSADHPFCR) to enable or disable the high-pass filter.

- When 24-bit resolution is set (DSADTYP in the DSADMR register = 0)Note

  The DSADCRLPFn registers can be used to read the 16 lower-order bits of the A/D conversion result register (LPF) n.

  A word memory manipulation instruction reads out the 16 lower-order bits of the corresponding 24-bit value, allowing the acceleration of access to the 24-bit values.

Bit	Symbol	Result of conversion (LPF) by channel n
b15 to b0	DSADCRLPFn [15:0]	Bits 15 to 0 of the results of conversion (LPF) by channel n

- When 16-bit resolution is set (DSADTYP in the DSADMR register = 1)Note

  The DSADCRLPFn registers can be used to read the 16 higher-order bits of the A/D conversion result register (LPF) n. Furthermore, when A/D conversion is with 16-bit resolution, access to the full 16-bit results of conversion by single word memory manipulation instructions becomes possible.

Bit	Symbol	Result of conversion (LPF) by channel n
b15 to b0	DSADCRLPFn [15:0]	Bits 23 to 8 of the results of conversion (LPF) by channel n

Note Access to the results of conversion by the ΔΣ A/D converter through these registers depends on the setting of the DSADTYP bit in the DSADMR register.

- DSADTYP = 0: The 16 lower-order bits can be read from these registers.
  Read DSADCRLPFnH for the 8 higher-order bits.
- DSADTYP = 1: The 16 higher-order bits can be read from these registers.

(2) Reference to results from the decimation filter (the DSADDECSEL bit = 1)

When the value of the DSADDECSEL bit is 1, the DSADCRLPFn registers act as registers for use in access to the 24-bit registers that hold the 17-bit results of conversion after filtering by using the decimation filter of each channel by 16-bit memory manipulation instructions.

- Operation of the registers is common to the cases where the setting is for 24-bit resolution (the DSADTYP bit in the DSADMR register = 0) and for 16-bit resolution (the DSADTYP bit in the DSADMR register = 1)

  The DSADCRLPFn registers can be used to read the 16 lower-order bits of the 17-bit results of conversion after filtering by using the decimation filter of each channel of the ΔΣ A/D converter.

  A word memory manipulation instruction reads out the 16 lower-order bits of the corresponding 24-bit value, allowing the acceleration of access to the 24-bit values.

Bit	Symbol	Result of conversion by channel n
b15 to b0	DSADCRLPFn [15:0]	Bits 15 to 0 of the output from the decimation filter

19.2.11 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the 24-bit ΔΣ A/D converter is to be used, be sure to set bit 0 (DSADCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 19-13. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

DSADCEN	Control of 24-bit ΔΣ A/D converter input clock supply
0	Stops input clock supply. ● SFR used by the 24-bit ΔΣ A/D converter cannot be written. The read value is 00H. However, the SFR is not initialized. Note
1	Enables input clock supply. ● SFR used by the 24-bit ΔΣ A/D converter can be read and written.

Note To initialize the 24-bit ΔΣ A/D converter and the SFR used by the 24-bit ΔΣ A/D converter, use bit 0 (DSADRES) of PRR1.

Cautions 1. When setting the 24-bit ΔΣ A/D converter, be sure to set the DSADCEN bit to 1 first. If DSADCEN = 0, writing to a control register of the ΔΣ A/D converter is ignored.

2. Be sure to clear bit 7 to “0”.

3. When a high-speed on-chip oscillator is selected as the input clock, be sure to run the high-speed on-chip oscillator clock frequency correction function to input clock with high frequency precision.

19.2.12 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.

To place the 24-bit ΔΣ A/D converter in the reset state, be sure to set bit 0 (DSADRES) to 1.

The PRR1 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears the PRR1 register to 00H.

Figure 19-14. Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	<2>	<1>	<0>
PRR1	0	0	SMOTD1RES	SMOTD0RES	0	TRJ1RES	TRJ0RES	DSADRES

DSADRES	Control resetting of the 24-bit ΔΣ A/D converter
0	The 24-bit ΔΣ A/D converter is released from the reset state.
1	The 24-bit ΔΣ A/D converter is in the reset state.

19.2.13 Peripheral clock control register (PCKC)

The PCKC register is used to control peripheral clocks. Set bit 0 to select a clock for the 24-bit ΔΣ A/D converter.

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

Reset signal generation clears this register to 00H.

Figure 19-15. Format of Peripheral Clock Control Register (PCKC)

Address: F0098H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PCKC	0	0	0	0	0	0	PLLCK	DSADCK

DSADCK	Selection of operation clock for 24-bit ΔΣ A/D converter
0	Supply high-speed on-chip oscillator clock (f_{HOCO}/2). (Stop f_{MX} supply)Note 1
1	Supply high-speed system clock (f_{MX})Note 2

Notes 1. When selecting the high-speed on-chip oscillator clock, be sure to run the high-speed on-chip oscillator clock frequency correction function.

2. Only a 12 MHz crystal oscillator can be used as the high-speed system clock frequency (f_{MX}).

Caution Be sure to clear bits 7 to 2 to “0”.

19.3 Operation

The 24-bit ΔΣ A/D converter has the digital signal input pins for four ΔΣ A/D converter conversion results. By passing 2-bit values obtained from these ΔΣ A/D converter conversion results through the digital filter, the value is converted into 24-bit digital values.

The mode setting of the ΔΣ A/D converter of the analog block depends on the values of the DSADMR, DSADGCR0, and DSADGCR1 register. **Table 19-4** lists the mode settings.

Table 19-4. Mode Settings

Signal/Mode	<1> Normal	<2> ΔΣ A/D Conversion Stop	<3> Power-down
DSADGAINn2 to DSADGAINn0	Any value	Any value	Any value
DSADPONn	1	1	0
DSADCEn	1	0	0

Remark n = 0 to 3

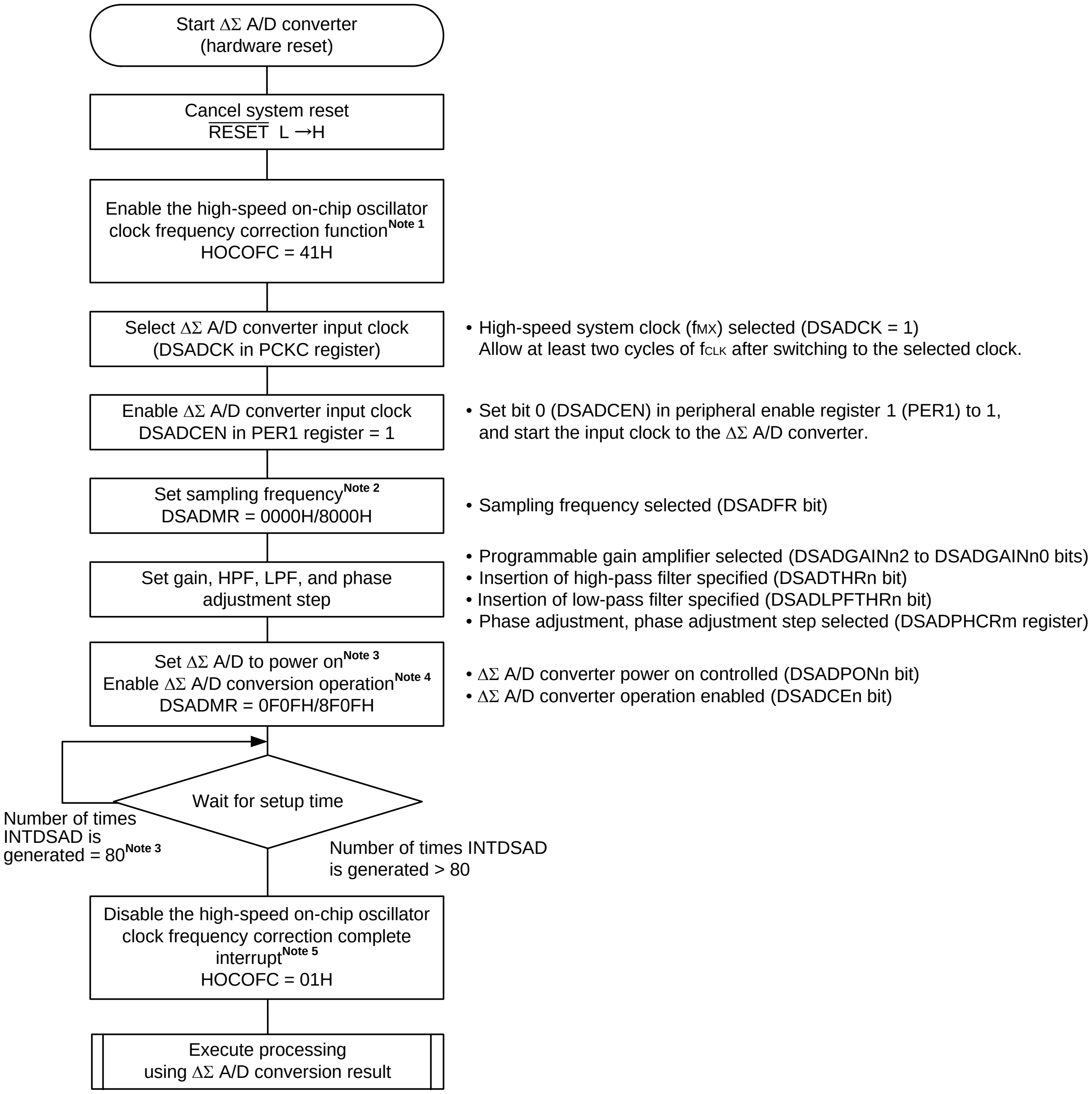
19.3.1 Operation of 24-bit ΔΣ A/D converter

When selecting the high-speed on-chip oscillator clock (f_{HOCO}/2), be sure to run the high-speed on-chip oscillator clock frequency correction function according to **7.3.2 Operation procedure** before running the ΔΣ A/D converter.

When selecting the high-speed system clock (f_{MX}), allow at least two cycles of f_{CLK} after switching to the selected clock.

The 24-bit ΔΣ A/D converter starts operating when the DSADPONn bit (n = 0 to 3) and the DSADCEn bit in the DSADMR register are set to 1. The setup time of the analog block and digital filter block is required after power on and start of conversion. Perform initialization in accordance with the flowchart below.

Figure 19-16. Initialization Flowchart


Start ΔΣ A/D converter
(hardware reset)
Cancel system reset
RESET L →H
Enable the high-speed on-chip oscillator clock frequency correction function Note 1
HOCOFC = 41H
Select ΔΣ A/D converter input clock (DSADCK in PCKC register)
• High-speed system clock (fMX) selected (DSADCK = 1) Allow at least two cycles of fCLK after switching to the selected clock.
Enable ΔΣ A/D converter input clock DSADCEN in PER1 register = 1
• Set bit 0 (DSADCEN) in peripheral enable register 1 (PER1) to 1, and start the input clock to the ΔΣ A/D converter.
Set sampling frequency Note 2
DSADMR = 0000H/8000H
• Sampling frequency selected (DSADFR bit)
Set gain, HPF, LPF, and phase adjustment step
• Programmable gain amplifier selected (DSADGAINn2 to DSADGAINn0 bits)
• Insertion of high-pass filter specified (DSADTHRn bit)
• Insertion of low-pass filter specified (DSADLPFTHRn bit)
• Phase adjustment, phase adjustment step selected (DSADPHCRm register)
Set ΔΣ A/D to power on Note 3
Enable ΔΣ A/D conversion operation Note 4
DSADMR = 0F0FH/8F0FH
• ΔΣ A/D converter power on controlled (DSADPONn bit)
• ΔΣ A/D converter operation enabled (DSADCEn bit)
Wait for setup time
Number of times INTDSAD is generated = 80 Note 3
Number of times INTDSAD is generated > 80
Disable the high-speed on-chip oscillator clock frequency correction complete interrupt Note 5
HOCOFC = 01H
Execute processing using ΔΣ A/D conversion result

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

Notes 1. When selecting the high-speed on-chip oscillator clock, be sure to run the high-speed on-chip oscillator clock frequency correction function before running the ΔΣ A/D converter.

2. Set the sampling frequency while the ΔΣ A/D converter is powered down.

3. The setup time (the number of times INTDSAD is to be generated) when DSADPONn is set to 0 and then 1 will be officially determined after evaluation.

4. If the ΔΣ A/D converter is temporarily stopped for initialization (DSADCEn = 0 with DSADPONn = 1) and then restarted, it is necessary to wait for a certain setup time. In this case, since stabilization time is necessary for the converter, wait for one INTDSAD to be generated as the setup time.
To initialize the ΔΣ A/D converter, make sure that DSADCEn remains 0 for at least 1.4 μs.

5. Perform only when selecting the high-speed on-chip oscillator clock.

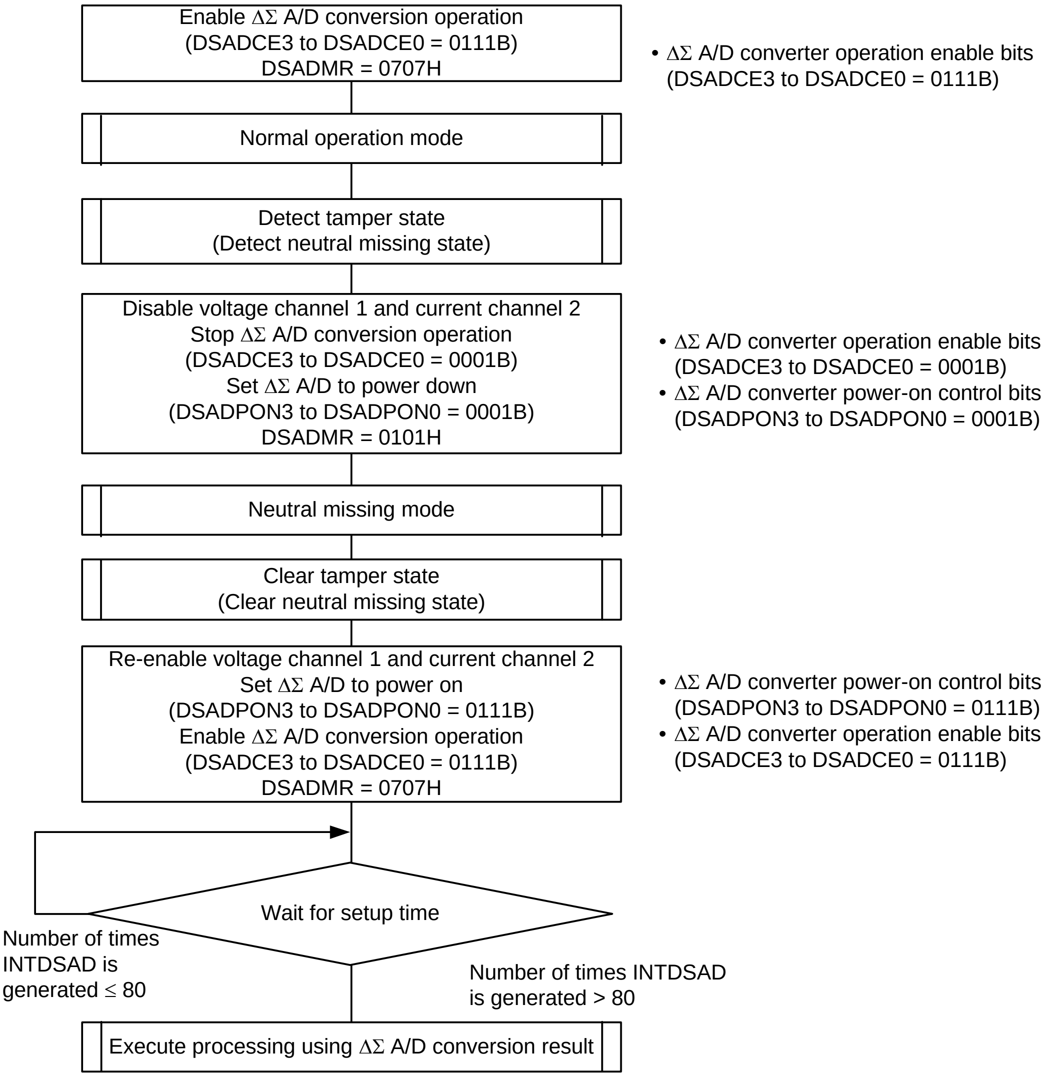
Remark n = 0 to 3; m = 0, 1

19.3.2 Procedure for switching from normal operation mode to neutral missing mode

Figure 19-17 shows the procedure for switching from normal operation (with anti-tamper) (a total of three: current channel 0, voltage channel 1, and current channel 2 operate) to neutral missing mode (only current channel 0 operates), in single-phase two-wire mode.

In neutral missing mode, there are cases when only current channel 0 operates and only current channel 2 operates. Use the same procedure when switching the mode.

Figure 19-17. Procedure for Switching from Normal Operation Mode to Neutral Missing Mode


Enable ΔΣ A/D conversion operation
(DSADCE3 to DSADCE0 = 0111B)
DSADMR = 0707H
• ΔΣ A/D converter operation enable bits
(DSADCE3 to DSADCE0 = 0111B)
Normal operation mode
Detect tamper state
(Detect neutral missing state)
Disable voltage channel 1 and current channel 2
Stop ΔΣ A/D conversion operation
(DSADCE3 to DSADCE0 = 0001B)
Set ΔΣ A/D to power down
(DSADPON3 to DSADPON0 = 0001B)
DSADMR = 0101H
• ΔΣ A/D converter operation enable bits
(DSADCE3 to DSADCE0 = 0001B)
• ΔΣ A/D converter power-on control bits
(DSADPON3 to DSADPON0 = 0001B)
Neutral missing mode
Clear tamper state
(Clear neutral missing state)
Re-enable voltage channel 1 and current channel 2
Set ΔΣ A/D to power on
(DSADPON3 to DSADPON0 = 0111B)
Enable ΔΣ A/D conversion operation
(DSADCE3 to DSADCE0 = 0111B)
DSADMR = 0707H
• ΔΣ A/D converter power-on control bits
(DSADPON3 to DSADPON0 = 0111B)
• ΔΣ A/D converter operation enable bits
(DSADCE3 to DSADCE0 = 0111B)
Wait for setup time
Number of times
INTDSAD is
generated ≤ 80
Number of times INTDSAD
is generated > 80
Execute processing using ΔΣ A/D conversion result

19.3.3 Interrupt operation

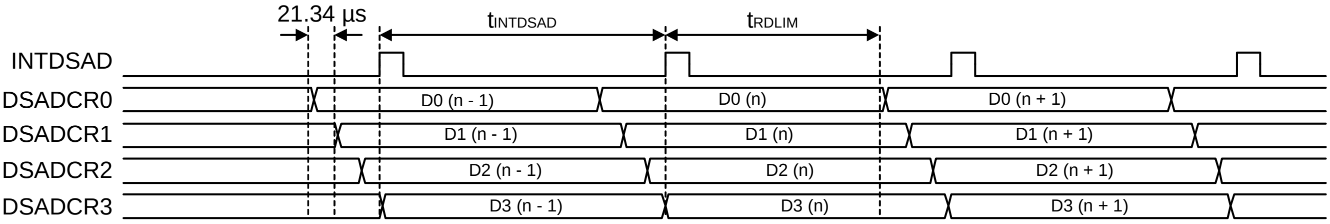
This section describes the interrupt function of the 24-bit ΔΣ A/D converter in the RL78/I1C (512 KB).

19.3.3.1 ΔΣ A/D conversion end interrupt operation

When ΔΣ A/D conversion is enabled, conversion of the signals on the four channels of analog input pins (ANINn and ANIPn) is started. Four sets of ΔΣ A/D converter circuits are provided, and each of which independently executes conversion. Each time conversion by all four channels is completed, an interrupt request signal (INTDSAD) is generated to inform the CPU that the result of conversion can be read.

The cycle of INTDSAD generation ($t_{INTDSAD}$) depends on the sampling frequency specified by the DSADFR bit in the DSADMR register. When a result of conversion is read from the ΔΣ A/D converter conversion result register n (DSADCRn) from within the interrupt service routine, make sure that reading is completed within the appropriate maximum retention time shown in **Figure 19-18**.

Figure 19-18. Timing of Generation of INTDSAD Signal and Storing in DSADCRn Register



$t_{INTDSAD}$: Interrupt generation cycle: 256 µs (DSADFR = 0)
512 µs (DSADFR = 1)
t_{RDLIM}: DSADCR read retention time (max.): 192 µs (DSADFR = 0)
384 µs (DSADFR = 1)

Remark n = 0 to 3

19.3.3.2 Operation of the decimation filter output end interrupt (INTDSADDEC)

Each time processing by a decimation filter is completed, an interrupt request signal (INTDSADDEC) is generated to inform the CPU that the result of conversion can be read.

The DSADDECEN bit in the DSADDECCR register can be used to select enabling or disabling of the generation of the decimation filter output end interrupt (INTDSADDEC).

The cycle of INTDSADDEC generation ($t_{INTDSADDEC}$) depends on the sampling frequency specified by the DSADFR bit in the DSADMR register. When a result of conversion is read from the ΔΣ A/D converter conversion result register (LPF) n (DSADCRLPFn) from within the interrupt service routine, make sure that reading is completed within the appropriate maximum retention time ($t_{RDDECLIM}$) shown blow.

$t_{INTDSADDEC}$: Interrupt generation cycle: 21.3 µs (DSADFR = 0), 42.6 µs (DSADFR = 1)
$t_{RDDECLIM}$: DSADCR read retention time (max.): 21.3 µs (DSADFR = 0), 42.6 µs (DSADFR = 1)

19.3.4 Operation in standby state

In STOP operation mode, the ΔΣ A/D converter and the digital filter do not operate. To reduce current consumption, stop operation of the ΔΣ A/D converter (DSADCEn in the DSADMR register = 0000B) and power down the ΔΣ A/D converter (DSADPONn in the DSADMR register = 0000B) before executing the STOP instruction.

Remark n = 0 to 3

19.3.5 Bypassing the high-pass filter (HPF)

Even while the high-pass filter is enabled, the results of ΔΣ A/D converter conversion with the high-pass filter disabled can be read out at the same time.

For details on reading the results of conversion by the ΔΣ A/D converter while the high-pass filter is disabled, see **(1) Reference to results from the low-pass filter (the DSADDECSEL bit = 0)** in **19.2.9 ΔΣ A/D converter conversion result register (LPF) n (DSADCRLPFnL, DSADCRLPFnM, DSADCRLPFnH) (n = 0, 1, 2, 3)**.

19.3.6 Bypassing the low-pass filter (LPF)

The digital filter (DF) block provides the functions of filtering by the internal decimation filters and by the low-pass filter (LPF), and is usually enabled. To bypass the low-pass filter (LPF), set the DSADDECSEL bit of the DSADDECCR register to 1 to read the 17-bit results of conversion by the ΔΣ A/D converter after filtering by only using the decimation filters of each of the channels. Using the results output from the decimation filters and proceeding with LPF processing in software makes it possible to analyze the power in the higher-order harmonics as well as conducting the typical analysis of power.

For details on reading the results of conversion by the ΔΣ A/D converter while the low-pass filter is disabled, see **(2) Reference to results from the decimation filter (the DSADDECSEL bit = 1)** in **19.2.9 ΔΣ A/D converter conversion result register (LPF) n (DSADCRLPFnL, DSADCRLPFnM, DSADCRLPFnH) (n = 0, 1, 2, 3)**.

19.4 Notes on Using 24-Bit ΔΣ A/D Converter

19.4.1 External pins

The AV_{DD} pin is the analog power supply pin of the ΔΣ A/D converter.

The AV_{SS0} pin is the ground power supply pin of the ΔΣ A/D converter. Always keep the voltage on this pin the same as that on the V_{SS} pin even when the ΔΣ A/D converter is not used.

19.4.2 SFR access

(1) Read the DSADCRn and DSADCRLPFn registers by ΔΣ A/D conversion end interrupt (INTDSAD) servicing. If the DSADCRn and DSADCRLPFn registers are read before a ΔΣ A/D conversion end interrupt is generated, an illegal value may be read because of a conflict between storing the conversion value in the DSADCRn and DSADCRLPFn registers and reading the registers.

The period of the INTDSAD processing during which the DSADCRn and DSADCRLPFn registers are read is 192 µs (when DSADFR is set to 0) or 384 µs (when DSADFR is set to 1), so complete reading of the registers within this time.

Reading the DSADCRnL, DSADCRnM, DSADCRnH, DSADCRLPFnL, DSADCRLPFnM, and DSADCRLPFnH registers are performed in the same conditions as those described above.

(2) After powering on the ΔΣ A/D converter (DSADPONn in the DSADMR register = 1), internal setup time is necessary. Consequently, the data of the first 80 conversions is invalid.

(3) Setup time is also necessary when the ΔΣ A/D converter has been temporarily stopped for initialization (by clearing the DSADCEn bit in the DSADMR register to 0 with DSADPONn = 1) and then restarted. In this case, since stabilization time is necessary for the converter, wait for one INTDSAD to be generated as the setup time. To initialize the ΔΣ A/D converter, make sure that DSADCEn remains 0 for at least 1.4 µs.

(4) The time required for the correct data to be output after the conversion operation has been enabled (by setting the DSADCEn bit to 1) differs depending on the analog input status at that time. This is because the stabilization time of the high-pass filter changes depending on the analog input status.

(5) Set the sampling frequency (DSADFR bit in the DSADMR register) while the DSADPONn bit in the DSADMR register is 0.

Be sure to set the DSADGCR1 and DSADGCR0 registers, DSADCOF[1:0] bits in the DSADHPFCR register, and DSADPHCRn register while the ΔΣ A/D converter is stopped (DSADCEn = 0).

(6) Since the DSADCRn and DSADCRLPFn registers are initialized when the DSADCEn bit is 0, read the DSADCRn and DSADCRLPFn registers when the DSADCEn bit is 1.

(7) Clear the DSADPONn bit in the DSADMR register to 0 before shifting to software STOP mode. If software STOP mode is entered with the DSADPONn bit set to 1, a current will flow.

(8) Writing to the DSADTHRn bit in the DSADHPFCR register shall be completed when the following condition is satisfied:

● DSADCEn = 0 (Conversion is being stopped)

Remark n = 0 to 3

19.4.3 Setting operating clock

When selecting two frequency division of the high-speed on-chip oscillator clock (f_{HOCO}) as an operation clock for the 24-bit ΔΣ A/D converter, set FRQSEL3 of the user option byte to “0” to make base oscillator clock frequency of the high-speed on-chip oscillator clock (f_{HOCO}) 24 MHz.

When using the high-speed system clock (f_{MX}) by setting DSADCK in the PCKC register to 1, supply 12 MHz.

Also, when selecting the high-speed on-chip oscillator clock (f_{HOCO}/2), be sure to run the high-speed on-chip oscillator frequency correction function.

Cautions 1. Count the INTDSAD signal 80 times after the ΔΣ A/D converter is started and then load the converted data when the next INTDSAD signal is generated.

2. Thoroughly evaluate the stabilization time in the environment in which the ΔΣ A/D converter is used.

To stop the 24-bit ΔΣ A/D converter while it is operating, set the DSADPON3 to DSADPON0 bits in the DSADMR register to 0000B, and then set the DSADCEN bit in the PER1 register to 0.

19.4.4 Input range

The 24-bit ΔΣ A/D converter must be used with input within the range stated in **43.6.2 24-bit ΔΣ A/D converter characteristics**. The input of a signal exceeding the input voltage range and at a frequency of 20 kHz or higher may result in a conversion error. Such cases may necessitate a measure in the form of an external circuit etc.

CHAPTER 20 SERIAL ARRAY UNIT

Serial array unit has up to four serial channels. Each channel can achieve Simplified SPI (CSI[Note]), UART, and simplified I^{2}C communication.

Function assignment of each channel supported by the RL78/I1C (512 KB) 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.

80-pin product

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2 (supporting IrDA)	–
	1	–		–
	2	–	–	–
	3	–	–	–

100-pin product

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2 (supporting IrDA)	–
	1	–		–
	2	CSI30	UART3	IIC30
	3	–		–
2	0	–	UART4	–
	1	–		–

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and IIC00 cannot be used, but UART1 or IIC10 can be used.

20.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/I1C (512 KB) has the following features.

20.1.1 Simplified SPI (CSI00, CSI10, CSI30)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

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 **20.5 Operation of Simplified SPI (CSI00, CSI10, CSI30) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate[Note]
  During master communication: Max. $f_{MCK}/2$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**.

20.1.2 UART (UART0 to UART4)

This is a start-stop synchronization 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 **20.6 Operation of UART (UART0 to UART4) Communication.**

[Data transmission/reception]

- Data length of 7, 8, or 9 bits**Note**
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

The LIN-bus is accepted in UART0 (0 and 1 channels of unit 0).

[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 can be specified for the 9-bit data length.

20.1.3 Simplified I²C (IIC00, IIC10, IIC30)

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 **20.8 Operation of Simplified I²C (IIC00, IIC10, IIC30) Communication.**

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

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 the processing flow in **20.8.3 (2)** for details.

Remarks 1. To use an I²C bus of full function, see **CHAPTER 21 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

20.1.4 IrDA

By combining UART2 of the serial array unit and the IrDA module, IrDA communication waveforms can be transmitted or received based on IrDA (Infrared Data Association) standard 1.0. For details, see **CHAPTER 23 IrDA**.

[Data transmission/reception]

- Transfer rate: 115.2 kbps/57.6 kbps/38.4 kbps/19.2 kbps/9600 bps/2400 bps

20.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 20-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits 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, SCK10, SCK30 pins (for simplified SPI), SCL00, SCL10, SCL30 pins (for simplified I^{2}C)
Serial data input	SI00, SI10, SI30 pins (for simplified SPI), RxD1 to RxD4 pins (for UART), RxD0 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO10, SO30 pins (for simplified SPI), TxD1 to TxD4 pins (for UART), TxD0 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA10, SDA30 pins (for simplified I^{2}C)
Control registers	<Registers of unit setting block> ● Peripheral enable register 0 (PER0) ● Serial clock select register m (SPSm) ● Serial channel enable status register m (SEm) ● Serial channel start register m (SSm) ● Serial channel stop register m (STm) ● Serial output enable register m (SOEm) ● Serial output register m (SOm) ● Serial output level register m (SOLm) ● Serial standby control register 0 (SSC0) ● Input switch control register (ISC) ● Noise filter enable register 0 (NFEN0) ● Peripheral reset control register 0 (PRR0)
	<Registers of each channel> ● Serial data register mn (SDRmn) ● Serial mode register mn (SMRmn) ● Serial communication operation setting register mn (SCRmn) ● Serial status register mn (SSRmn) ● Serial flag clear trigger register mn (SIRmn)
	● Port input mode registers 0, 1, 5, 8 (PIM0, PIM1, PIM5, PIM8) ● Port output mode registers 0, 1, 5, 8 (POM0, POM1, POM5, POM8) ● Port mode registers 0, 1, 5, 8 (PM0, PM1, PM5, PM8) ● Port registers 0, 1, 5, 8 (P0, P1, P5, P8)

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

Notes 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.

● mn = 00, 01:	lower 9 bits
● Other than above:	lower 8 bits

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 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30), q: UART number (q = 0 to 4), r: IIC number (r = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

Figure 20-1 shows the block diagram of the serial array unit 0.

Figure 20-1. Block Diagram of Serial Array Unit 0

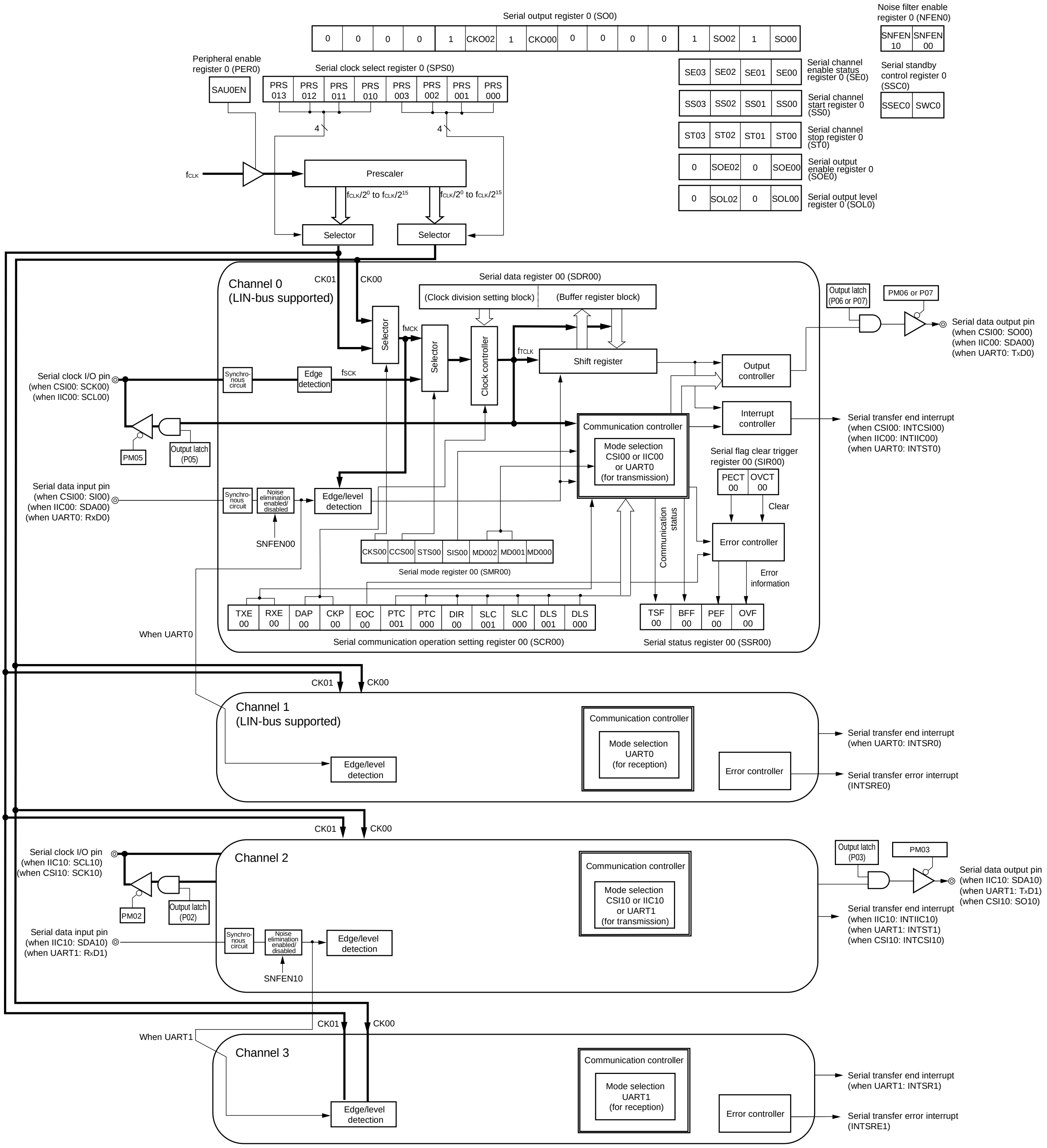


Figure 20-2 shows the block diagram of the serial array unit 1.

Figure 20-2. Block Diagram of Serial Array Unit 1

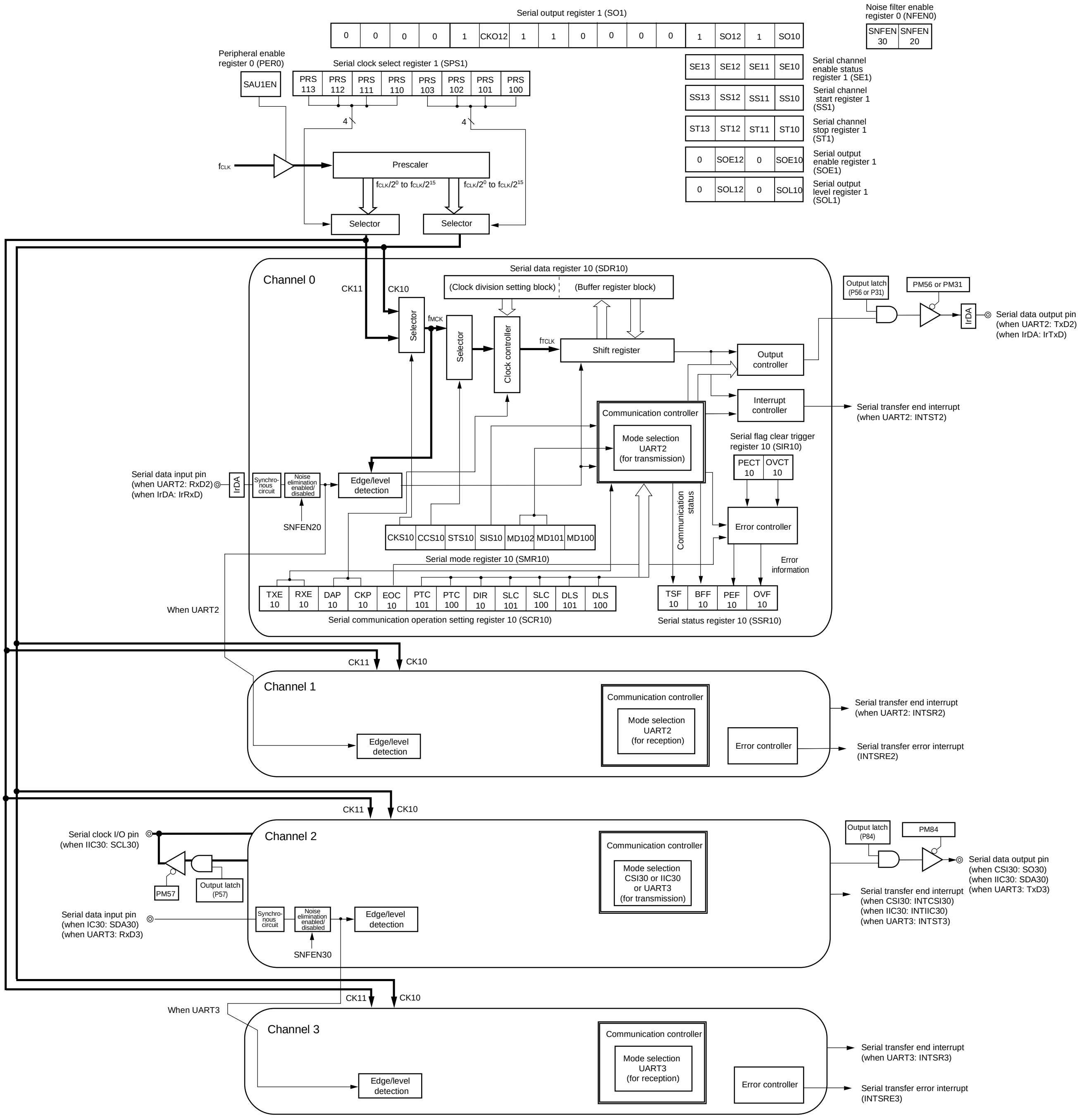
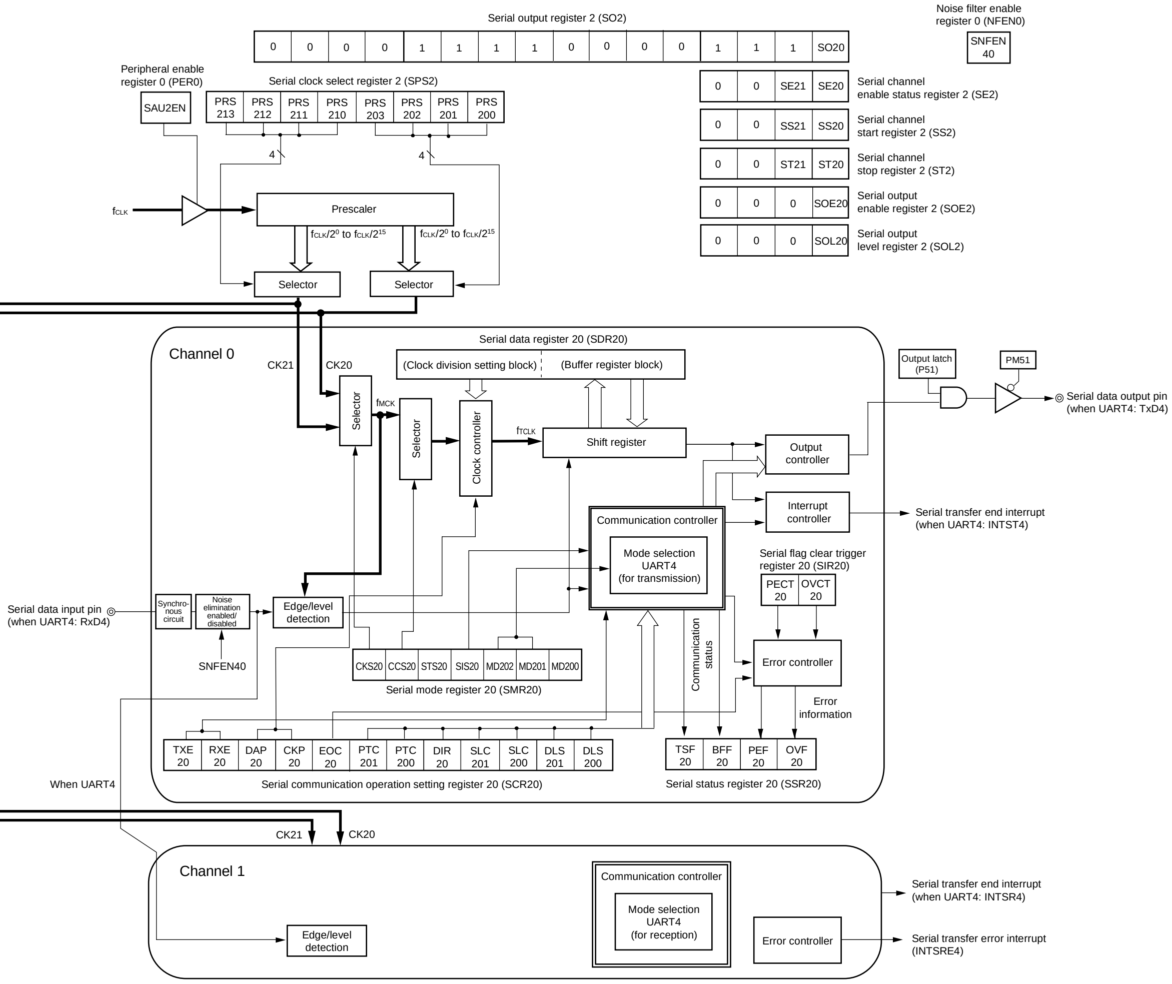

Serial output register 1 (SO1)
0 0 0 0 1 CKO12 1 1 0 0 0 0 1 SO12 1 SO10
Noise filter enable register 0 (NFEN0)
SNFEN 30 SNFEN 20
Peripheral enable register 0 (PER0)
SAU1EN
Serial clock select register 1 (SPS1)
PRS 113 PRS 112 PRS 111 PRS 110 PRS 103 PRS 102 PRS 101 PRS 100
4
4
fCLK
Prescaler
fCLK/2⁰ to fCLK/2¹⁵
fCLK/2⁰ to fCLK/2¹⁵
Selector
Selector
SE13 SE12 SE11 SE10
Serial channel enable status register 1 (SE1)
SS13 SS12 SS11 SS10
Serial channel start register 1 (SS1)
ST13 ST12 ST11 ST10
Serial channel stop register 1 (ST1)
0 SOE12 0 SOE10
Serial output enable register 1 (SOE1)
0 SOL12 0 SOL10
Serial output level register 1 (SOL1)
Channel 0
CK11
CK10
Serial data register 10 (SDR10)
(Clock division setting block)
(Buffer register block)
Selector
fMCK
Selector
Clock controller
fTCLK
Shift register
Output controller
Interrupt controller
Output latch (P56 or P31)
PM56 or PM31
IrDA
Serial data output pin (when UART2: TxD2) (when IrDA: IrTxD)
Serial transfer end interrupt (when UART2: INTST2)
Communication controller
Mode selection UART2 (for transmission)
Serial flag clear trigger register 10 (SIR10)
PECT 10 OVCT 10
Serial data input pin (when UART2: RxD2) (when IrDA: IrRxD)
IrDA
Synchro-nous circuit
Noise elimination enabled/ disabled
Edge/level detection
SNFEN20
CKS10 CCS10 STS10 SIS10 MD102 MD101 MD100
Serial mode register 10 (SMR10)
Communication status
Error controller
Error information
When UART2
TXE 10 RXE 10 DAP 10 CKP 10 EOC 10 PTC 101 PTC 100 DIR 10 SLC 101 SLC 100 DLS 101 DLS 100
Serial communication operation setting register 10 (SCR10)
TSF 10 BFF 10 PEF 10 OVF 10
Serial status register 10 (SSR10)
CK11
CK10
Channel 1
Communication controller
Mode selection UART2 (for reception)
Edge/level detection
Error controller
Serial transfer end interrupt (when UART2: INTSR2)
Serial transfer error interrupt (INTSRE2)
CK11
CK10
Channel 2
Serial clock I/O pin (when IIC30: SCL30)
PM57
Output latch (P57)
Serial data input pin (when IC30: SDA30) (when UART3: RxD3)
Synchro-nous circuit
Noise elimination enabled/ disabled
Edge/level detection
SNFEN30
Communication controller
Mode selection CSI30 or IIC30 or UART3 (for transmission)
Output latch (P84)
PM84
Serial data output pin (when CSI30: SO30) (when IIC30: SDA30) (when UART3: TxD3)
Serial transfer end interrupt (when CSI30: INTCSI30) (when IIC30: INTIIC30) (when UART3: INTST3)
When UART3
CK11
CK10
Channel 3
Edge/level detection
Communication controller
Mode selection UART3 (for reception)
Error controller
Serial transfer end interrupt (when UART3: INTSR3)
Serial transfer error interrupt (INTSRE3)

Figure 20-3 shows the block diagram of the serial array unit 2.

Figure 20-3. Block Diagram of Serial Array Unit 2


Serial output register 2 (SO2)
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 SO20
Noise filter enable register 0 (NFEN0)
SNFEN 40
Peripheral enable register 0 (PER0)
SAU2EN
Serial clock select register 2 (SPS2)
PRS 213 PRS 212 PRS 211 PRS 210 PRS 203 PRS 202 PRS 201 PRS 200
4
4
0 0 SE21 SE20
Serial channel enable status register 2 (SE2)
0 0 SS21 SS20
Serial channel start register 2 (SS2)
0 0 ST21 ST20
Serial channel stop register 2 (ST2)
0 0 0 SOE20
Serial output enable register 2 (SOE2)
0 0 0 SOL20
Serial output level register 2 (SOL2)
fCLK
Prescaler
fCLK/2^0 to fCLK/2^15
fCLK/2^0 to fCLK/2^15
Selector
Selector
Channel 0
CK21
CK20
Serial data register 20 (SDR20)
(Clock division setting block)
(Buffer register block)
Selector
fMCK
Selector
Clock controller
fTCLK
Shift register
Output latch (P51)
PM51
Serial data output pin (when UART4: TxD4)
Output controller
Interrupt controller
Serial transfer end interrupt (when UART4: INTST4)
Communication controller
Mode selection UART4 (for transmission)
Serial flag clear trigger register 20 (SIR20)
PECT 20
OVCT 20
Serial data input pin (when UART4: RxD4)
Synchro-nous circuit
Noise elimination enabled/ disabled
Edge/level detection
SNFEN40
CKS20 CCS20 STS20 SIS20 MD202 MD201 MD200
Serial mode register 20 (SMR20)
Communication status
Error controller
Error information
TXE 20 RXE 20 DAP 20 CKP 20 EOC 20 PTC 201 PTC 200 DIR 20 SLC 201 SLC 200 DLS 201 DLS 200
Serial communication operation setting register 20 (SCR20)
TSF 20 BFF 20 PEF 20 OVF 20
Serial status register 20 (SSR20)
When UART4
CK21
CK20
Channel 1
Communication controller
Mode selection UART4 (for reception)
Serial transfer end interrupt (when UART4: INTSR4)
Edge/level detection
Error controller
Serial transfer error interrupt (INTSRE4)

20.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used[Note].

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 8/9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

Note Only UART0 can be specified for the 9-bit data length.

20.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register)[Note 1]

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 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)

Reset signal generation clears the SDRmn register to 0000H.

Notes 1. Only UART0 can be specified for the 9-bit data length.

2. Rewriting SDRmn[7:0] by 8-bit memory manipulation instruction is prohibited when the operation is stopped (SEmn = 0) (all of SDRmn[15:9] are cleared (0)).

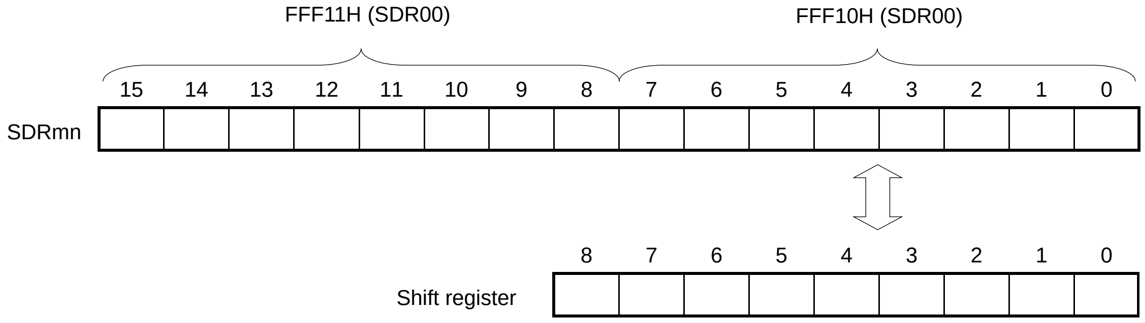
Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30),
q: UART number (q = 0 to 4), r: IIC number (r = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

Figure 20-4. Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11, 20, 21)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01),
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),
FFF58H, FFF59H (SDR20), FFF5AH, FFF5BH (SDR21)

After reset: 0000H R/W

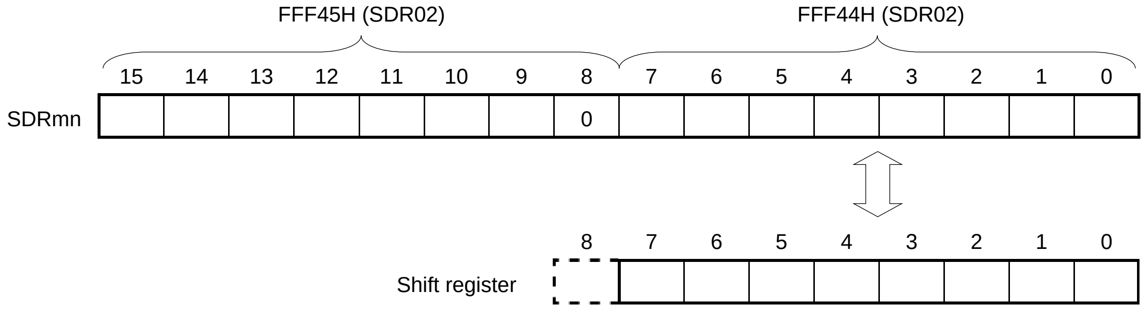


Remark For the function of the higher 7 bits of the SDRmn register, see **20.3 Registers Controlling Serial Array Unit**.

Figure 20-5. 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)

After reset: 0000H R/W



Caution Be sure to clear bit 8 to “0”.

Remark For the function of the higher 7 bits of the SDRmn register, see **20.3 Registers Controlling Serial Array Unit**.

20.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register 0 (SSC0)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 5, 8 (PIM0, PIM1, PIM5, PIM8)
- Port output mode registers 0, 1, 5, 8 (POM0, POM1, POM5, POM8)
- Port mode registers 0, 1, 5, 8 (PM0, PM1, PM5, PM8)
- Port registers 0, 1, 5, 8 (P0, P1, P5, P8)
- Peripheral reset control register 0 (PRR0)

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

20.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When serial array unit 0 is to be used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is to be used, be sure to set bit 3 (SAU1EN) of this register to 1.

When serial array unit 2 is to be used, be sure to set bit 1 (SAU2EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

Figure 20-6. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. ● SFR used by the serial array unit m cannot be written. The read value is 00H. However, the SFR is not initialized. Notes 1, 2, 3
1	Enables input clock supply. ● SFR used by serial array unit m can be read/written.

Notes 1. To initialize the serial array unit 1 and the SFR used by the serial array unit 1, use bit 3 (SAU1RES) of PRR0.

2. To initialize the serial array unit 0 and the SFR used by the serial array unit 0, use bit 2 (SAU0RES) of PRR0.

3. To initialize the serial array unit 2 and the SFR used by the serial array unit 2, use bit 1 (SAU2RES) of PRR0.

Cautions 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode registers 0, 1, 5, 8 (PIM0, PIM1, PIM5, PIM8), port output mode registers 0, 1, 5, 8 (POM0, POM1, POM5, POM8), port mode registers 0, 1, 5, 8 (PM0, PM1, PM5, PM8), and port registers 0, 1, 5, 8 (P0, P1, P5, P8)).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register 0 (SSC0)**

2. Be sure to clear bit 7 to “0”.

Remark m: Unit number (m = 0 to 2)

20.3.2 Serial clock select register m (SPSm)

The SPSm register 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.

Reset signal generation clears the SPSm register to 0000H.

Figure 20-7. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1), After reset: 0000H R/W
F01E6H, F01E7H (SPS2)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk)[Note]						
					f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 12 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	4 MHz	8 MHz	12 MHz	20 MHz	24 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	2 MHz	4 MHz	6 MHz	10 MHz	12 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	1 MHz	2 MHz	3 MHz	5 MHz	6 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	500 kHz	1 MHz	1.5 MHz	2.5 MHz	3 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	250 kHz	500 kHz	750 kHz	1.25 MHz	1.5 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	125 kHz	250 kHz	375 kHz	625 kHz	750 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	62.5 kHz	125 kHz	187.5 kHz	313 kHz	375 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	31.25 kHz	62.5 kHz	93.8 kHz	156 kHz	187.5 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	15.62 kHz	31.25 kHz	46.9 kHz	78.1 kHz	93.8 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	7.81 kHz	15.62 kHz	23.4 kHz	39.1 kHz	46.9 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	11.7 kHz	19.5 kHz	23.4 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	1.95 kHz	3.91 kHz	5.86 kHz	9.77 kHz	11.7 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	976 Hz	1.95 kHz	2.93 kHz	4.88 kHz	5.86 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	488 Hz	976 Hz	1.46 kHz	2.44 kHz	2.93 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	244 Hz	488 Hz	732 Hz	1.22 kHz	1.46 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	122 Hz	244 Hz	366 Hz	610 Hz	732 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (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”.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. m: Unit number (m = 0 to 2)

3. k = 0, 1

20.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (Simplified SPI (CSI), UART, or simplified I^{2}C), 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.

Reset signal generation sets the SMRmn register to 0020H.

Figure 20-8. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W
F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13),
F01D0H, F01D1H (SMR20), F01D2H, F01D3H (SMR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn^Note	0	SIS mn0^Note	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) 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 (f_{TCLK}) is generated.	

CCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in Simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for Simplified SPI (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.	

Note The SMR01, SMR03, SMR11, SMR13, and SMR21 registers only.

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, SMR12, or SMR20 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30),
q: UART number (q = 0 to 4), r: IIC number (r = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

Figure 20-8. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W
F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13),
F01D0H, F01D1H (SMR20), F01D2H, F01D3H (SMR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn[Note]	0	SIS mn0[Note]	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	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.

MD mn2	MD mn1	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

MD mn0	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 successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01, SMR03, SMR11, SMR13, and SMR21 registers only.

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, SMR12, or SMR20 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30), q: UART number (q = 0 to 4), r: IIC number (r = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

20.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register 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.

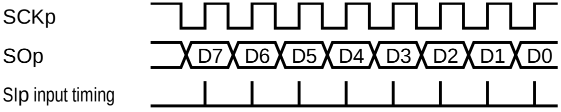
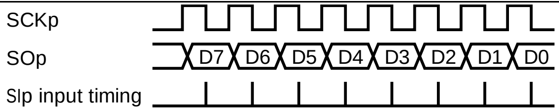
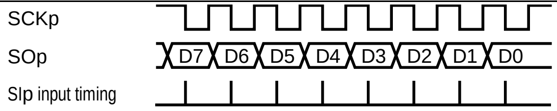
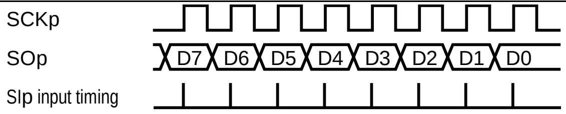
Reset signal generation sets the SCRmn register to 0087H.

Figure 20-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W
F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13),
F01D8H, F01D9H (SCR20), F01DAH, F01DBH (SCR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1[Note 1]	SLC mn0	0	1	DLSm n1[Note 2]	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^{2}C mode.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 0 to 4))
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 I^{2}C mode, and during UART transmission[Note 3].	

Notes 1. The SCR00, SCR02, SCR10, SCR12, and SCR20 registers only.

2. The SCR00, SCR01, and SCR10 registers, and the SCR11, SCR20, and SCR21 registers of 100-pin product only. Others are fixed to 1.

3. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution Be sure to clear bits 3, 6, and 11 to "0" (Also clear bit 5 of the SCR01, SCR03, SCR11, SCR13, or SCR21 register to 0). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

Figure 20-9. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03),   After reset: 0087H   R/W
F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13),
F01D8H, F01D9H (SCR20), F01DAH, F01DBH (SCR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[Note 1]	SLCmn0	0	1	DLSmn1[Note 2]	DLSmn0

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 3].	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 1]	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, 20 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 2]	DLSmn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^{2}C mode.		

Notes 1. The SCR00, SCR02, SCR10, SCR12, and SCR20 registers only.

2. The SCR00, SCR01, and SCR10 registers, and the SCR11, SCR20, and SCR21 registers of 100-pin product only. Others are fixed to 1.

3. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to "0" (Also clear bit 5 of the SCR01, SCR03, SCR11, SCR13, or SCR21 register to 0). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 30), mn = 00 to 03, 10 to 13, 20, 21

20.3.5 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00, SDR01, SDR10, SDR11, SDR20[Note], and SDR21[Note] or bits 7 to 0 (lower 8 bits) of SDR02, SDR03, SDR12[Note], 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 (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00, SDR02, and SDR12 to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 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 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

Figure 20-10. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),
FFF58H, FFF59H (SDR20), FFF5AH, FFF5BH (SDR21)

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W
FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

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

Note 100-pin product only

Cautions 1. Be sure to clear bit 8 of the SDR02, SDR03, SDR12, and SDR13 registers to “0”.

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^{2}C is used. Set SDRmn[15:9] to 0000001B or greater.

4. Rewriting SDRmn[7:0] by 8-bit memory manipulation instruction is prohibited when the operation is stopped (SEmn = 0) (all of SDRmn[15:9] are cleared (0)).

Remarks 1. For the function of the lower 8/9 bits of the SDRmn register, see **20.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

20.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 20-11. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), After reset: 0000H R/W
F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13),
F01C8H, F01C9H (SIR20), F01CAH, F01CBH (SIR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECT mn[Note]	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error flag of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01, SIR03, SIR11, SIR13, and SIR21 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, SIR12, or SIR20 register) to “0”.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

2. When the SIRmn register is read, 0000H is always read.

20.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of 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 set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 20-12. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R
F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13),
F01C0H, F01C1H (SSR20), F01C2H, F01C3H (SSR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

TSFmn	Communication status indication flag of 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	Buffer register status indication flag of 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 conditions> ● Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and 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 and reception mode in each communication mode). ● A reception error occurs.	

Note The SSR01, SSR03, SSR11, SSR13, and SSR21 registers only.

Caution When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

Figure 20-12. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R
F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13),
F01C0H, F01C1H (SSR20), F01C2H, F01C3H (SSR21)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

FEFmn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>

- 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^2C transmission).

<Clear condition>

- 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 channel at the ACK reception timing during I^2C transmission (ACK is not detected).

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>

- 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 and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in simplified SPI (CSI) mode.

Note The SSR01, SSR03, SSR11, SSR13, and SSR21 registers only.

Cautions 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.

2. When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

20.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 20-13. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	SS11	SS10

Address: F01E2H, F01E3H (SS2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS21	SS20

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status [Note].

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Cautions 1. Be sure to clear bits 15 to 4 of the SS0 register, bits 15 to 2 of the SS1 register in the 80-pin product, bits 15 to 4 of the SS1 register in the 100-pin product, and bits 15 to 2 of the SS2 register to “0”.

2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

2. When the SSm register is read, 0000H is always read.

20.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written 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.

Reset signal generation clears the STm register to 0000H.

Figure 20-14. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	ST13	ST12	ST11	ST10

Address: F01E4H, F01E5H (ST2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST21	ST20

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 Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 4 of the ST0 register, bits 15 to 2 of the ST1 register in the 80-pin product, bits 15 to 4 of the ST1 register in the 100-pin product, and bits 15 to 2 of the ST2 register to “0”.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

2. When the STm register is read, 0000H is always read.

20.3.10 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 a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value 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 set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 20-15. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	SE13	SE12	SE11	SE10

Address: F01E0H, F01E1H (SE2) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE21	SE20

SEm n	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

20.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and 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.

Reset signal generation clears the SOEm register to 0000H.

Figure 20-16. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	0	SOE00

Address: F016AH, F016BH (SOE1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10

Address: F01EAH, F01EBH (SOE2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE20

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 3 and 1 of the SOE0 register, bits 15 to 1 of the SOE1 register in the 80-pin product, bits 15 to 3 and 1 of the SOE1 register in the 100-pin product, and bits 15 to 1 of the SOE2 register to “0”.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20

20.3.12 Serial output register m (SOm)

The SOm register 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.

Reset signal generation sets the SOm register to 0F0FH.

Figure 20-17. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H (SO0) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	1	CKO 00	0	0	0	0	1	SO 02	1	SO 00

Address: F0168H, F0169H (SO1) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	CKO 12	1	1	0	0	0	0	1	SO 12	1	SO 10

Address: F01E8H, F01E9H (SO2) After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	SO 20

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

(**Caution** and **Remark** are listed on the next page.)

Caution Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register to "0". And be sure to set bits 11, 9, 3, and 1 to "1".

Be sure to clear bits 15 to 12 and 7 to 4 of the SO1 register in the 80-pin product to 0, and set bits 11 to 8 and 3 to 1 to 1. Be sure to clear bits 15 to 12 and 7 to 4 of the SO1 register in the 100-pin product to 0, and set bits 11, 9, 8, 3, and 1 to 1.

Be sure to clear bits 15 to 12 and 7 to 4 of the SO2 register in the 100-pin product to 0, and set bits 11 to 8 and 3 to 1 to 1.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20

20.3.13 Serial output level register m (SOLm)

The SOLm register is a register that 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 corresponding bit in the simplified SPI (CSI) mode and simplifies I^{2}C 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.

Reset signal generation clears the SOLm register to 0000H.

Figure 20-18. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 02	0	SOL 00

Address: F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 12	0	SOL 10

Address: F01F4H, F01F5H (SOL2) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 20

SOL mn	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.

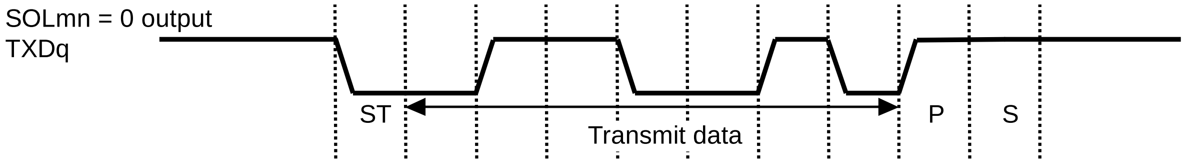
Caution Be sure to clear bits 15 to 3, and 1 of the SOL0 register, bits 15 to 1 of the SOL1 register in the 80-pin product, bits 15 to 3 and 1 of the SOL1 register in the 100-pin product, and bits 15 to 1 of the SOL2 register to "0".

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20

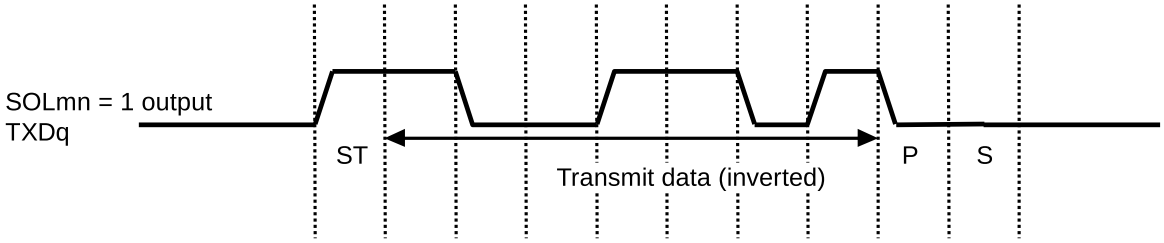
Figure 20-19 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 20-19. Examples of Reverse Transmit Data

(1) Non-reverse Output (SOLmn = 0)



(2) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20
q: UART number (q = 0 to 4)

20.3.14 Serial standby control register 0 (SSC0)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

The SSC0 register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSC0 register can be set with an 8-bit memory manipulation instruction with SSC0L.

Reset signal generation clears the SSC0 register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00 : Up to 1 Mbps**
- **When using UART0 : 4800 bps only**

Figure 20-20. Format of Serial Standby Control Register 0 (SSC0)

Address: F0138H, F0139H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS EC0	SWC 0

SS EC0	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0)
1	Stop the generation of error interrupts (INTSRE0)
● The SSECm bit can be set to 1 or 0 only when both the SWC0 and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSEC0 bit to 0. ● Setting SSEC0, SWC0 = 1, 0 is prohibited.	

SWC 0	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 A/D conversion is performed without operating the CPU (the SNOOZE mode). ● The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. ● Even when using SNOOZE mode, be sure to set the SWC0 bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWC0 bit to 0 after returning from STOP mode to normal operation mode.	

Figure 20-21. Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSR0 is generated.	INTSR0 is generated.
0	1	INTSR0 is generated.	INTSR0 is generated.
1	0	INTSR0 is generated.	INTSRE0 is generated.
1	1	INTSR0 is generated.	No interrupt is generated.

20.3.15 Input switch control register (ISC)

The ISC0 bit of the ISC register is used to realize a LIN-bus communication operation by UART0 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

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

Reset signal generation clears the ISC register to 00H.

Figure 20-22. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	0	ISC0

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 RxD0 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 1 to “0”.

20.3.16 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 I^{2}C 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.

Reset signal generation clears the NFEN0 register to 00H.

Figure 20-23. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	SNFEN40	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFENq0	Use of noise filter of RxDq pin
0	Noise filter OFF
1	Noise filter ON
Set SNFENq0 to 1 to use the RxDq pin. Clear SNFENq0 to 0 to use the other than RxDq pin.	

Caution Be sure to clear bits 7 to 5, 3, and 1 of the 80-pin product and bits 5, 3, and 1 of the 100-pin product to “0”.

Remark q: UART number (q = 0 to 4)

20.3.17 Registers controlling port functions of serial input/output pins

When using the serial array unit set the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, and **4.3.5 Port output mode registers (POMxx)**.

When using a port pin with a multiplexed serial data or serial clock output function (e.g. P07/SO00/TxD0/TI02/TO02/INTP2/TOOLTxD, P15/SEG9/(SCK00)/(SCL00)) for serial data or serial clock output, requires setting the corresponding bits in the port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (EV_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: When using P07/SO00/TxD0/TI02/TO02/INTP2/TOOLTxD for serial data output
Set the PM07 bit of the port mode register 0 to 0.
Set the P07 bit of the port register 0 to 1.

When using a port pin with a multiplexed serial data or serial clock input function (e.g. P05/SCK00/SCL00/TI04/TO04/INTP3, P06/SI00/RxD0/TI03/TO03/SDA00/TOOLRxD) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: When using P06/SI00/RxD0/TI03/TO03/SDA00/TOOLRxD for serial data input
Set the PM06 bit of port mode register 0 to 1.
Set the P06 bit of port register 0 to 0 or 1.

The PMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets the PMxx registers to FFH.
See **Tables 4-2 to 4-4** to see which PMxx registers are provided for each product.

20.3.18 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the serial array unit (SAU0, SAU1, SAU2) in the reset state, be sure to set bits 2, 3, and 1 (SAU0RES, SAU1RES, and SAU2RES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears the PRR0 register to 00H.

Figure 20-24. Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

SAUnRES	Control resetting of the serial array unit (units 0 to 2)
0	The serial array unit (unit n) is released from the reset state.
1	The serial array unit (unit n) is in the reset state.

Remark n = 0 to 2

20.4 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.

20.4.1 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 supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the 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.

To stop the operation of serial array unit 2, set bit 1 (SAU2EN) to 0.

Figure 20-25. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	0	IRDAEN ×	ADCEN ×	IICA0EN ×	SAU1EN 0/1	SAU0EN 0/1	SAU2EN 0/1	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read

Note that this does not apply to the following registers.

- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode register (PIMx)**
- **Port output mode register (POMx)**
- **Port mode register (PMx)**
- **Port register (Px)**

2. Be sure to clear bit 7 to 0.

Remark ×: 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
m: Unit number (m = 0 to 2)

20.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 20-26. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	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	0	SOEm2 0/1	0	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to "1".

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3)

2. ▨ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

20.5 Operation of Simplified SPI (CSI00, CSI10, CSI30) 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. $f_{MCK}/2$

  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. CSI00 supports the asynchronous reception.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**.

The channel supporting simplified SPI (CSI00) is channel 0 of SAU0.
The channel supporting simplified SPI (CSI10) is channel 2 of SAU0.
The channel supporting simplified SPI (CSI30) is channel 2 of SAU1.

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2 (supporting IrDA)	–
	1	–		–
	2	CSI30[Note]	UART3[Note]	IIC30[Note]
	3	–		–
2	0	–	UART4[Note]	–
	1	–		–

Note 100-pin product only

Simplified SPI (CSI00, CSI10, CSI30) performs the following seven types of communication operations.

● Master transmission	(See **20.5.1**.)
● Master reception	(See **20.5.2**.)
● Master transmission/reception	(See **20.5.3**.)
● Slave transmission	(See **20.5.4**.)
● Slave reception	(See **20.5.5**.)
● Slave transmission/reception	(See **20.5.6**.)
● SNOOZE mode function (CSI00 only)	(See **20.5.7**.)

20.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SO00	SCK10, SO10	SCK30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	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] 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 before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register ● CKPmn = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK) ● CKPmn = 1: Reverse (data output at the rising edge and data input at the falling edge of SCK)		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-27. Example of Contents of Registers for Master Transmission of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm						CKOm2		CKOm0						SOm2		SOm0
	0	0	0	0	1	0/1	1	0/1	0	0	0	0	1	0/1	1	0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

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

Figure 20-27. Example of Contents of Registers for Master Transmission of simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks **1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

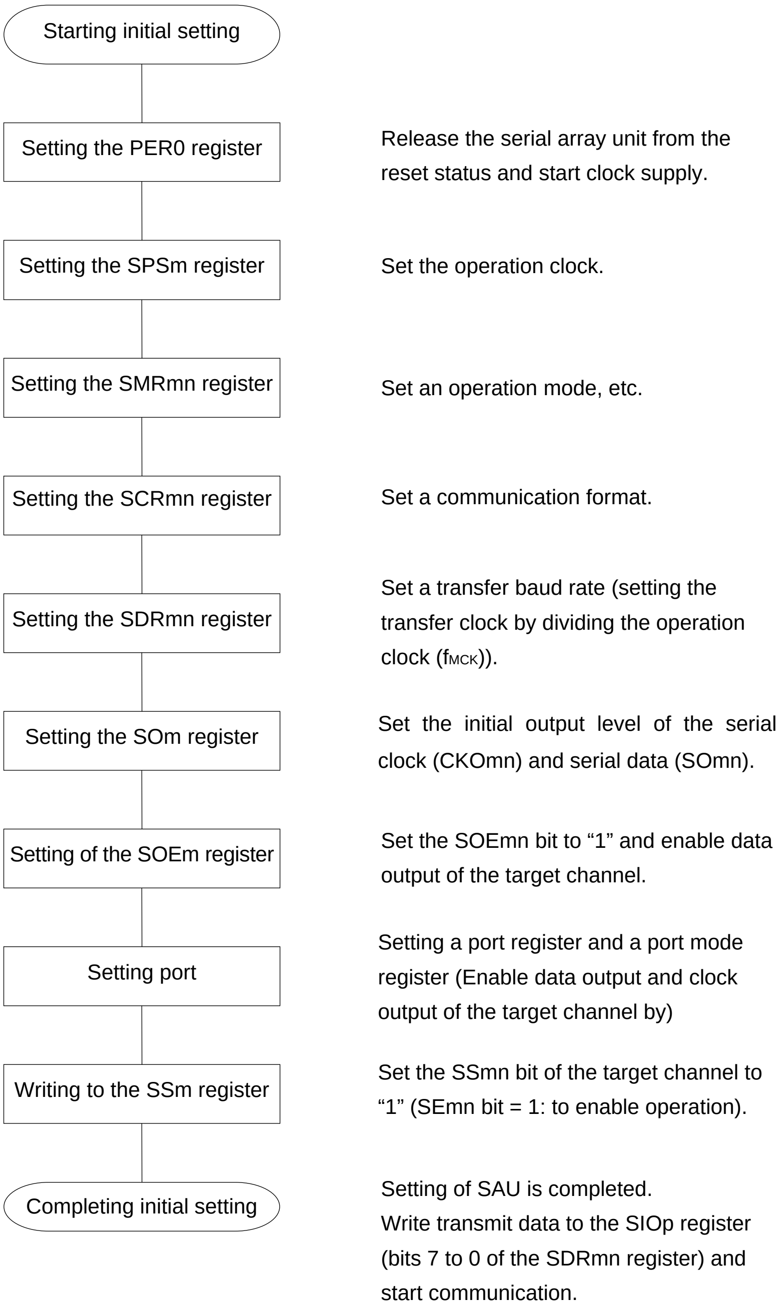
2. ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-28. Initial Setting Procedure for Master Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

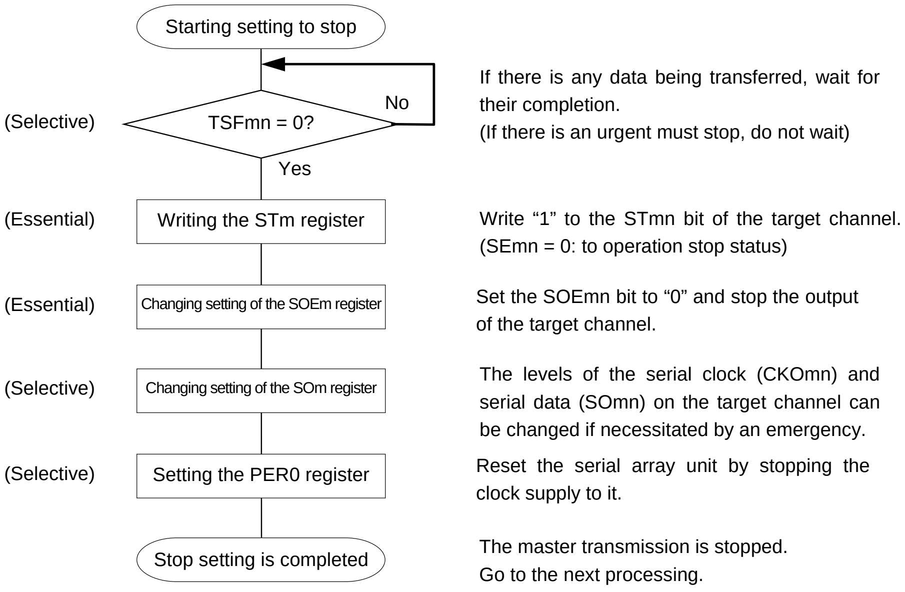
Set the SOEmn bit to “1” and enable data output of the target channel.

Setting a port register and a port mode register (Enable data output and clock output of the target channel by)

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Setting of SAU is completed.
Write transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 20-29. Procedure for Stopping Master Transmission


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

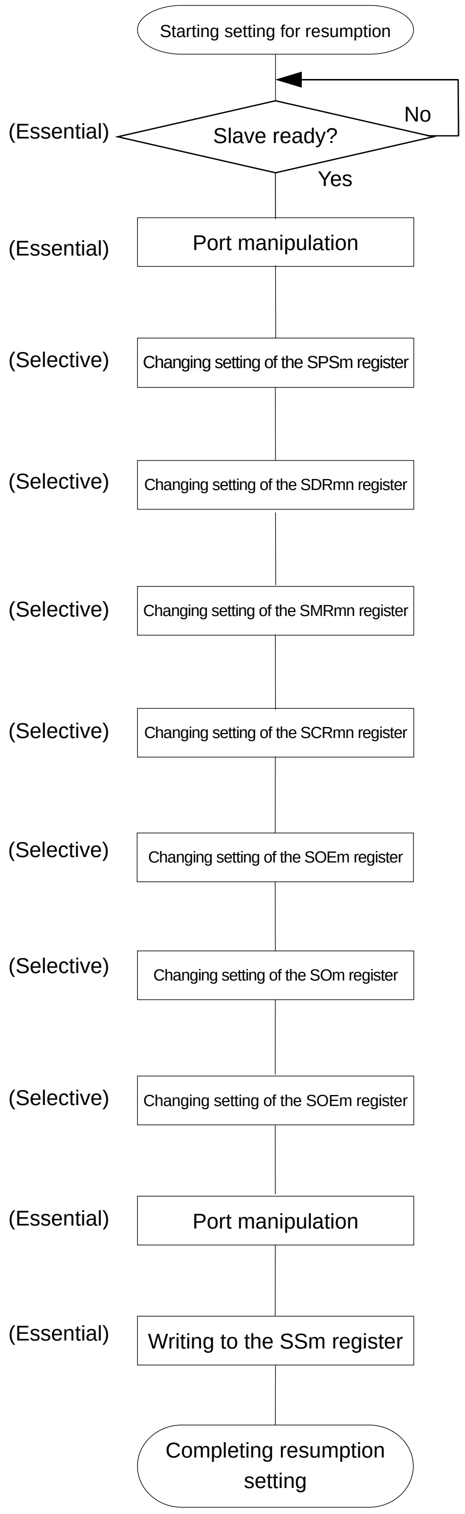
Set the SOEmn bit to “0” and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-30. Procedure for Resuming Master Transmission


Starting setting for resumption
(Essential)
Slave ready?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the SOEmn bit to “0” to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

Set the SOEmn bit to “1” and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

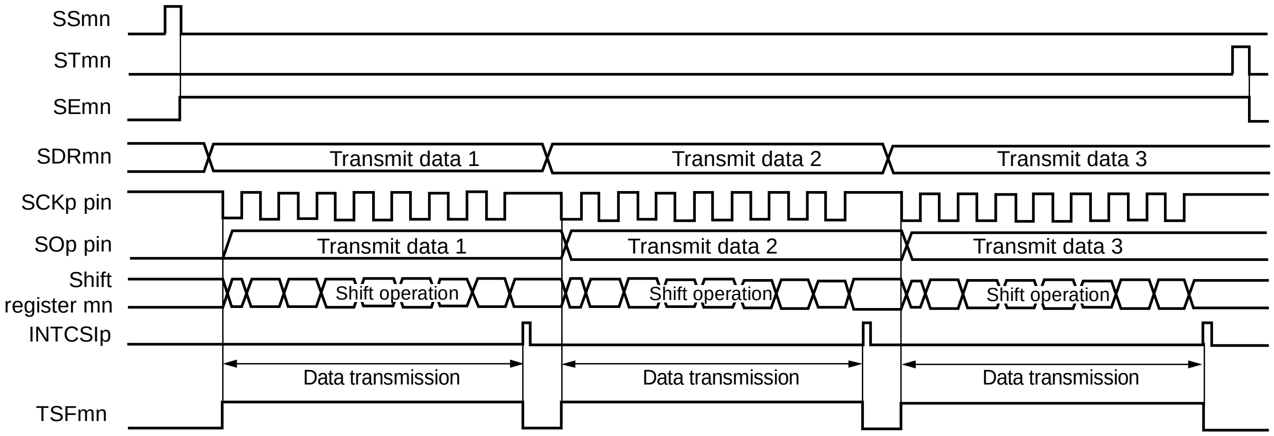
Set the SSmn bit of the target channel to “1” (SEmn = 1: to enable operation).

Setting is completed
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

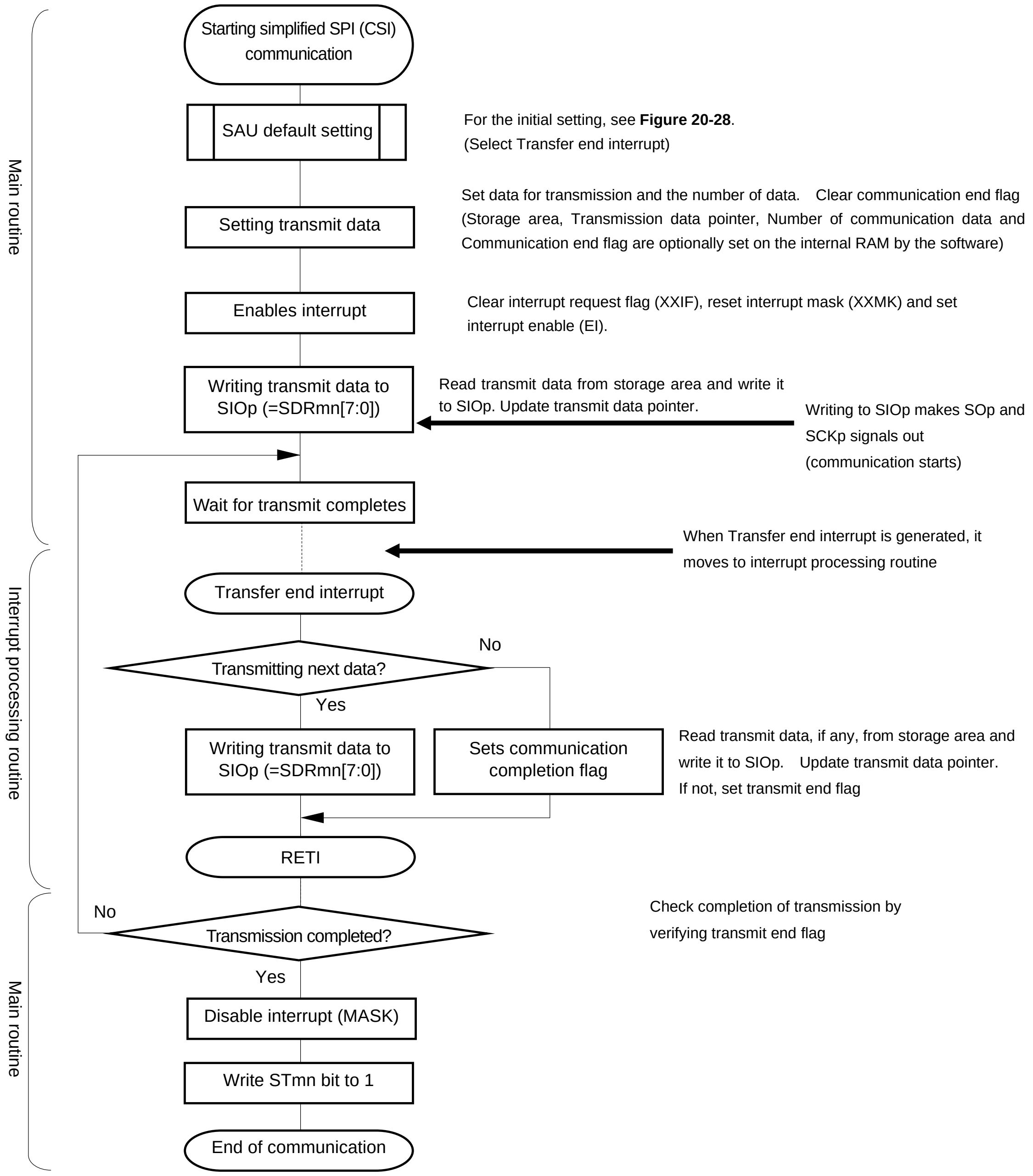
(3) Processing flow (in single-transmission mode)

Figure 20-31. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



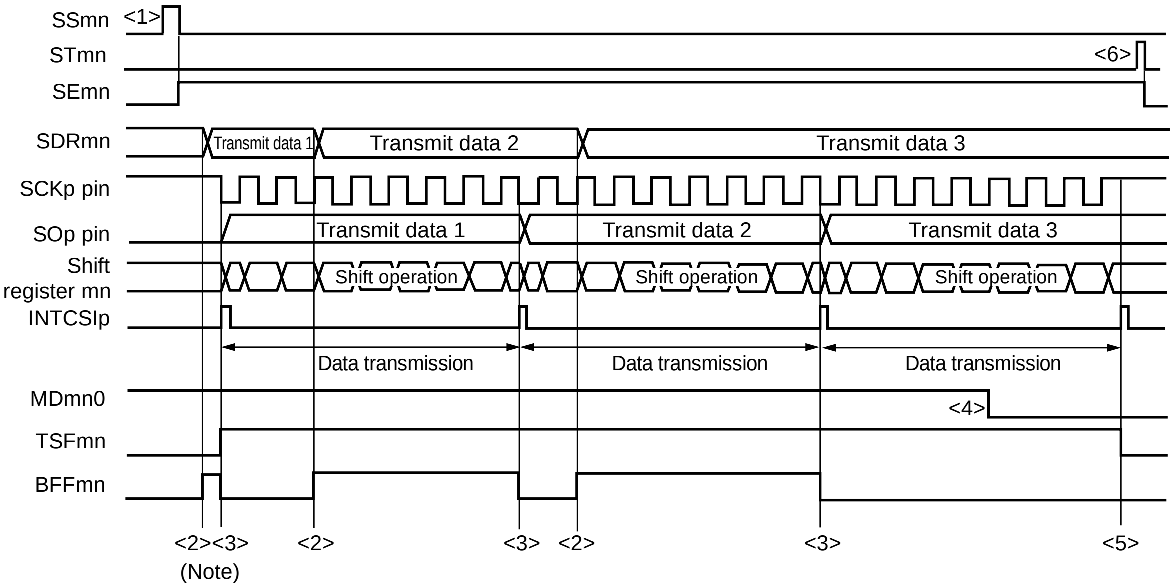
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-32. Flowchart of Master Transmission (in Single-Transmission Mode)


Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, see **Figure 20-28**.
(Select Transfer end interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp and SCKp signals out (communication starts)
Wait for transmit completes
When Transfer end interrupt is generated, it moves to interrupt processing routine
Transfer end interrupt
Transmitting next data?
No
Yes
Writing transmit data to SIOp (=SDRmn[7:0])
Sets communication completion flag
Read transmit data, if any, from storage area and write it to SIOp. Update transmit data pointer.
If not, set transmit end flag
RETI
No
Transmission completed?
Check completion of transmission by verifying transmit end flag
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

(4) Processing flow (in continuous transmission mode)

Figure 20-33. Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

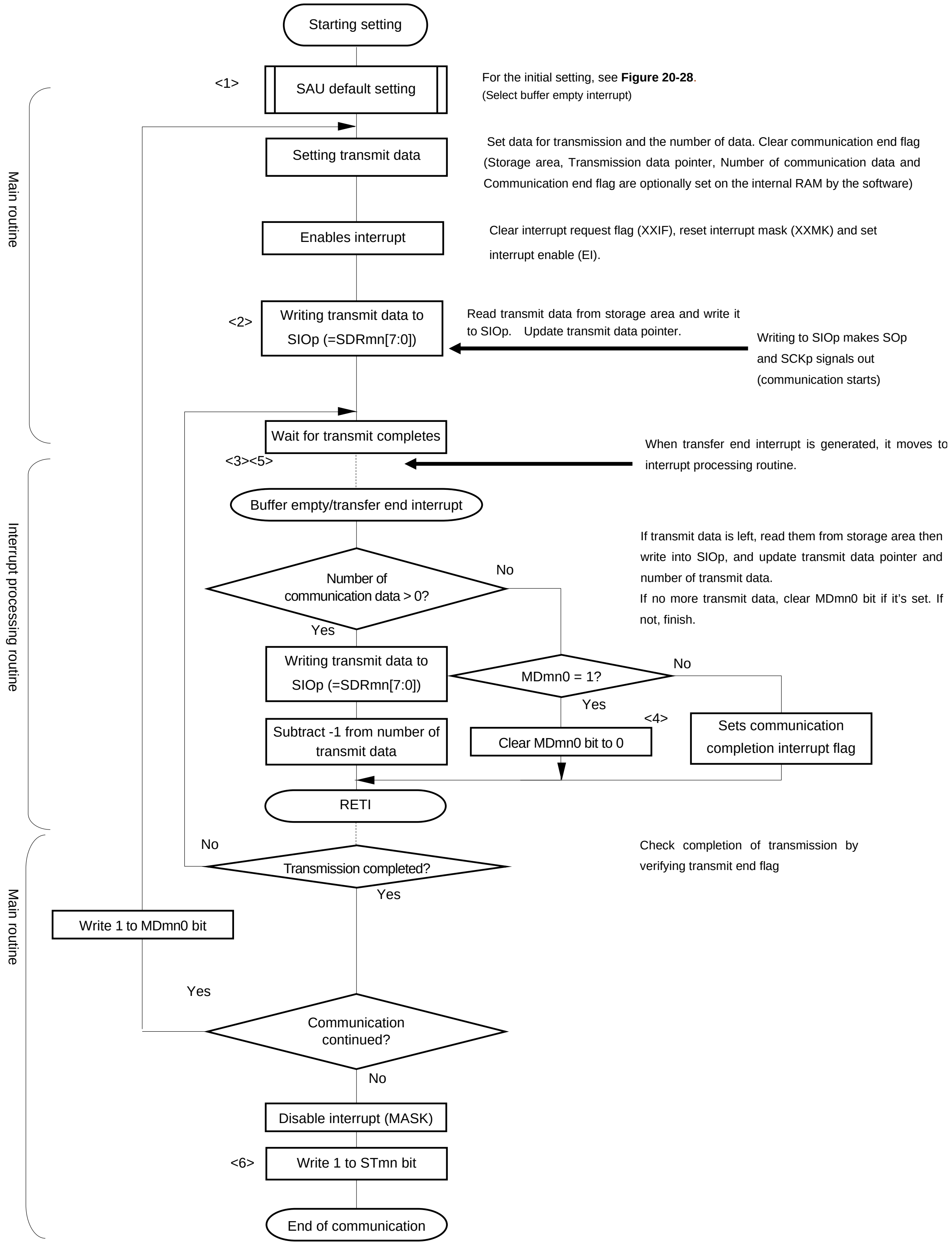


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-34. Flowchart of Master Transmission (in Continuous Transmission Mode)


Starting setting
<1>
SAU default setting
For the initial setting, see **Figure 20-28**.
(Select buffer empty interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
<2>
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp and SCKp signals out (communication starts)
Main routine
Wait for transmit completes
<3><5>
When transfer end interrupt is generated, it moves to interrupt processing routine.
Buffer empty/transfer end interrupt
Interrupt processing routine
Number of communication data > 0?
No
Yes
If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer and number of transmit data.
If no more transmit data, clear MDmn0 bit if it's set. If not, finish.
Writing transmit data to SIOp (=SDRmn[7:0])
MDmn0 = 1?
No
Yes
<4>
Subtract -1 from number of transmit data
Clear MDmn0 bit to 0
Sets communication completion interrupt flag
RETI
No
Transmission completed?
Yes
Check completion of transmission by verifying transmit end flag
Main routine
Write 1 to MDmn0 bit
Yes
Communication continued?
No
Disable interrupt (MASK)
<6>
Write 1 to STmn bit
End of communication

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 20-33 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

20.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SI00	SCK10, SI10	SCK30, SI30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate[Note]	Max. $f_{CLK}/2$ [Hz] 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 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 in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-35. Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	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 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)	0	Receive data (Write FFH as dummy data.)

Bits 7–0: SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	1	SOm0 ×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

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

Figure 20-35. Example of Contents of Registers for Master Reception of simplified SPI (CSI00) (2/2)

(e) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

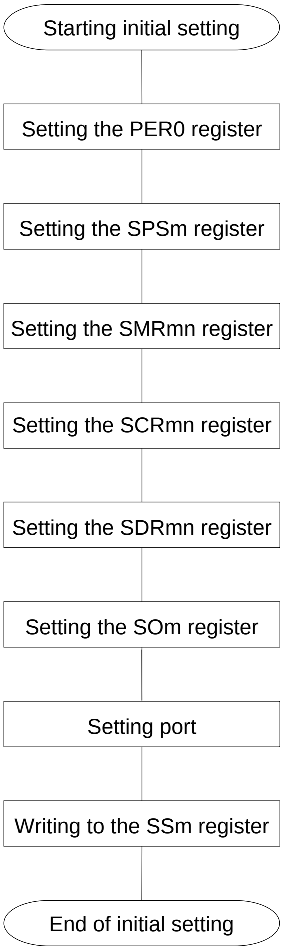
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-36. Initial Setting Procedure for Master Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Setting port
Writing to the SSm register
End of initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

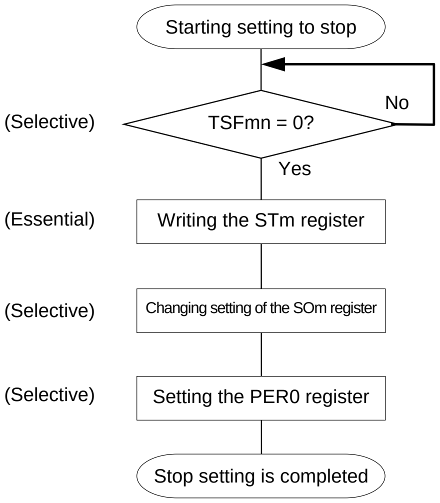
Set the initial output level of the serial clock (CKOmn).

Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Set dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 20-37. Procedure for Stopping Master Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

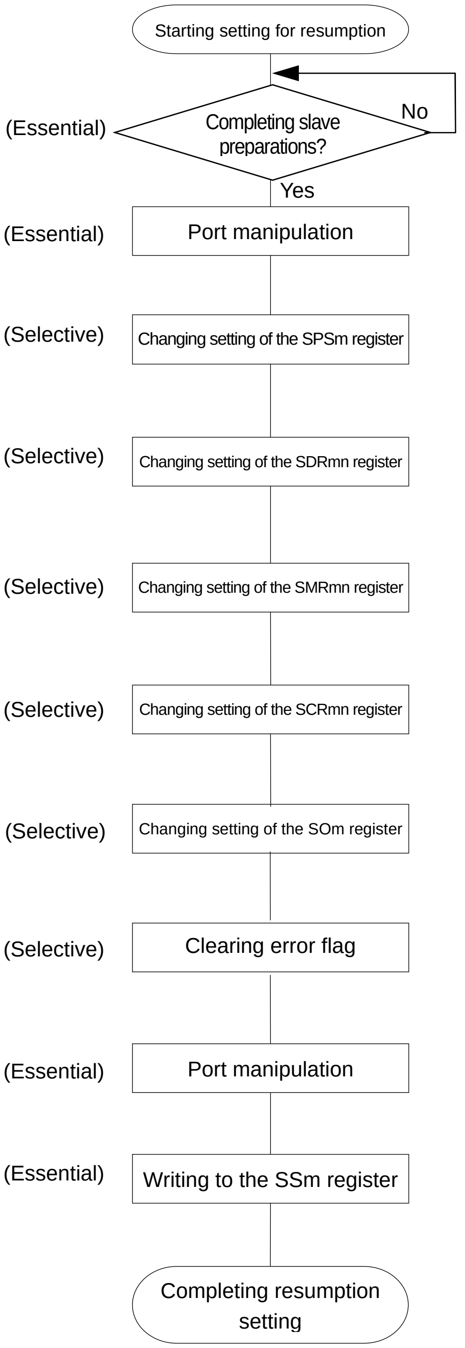
Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

The levels of the serial clock (CKOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-38. Procedure for Resuming Master Reception


Starting setting for resumption
(Essential)
Completing slave preparations?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOm register
(Selective)
Clearing error flag
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target (slave or communication operation completed

Disable clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the initial output level of the serial clock (CKOmn).

If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).

Enable clock output of the target channel by setting a port register and a port mode register.

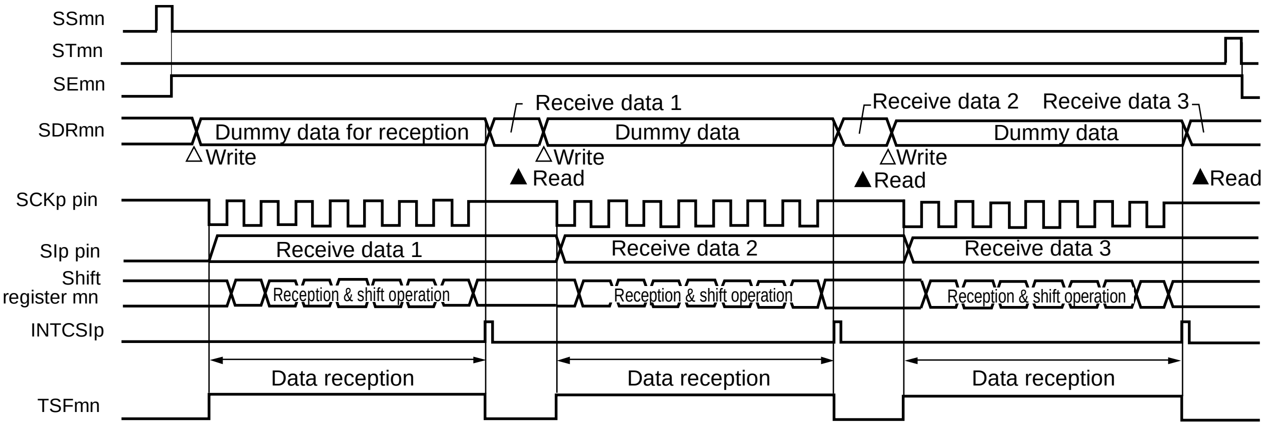
Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Setting is completed
Sets dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

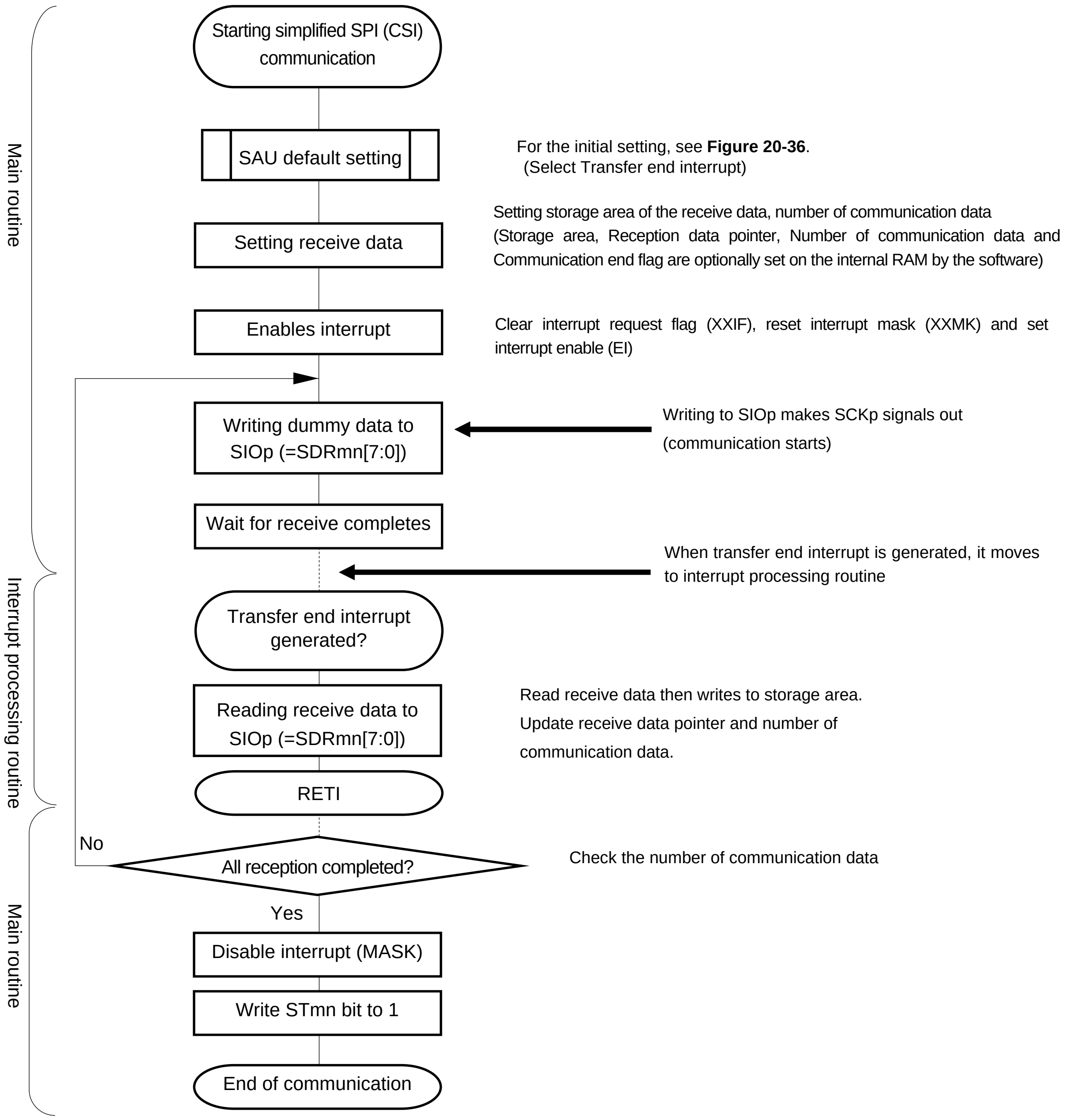
(3) Processing flow (in single-reception mode)

Figure 20-39. Timing Chart of Master Reception (in Single-reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-40. Flowchart of Master Reception (in Single-reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
Setting receive data
Enables interrupt
Writing dummy data to SIOp (=SDRmn[7:0])
Wait for receive completes
Transfer end interrupt generated?
Reading receive data to SIOp (=SDRmn[7:0])
RETI
No
All reception completed?
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

For the initial setting, see **Figure 20-36**.
(Select Transfer end interrupt)

Setting storage area of the receive data, number of communication data (Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Writing to SIOp makes SCKp signals out (communication starts)

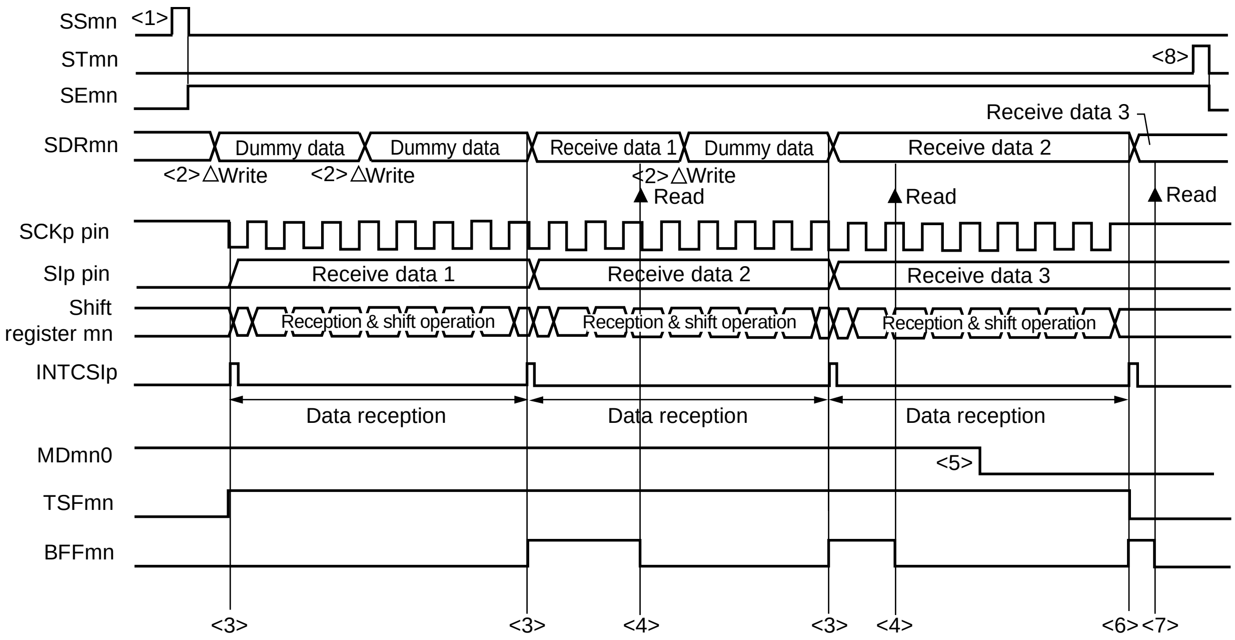
When transfer end interrupt is generated, it moves to interrupt processing routine

Read receive data then writes to storage area. Update receive data pointer and number of communication data.

Check the number of communication data

(4) Processing flow (in continuous reception mode)

Figure 20-41. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

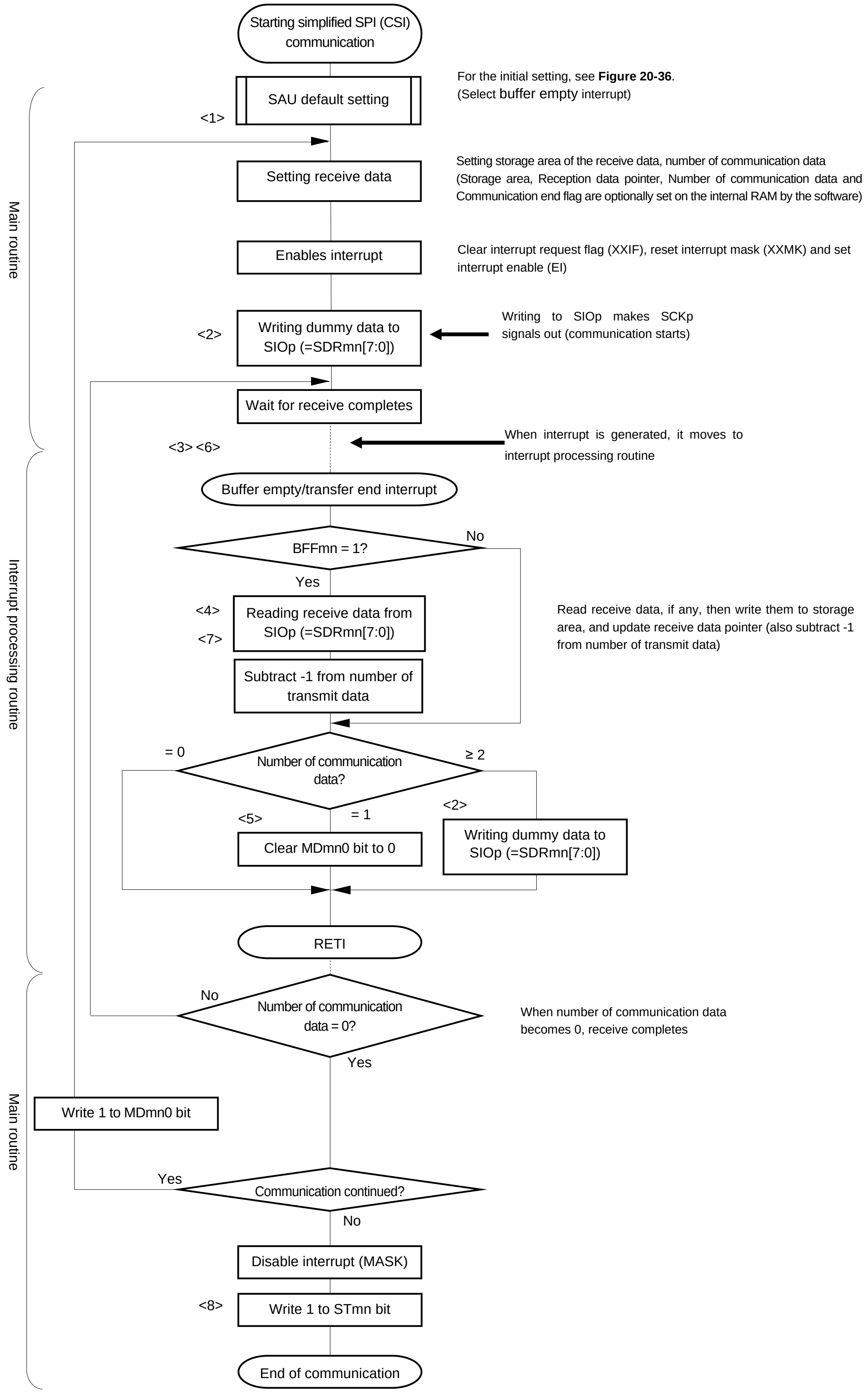


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 20-42 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-42. Flowchart of Master Reception (in Continuous Reception Mode)



Starting simplified SPI (CSI) communication
<1>
SAU default setting
For the initial setting, see **Figure 20-36**.
(Select buffer empty interrupt)
Main routine
Setting receive data
Setting storage area of the receive data, number of communication data
(Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing dummy data to SIOp (=SDRmn[7:0])
Writing to SIOp makes SCKp signals out (communication starts)
Wait for receive completes
<3> <6>
When interrupt is generated, it moves to interrupt processing routine
Interrupt processing routine
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
<7>
Reading receive data from SIOp (=SDRmn[7:0])
Read receive data, if any, then write them to storage area, and update receive data pointer (also subtract -1 from number of transmit data)
Subtract -1 from number of transmit data
= 0
Number of communication data?
≥ 2
<2>
<5>
= 1
Clear MDmn0 bit to 0
Writing dummy data to SIOp (=SDRmn[7:0])
RETI
No
Number of communication data = 0?
When number of communication data becomes 0, receive completes
Yes
Main routine
Write 1 to MDmn0 bit
Yes
Communication continued?
No
Disable interrupt (MASK)
<8>
Write 1 to STmn bit
End of communication

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 20-41 Timing Chart of Master Reception (in Continuous Reception Mode)**.

20.5.3 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	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10	SCK30, SI30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate[Note]	Max. $f_{CLK}/2$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency		
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. ● DAPmn = 1: Data I/O starts half a clock 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 in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-43. Example of Contents of Registers for Master Transmission/Reception of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	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 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							
									SIOp							

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reverse (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

2. □: Setting is fixed in the simplified SPI (CSI) master 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

Figure 20-43. Example of Contents of Registers for Master Transmission/Reception of simplified SPI (CSI00, CSI10, CSI30) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm														SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

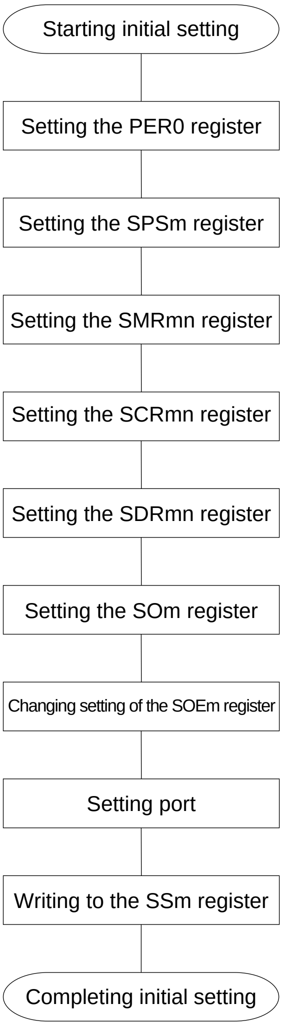
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-44. Initial Setting Procedure for Master Transmission/Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

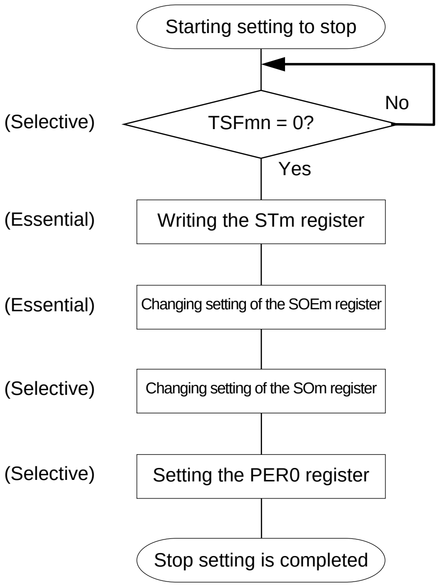
Set the SOEmn bit to “1” and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 20-45. Procedure for Stopping Master Transmission/Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

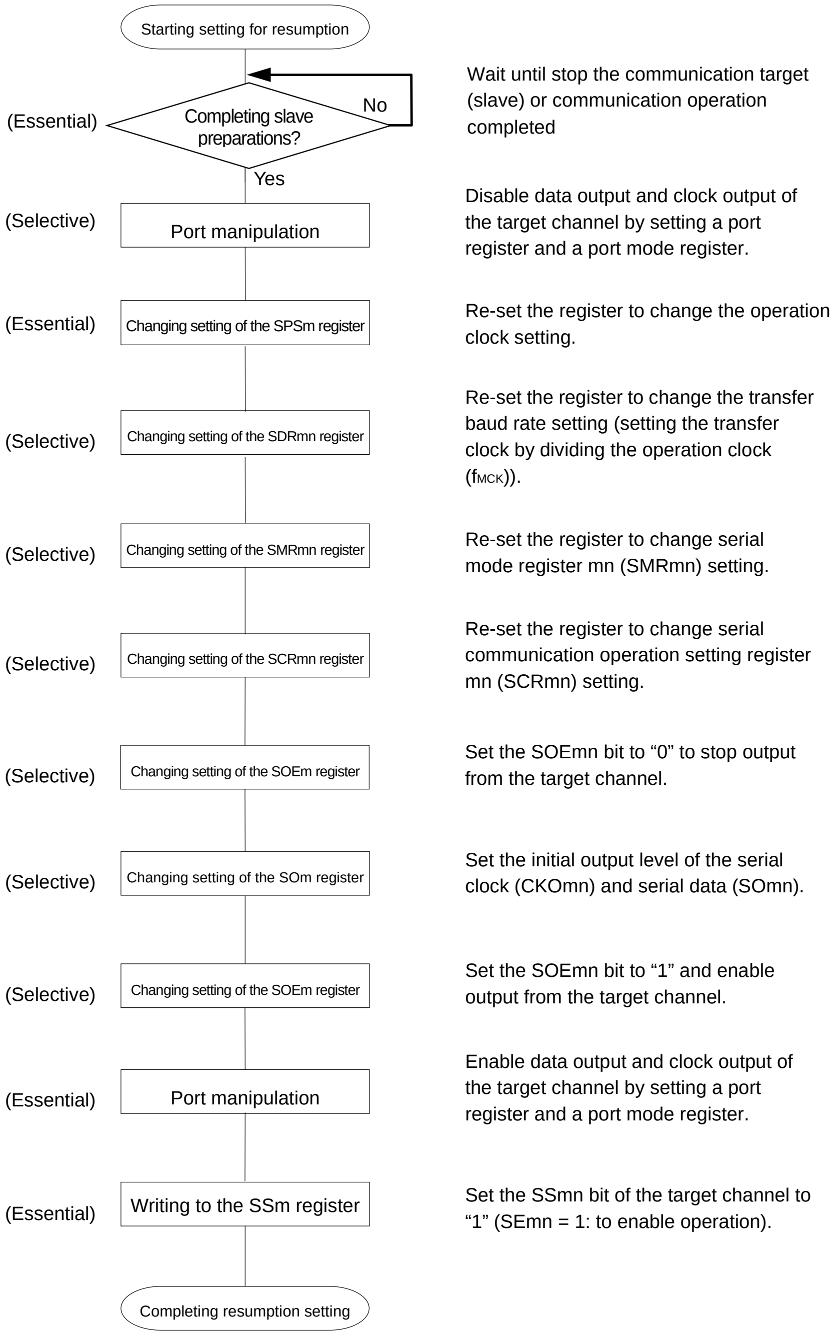
Set the SOEmn bit to “0” and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-46. Procedure for Resuming Master Transmission/Reception


Starting setting for resumption
(Essential)
Completing slave preparations?
No
Yes
(Selective)
Port manipulation
(Essential)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the SOEmn bit to “0” to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

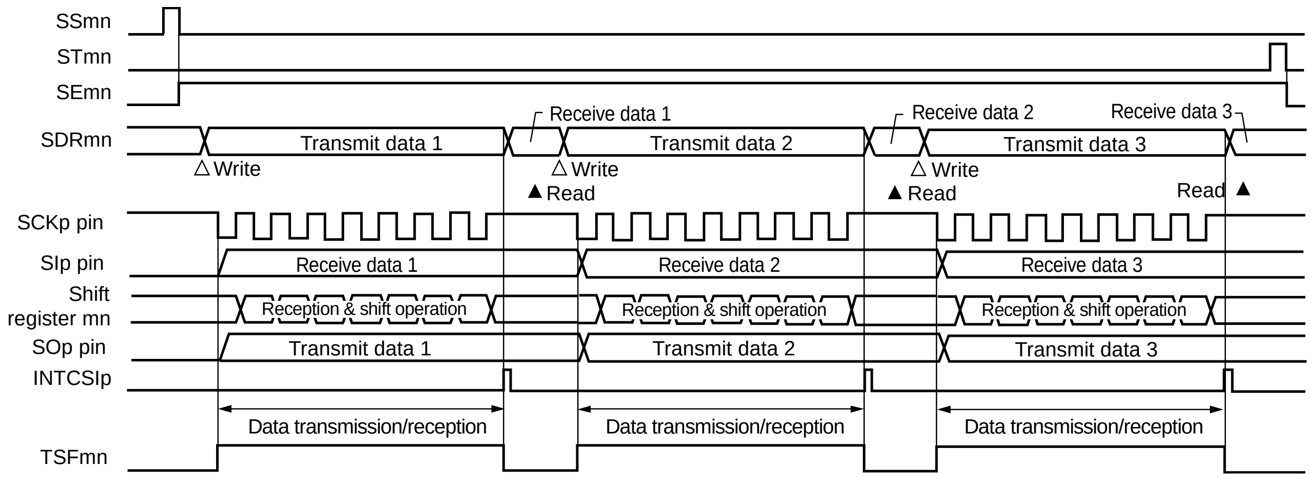
Set the SOEmn bit to “1” and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn = 1: to enable operation).

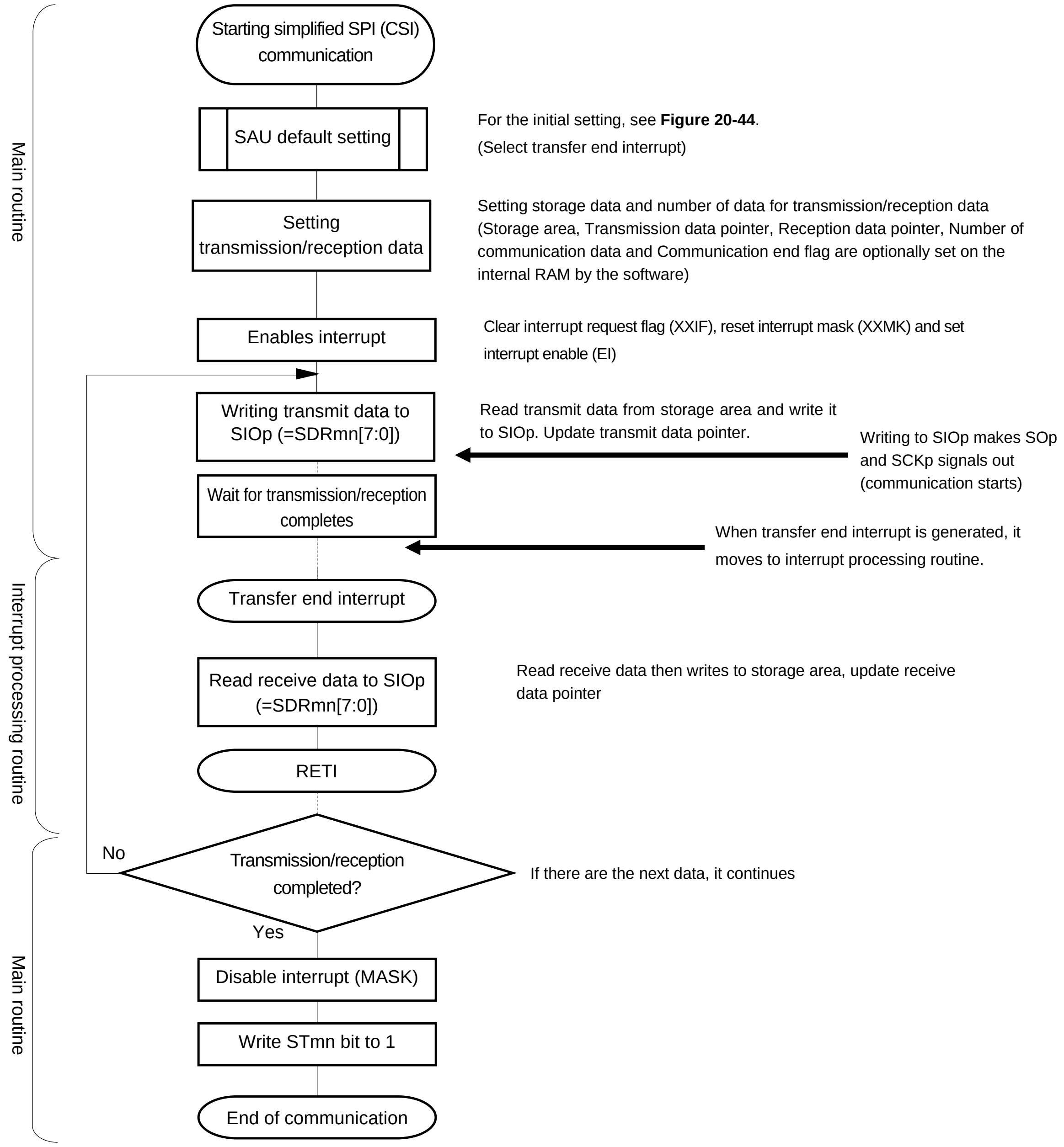
(3) Processing flow (in single-transmission/reception mode)

Figure 20-47. Timing Chart of Master Transmission/Reception (in Single-Transmission/reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-48. Flowchart of Master Transmission/Reception (in Single-Transmission/reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
Setting transmission/reception data
Enables interrupt
Writing transmit data to SIOp (=SDRmn[7:0])
Wait for transmission/reception completes
Transfer end interrupt
Read receive data to SIOp (=SDRmn[7:0])
RETI
Transmission/reception completed?
No
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

For the initial setting, see **Figure 20-44**.
(Select transfer end interrupt)

Setting storage data and number of data for transmission/reception data (Storage area, Transmission data pointer, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Writing to SIOp makes SOp and SCKp signals out (communication starts)

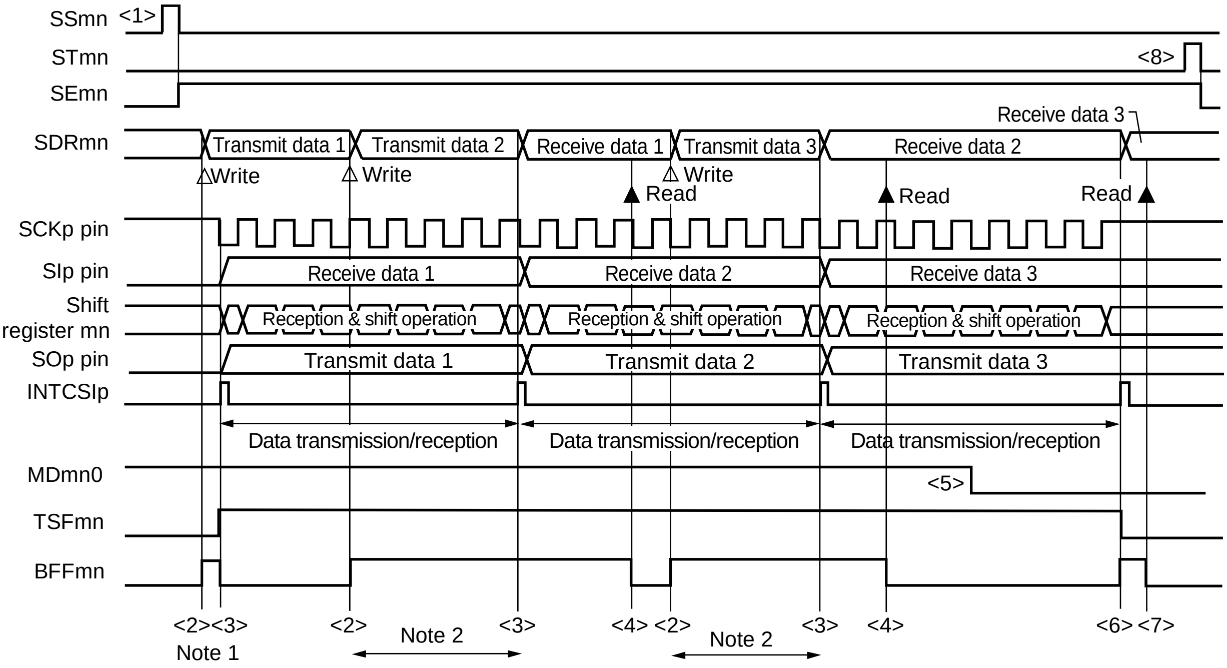
When transfer end interrupt is generated, it moves to interrupt processing routine.

Read receive data then writes to storage area, update receive data pointer

If there are the next data, it continues

(4) Processing flow (in continuous transmission/reception mode)

Figure 20-49. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

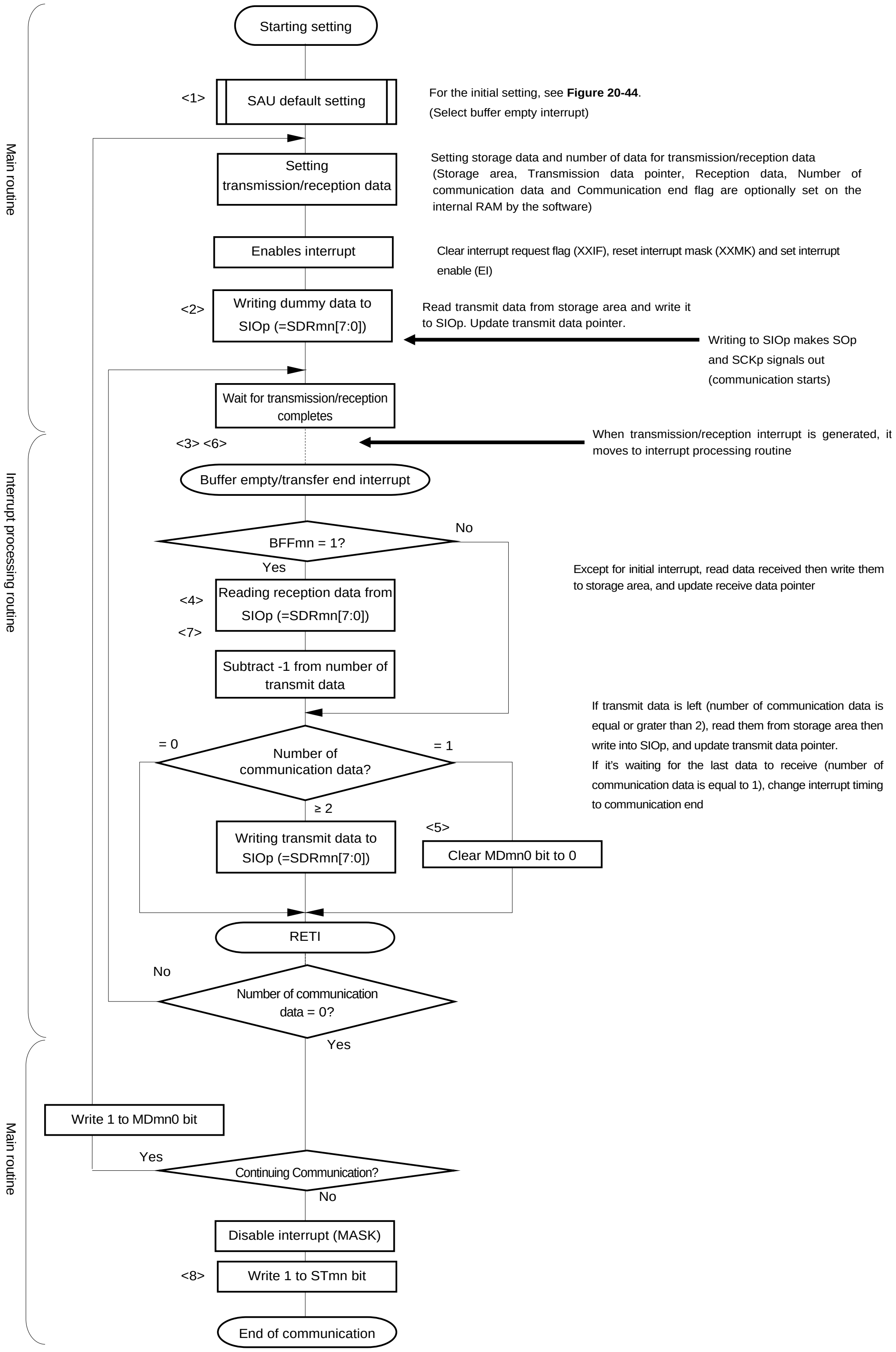
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) 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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 20-50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-50. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)


Main routine
Starting setting
<1>
SAU default setting
For the initial setting, see **Figure 20-44**.
(Select buffer empty interrupt)
Setting transmission/reception data
Setting storage data and number of data for transmission/reception data (Storage area, Transmission data pointer, Reception data, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing dummy data to SIOp (=SDRmn[7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp and SCKp signals out (communication starts)
Wait for transmission/reception completes
<3> <6>
When transmission/reception interrupt is generated, it moves to interrupt processing routine
Interrupt processing routine
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
Except for initial interrupt, read data received then write them to storage area, and update receive data pointer
<4>
Reading reception data from SIOp (=SDRmn[7:0])
<7>
Subtract -1 from number of transmit data
If transmit data is left (number of communication data is equal or grater than 2), read them from storage area then write into SIOp, and update transmit data pointer.
If it's waiting for the last data to receive (number of communication data is equal to 1), change interrupt timing to communication end
= 0
Number of communication data?
= 1
≥ 2
Writing transmit data to SIOp (=SDRmn[7:0])
<5>
Clear MDmn0 bit to 0
RETI
No
Number of communication data = 0?
Yes
Main routine
Write 1 to MDmn0 bit
Yes
Continuing Communication?
No
Disable interrupt (MASK)
<8>
Write 1 to STmn bit
End of communication

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 20-49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

20.5.4 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	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SO00	SCK10, SO10	SCK30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	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 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		

Notes 1. Because the external serial clock input to the SCK00, SCK10, and SCK30 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-51. Example of Contents of Registers for Slave Transmission of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f$_{MCK}$) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting							
									SIOp							

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

2. □: Setting is fixed in the simplified SPI (CSI) slave transmission mode,
▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 20-51. Example of Contents of Registers for Slave Transmission of simplified SPI (CSI00, CSI10, CSI30) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

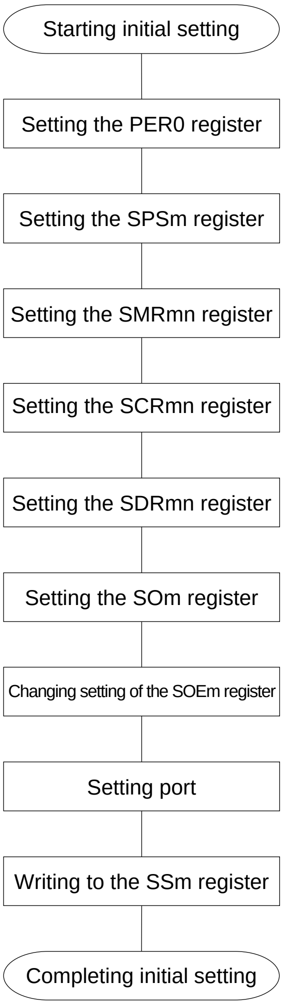
2. ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-52. Initial Setting Procedure for Slave Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to “1” and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Figure 20-53. Procedure for Stopping Slave Transmission

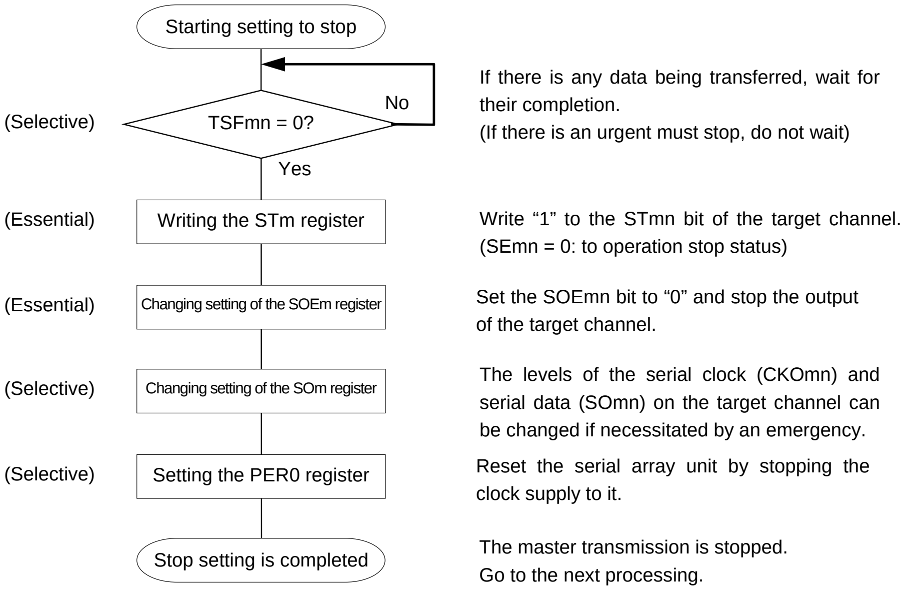
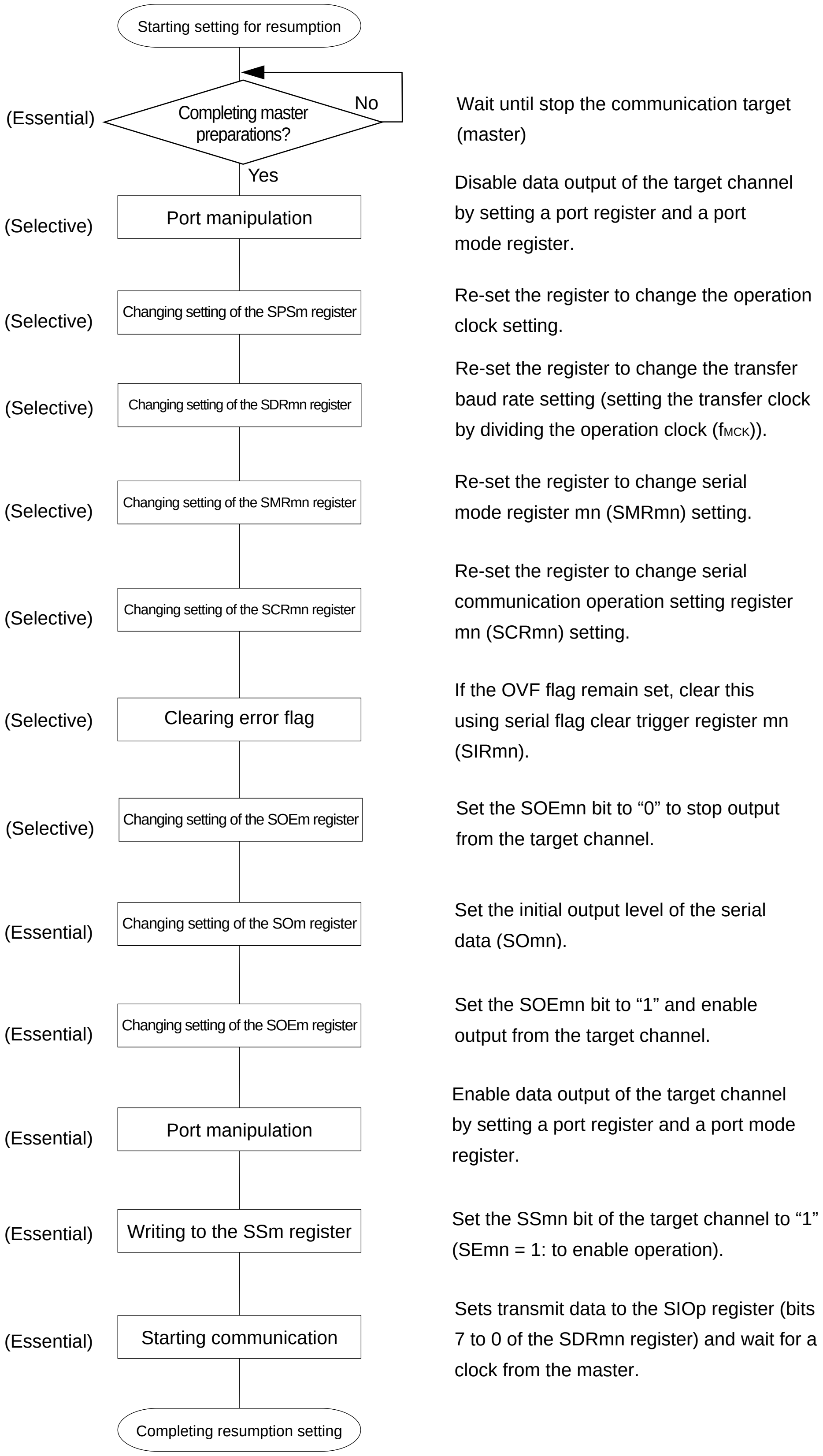

Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)
(Essential)
Writing the STm register
Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to “0” and stop the output of the target channel.
(Selective)
Changing setting of the SOm register
The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.
(Selective)
Setting the PER0 register
Reset the serial array unit by stopping the clock supply to it.
Stop setting is completed
The master transmission is stopped.
Go to the next processing.

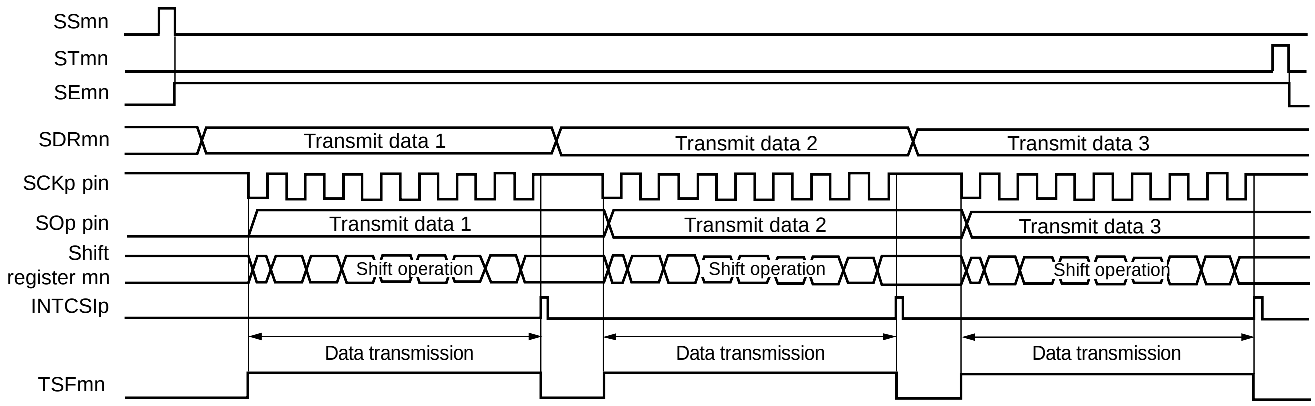
Figure 20-54. Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
Wait until stop the communication target (master)
(Selective)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SDRmn register
Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).
(Selective)
Changing setting of the SOEm register
Set the SOEmn bit to “0” to stop output from the target channel.
(Essential)
Changing setting of the SOm register
Set the initial output level of the serial data (SOmn).
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to “1” and enable output from the target channel.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to “1” (SEmn = 1: to enable operation).
(Essential)
Starting communication
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.
Completing resumption setting

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

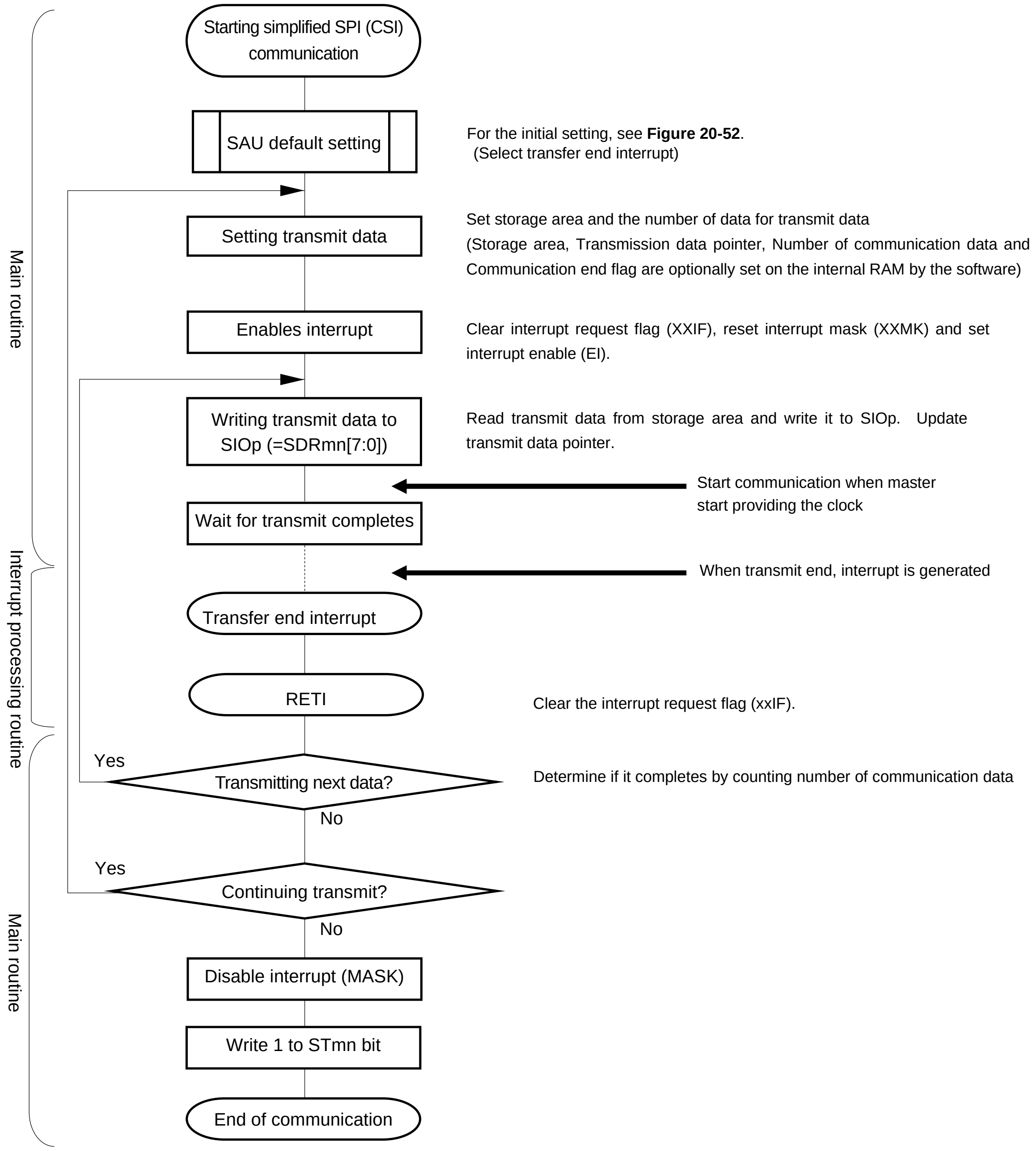
(3) Processing flow (in single-transmission mode)

Figure 20-55. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



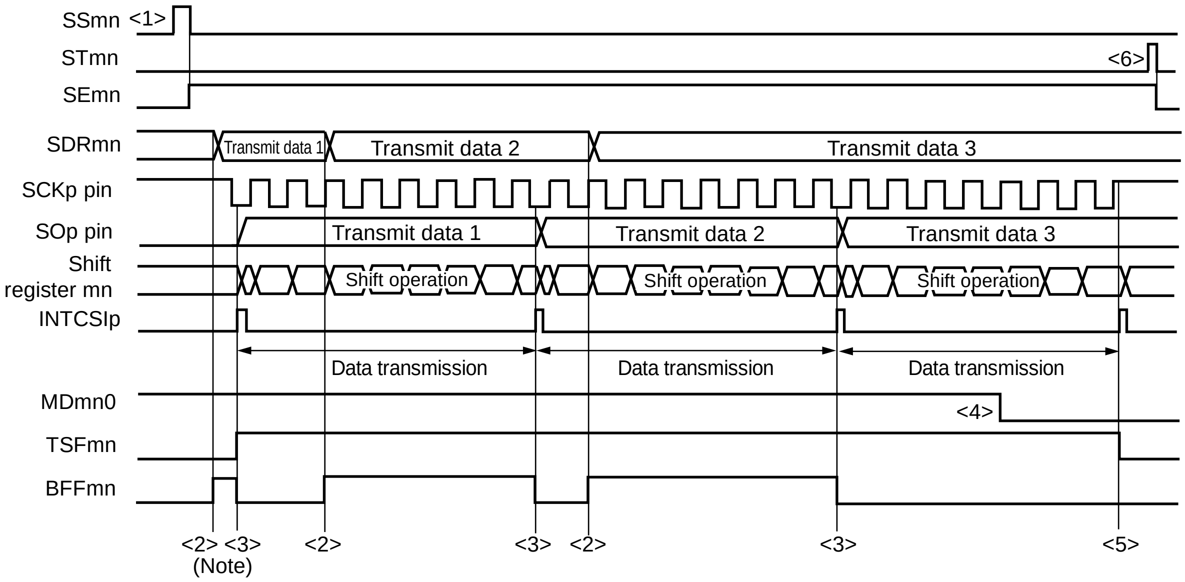
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-56. Flowchart of Slave Transmission (in Single-Transmission Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, see **Figure 20-52**.
(Select transfer end interrupt)
Setting transmit data
Set storage area and the number of data for transmit data
(Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Start communication when master start providing the clock
Wait for transmit completes
When transmit end, interrupt is generated
Interrupt processing routine
Transfer end interrupt
RETI
Clear the interrupt request flag (xxIF).
Main routine
Yes
Transmitting next data?
Determine if it completes by counting number of communication data
No
Yes
Continuing transmit?
No
Disable interrupt (MASK)
Write 1 to STmn bit
End of communication

(4) Processing flow (in continuous transmission mode)

Figure 20-57. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

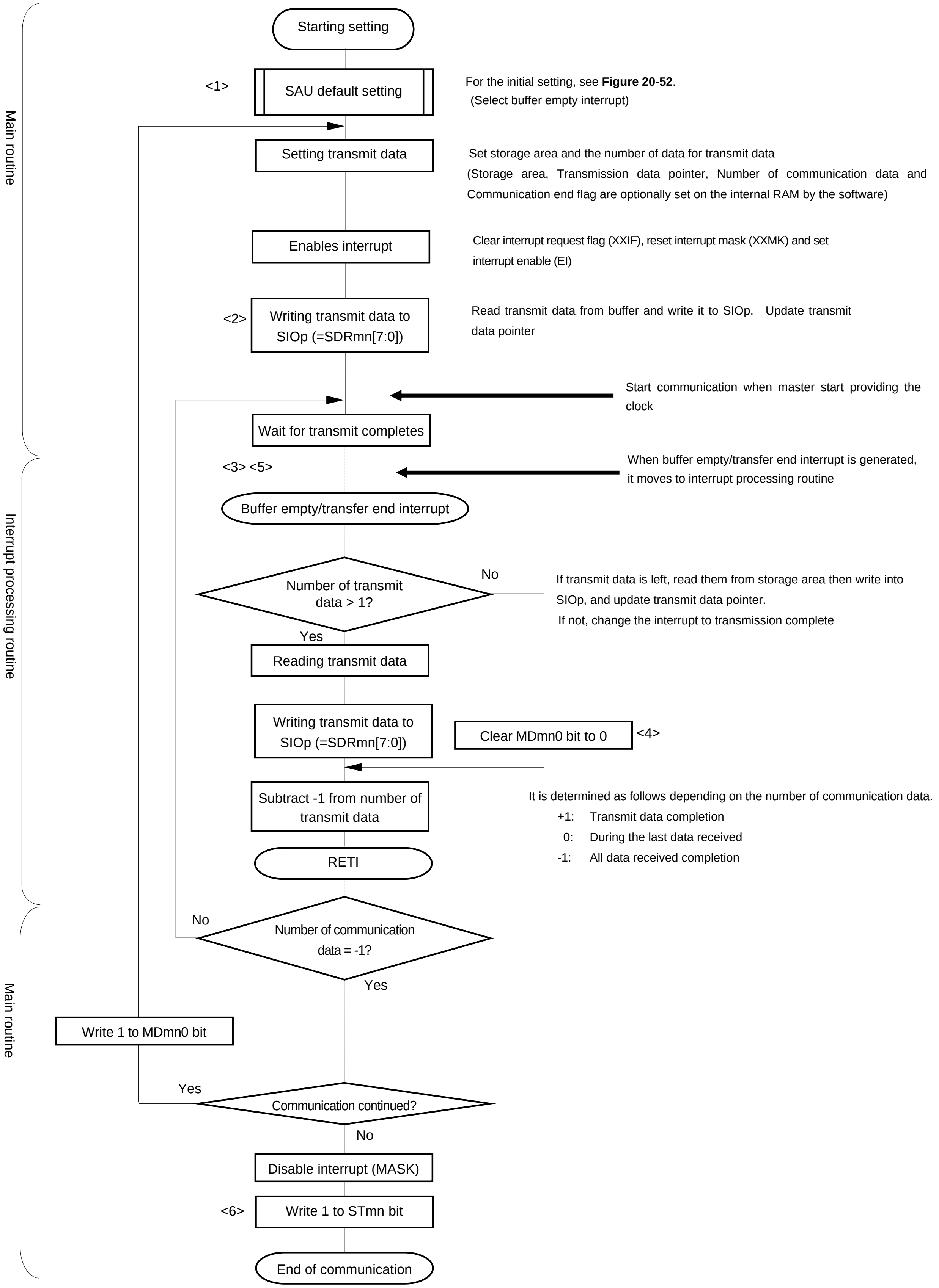


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) 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, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-58. Flowchart of Slave Transmission (in Continuous Transmission Mode)


Main routine
Starting setting
<1>
SAU default setting
For the initial setting, see **Figure 20-52**.
(Select buffer empty interrupt)
Setting transmit data
Set storage area and the number of data for transmit data
(Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from buffer and write it to SIOp. Update transmit data pointer
Start communication when master start providing the clock
Wait for transmit completes
<3> <5>
When buffer empty/transfer end interrupt is generated, it moves to interrupt processing routine
Interrupt processing routine
Buffer empty/transfer end interrupt
Number of transmit data > 1?
No
Yes
If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer.
If not, change the interrupt to transmission complete
Reading transmit data
Writing transmit data to SIOp (=SDRmn[7:0])
Clear MDmn0 bit to 0
<4>
Subtract -1 from number of transmit data
It is determined as follows depending on the number of communication data.
+1: Transmit data completion
0: During the last data received
-1: All data received completion
RETI
Main routine
No
Number of communication data = -1?
Yes
Write 1 to MDmn0 bit
Yes
Communication continued?
No
Disable interrupt (MASK)
<6>
Write 1 to STmn bit
End of communication

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 20-57 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

20.5.5 Slave reception

Slave reception is that the RL78 microcontroller receive data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SI00	SCK10, SI10	SCK30, SI30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	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 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		

Notes 1. Because the external serial clock input to the SCK00, SCK10, and SCK30 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-59. Example of Contents of Registers for Slave Reception of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							
									SIOp							

(d) Serial output register m (SOm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

2. □ : Setting is fixed in the simplified SPI (CSI) slave transmission mode,
▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 20-59. Example of Contents of Registers for Slave Reception of simplified SPI (CSI00, CSI10, CSI30) (2/2)

(e) Serial output enable register m (SOEm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

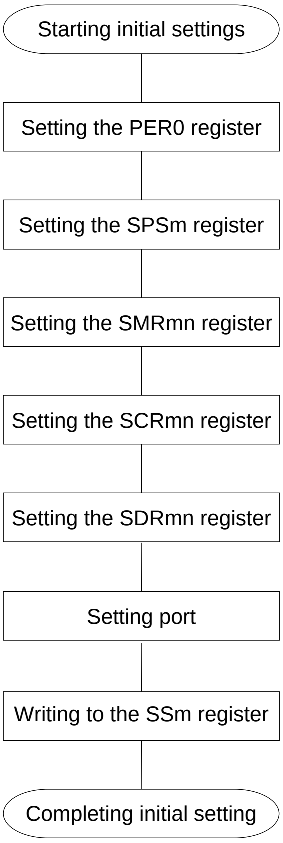
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-60. Initial Setting Procedure for Slave Reception


Starting initial settings
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

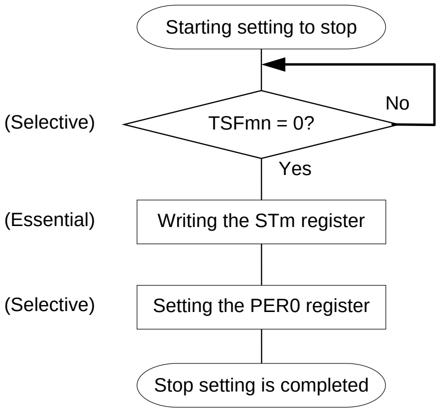
Set a communication format.

Set baud rate setting (bits 15 to 9) to 0000000B.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 20-61. Procedure for Stopping Slave Reception



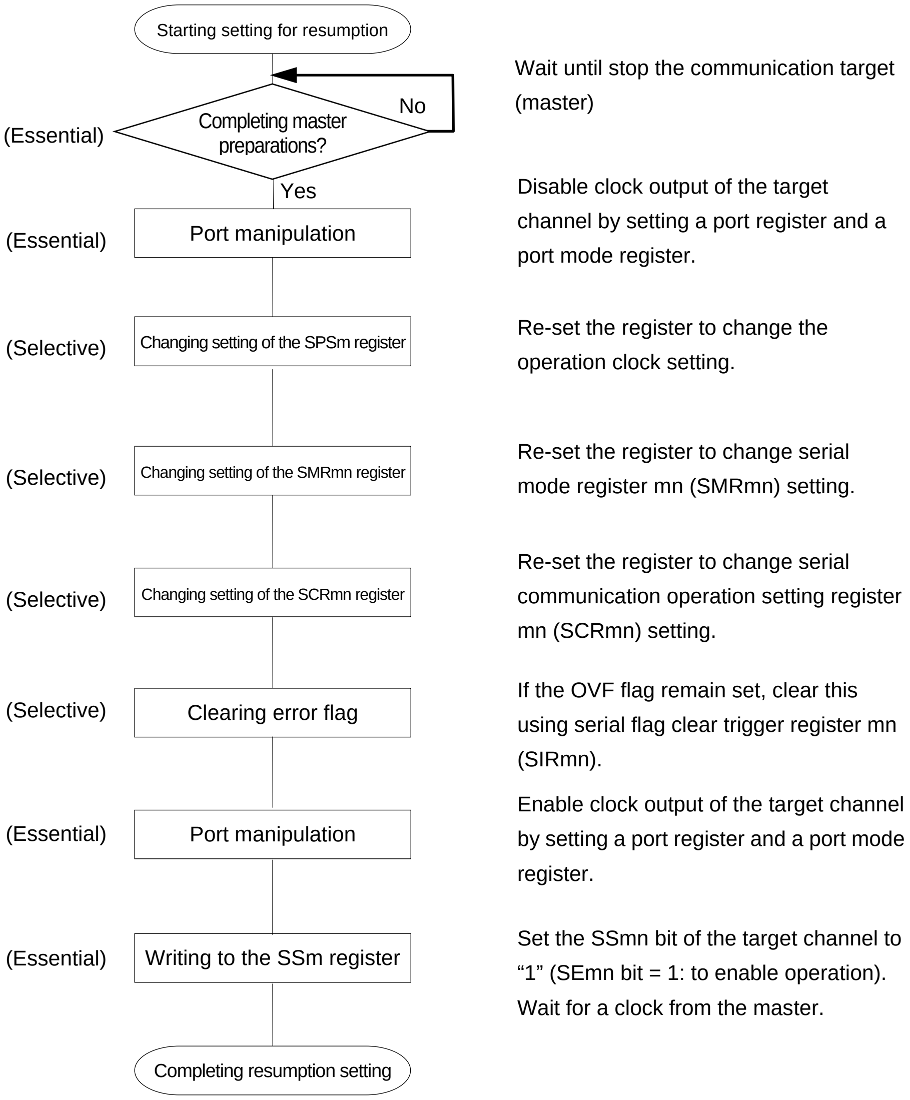
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

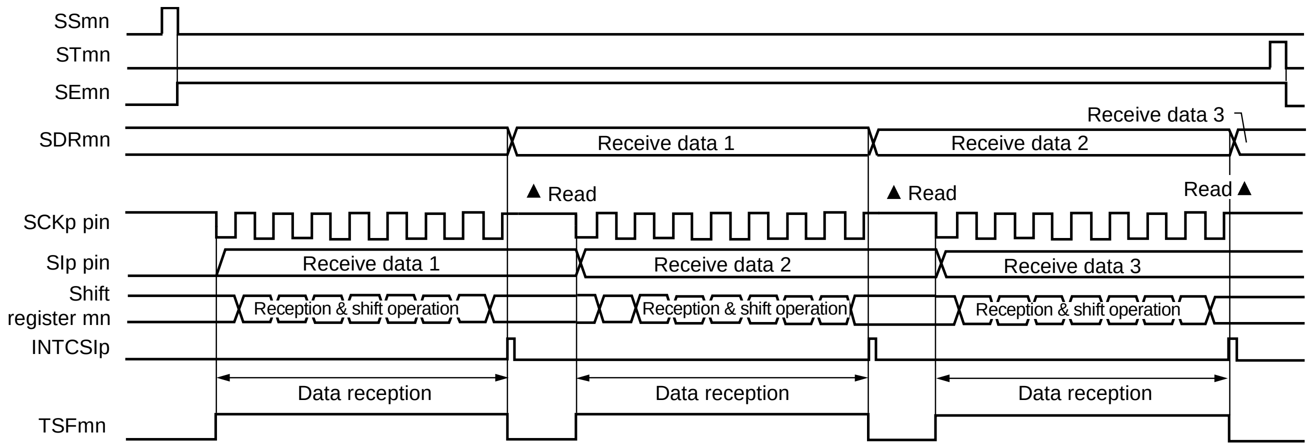
Figure 20-62. Procedure for Resuming Slave Reception


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
Wait until stop the communication target (master)
(Essential)
Port manipulation
Disable clock output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).
(Essential)
Port manipulation
Enable clock output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to "1" (SEmn bit = 1: to enable operation). Wait for a clock from the master.
Completing resumption setting

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

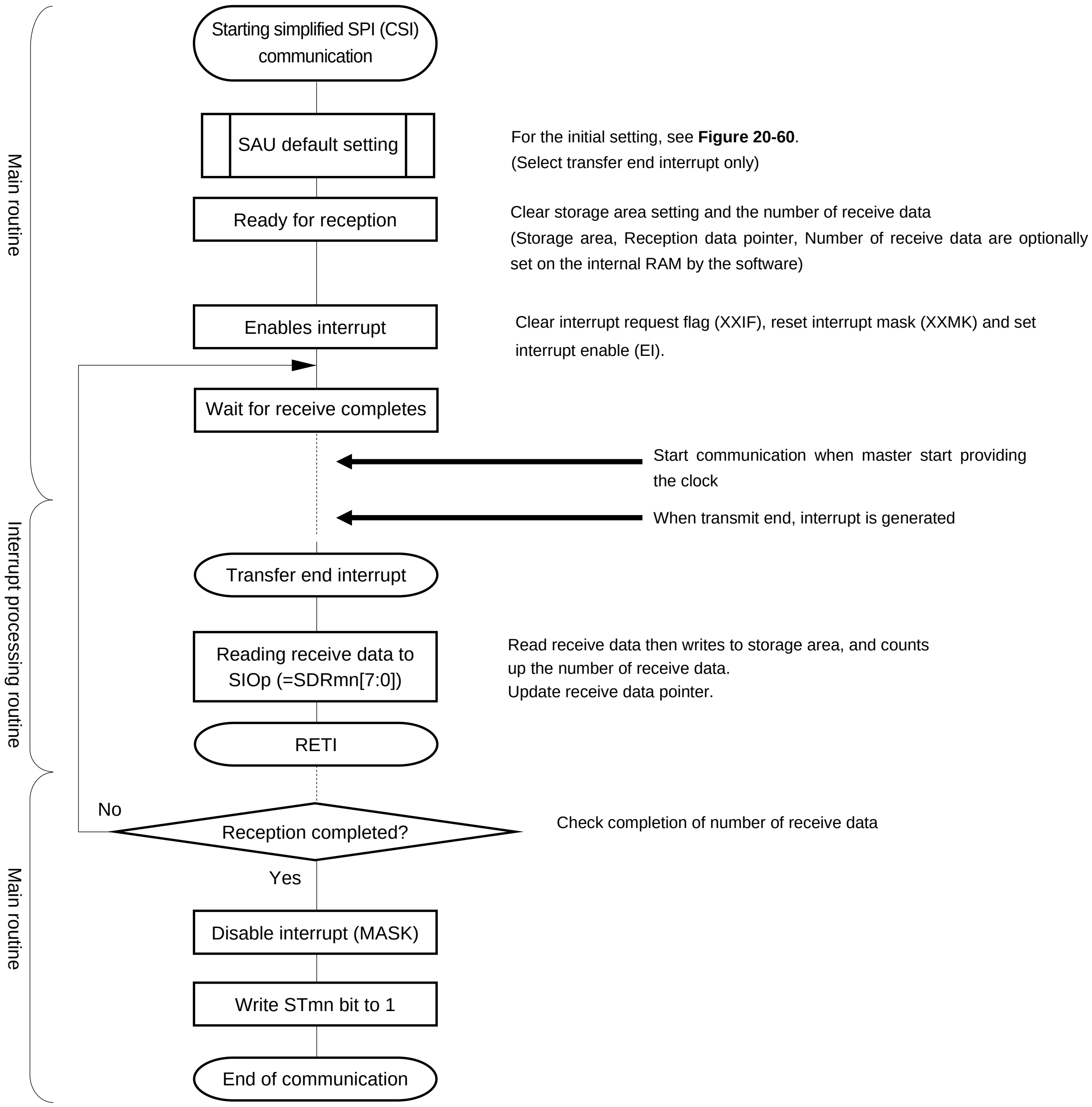
(3) Processing flow (in single-reception mode)

Figure 20-63. Timing Chart of Slave Reception (in Single-reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-64. Flowchart of Slave Reception (in Single-reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, see **Figure 20-60**.
(Select transfer end interrupt only)
Ready for reception
Clear storage area setting and the number of receive data
(Storage area, Reception data pointer, Number of receive data are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Wait for receive completes
Start communication when master start providing the clock
When transmit end, interrupt is generated
Transfer end interrupt
Reading receive data to SIOp (=SDRmn[7:0])
Read receive data then writes to storage area, and counts up the number of receive data.
Update receive data pointer.
RETI
No
Reception completed?
Check completion of number of receive data
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

20.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmit/receive data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI10	CSI30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10	SCK30, SI30, SO30
Interrupt	INTCSI00	INTCSI10	INTCSI30
	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 from the start of the operation of the serial clock. ● DAPmn = 1: Data I/O starts half a clock 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		

Notes 1. Because the external serial clock input to the SCK00, SCK10, and SCK30 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-65. Example of Contents of Registers for Slave Transmission/Reception of simplified SPI (CSI00, CSI10, CSI30) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_MCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **20.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

$\underbrace{\text{bits 7–0}}_{\text{SIOp}}$

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

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

Figure 20-65. Example of Contents of Registers for Slave Transmission/Reception of simplified SPI (CSI00, CSI10, CSI30) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

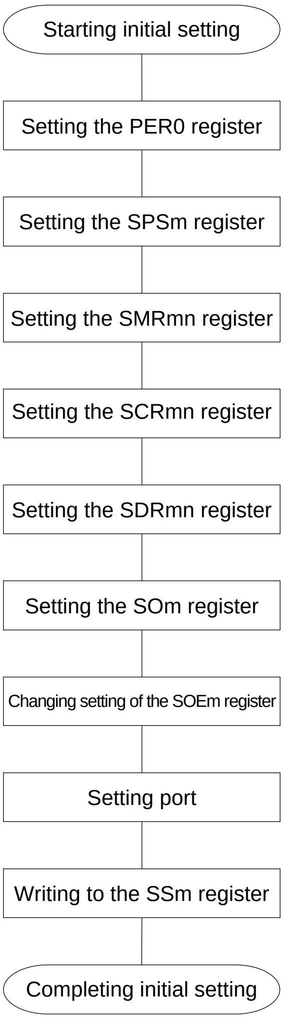
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-66. Initial Setting Procedure for Slave Transmission/Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to “1” and enable data output of the target channel.

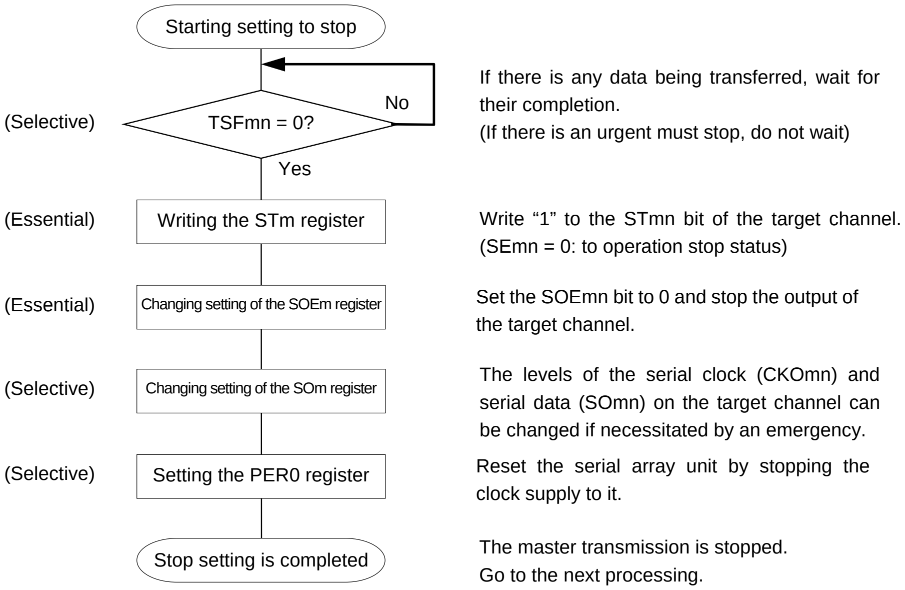
Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 20-67. Procedure for Stopping Slave Transmission/Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

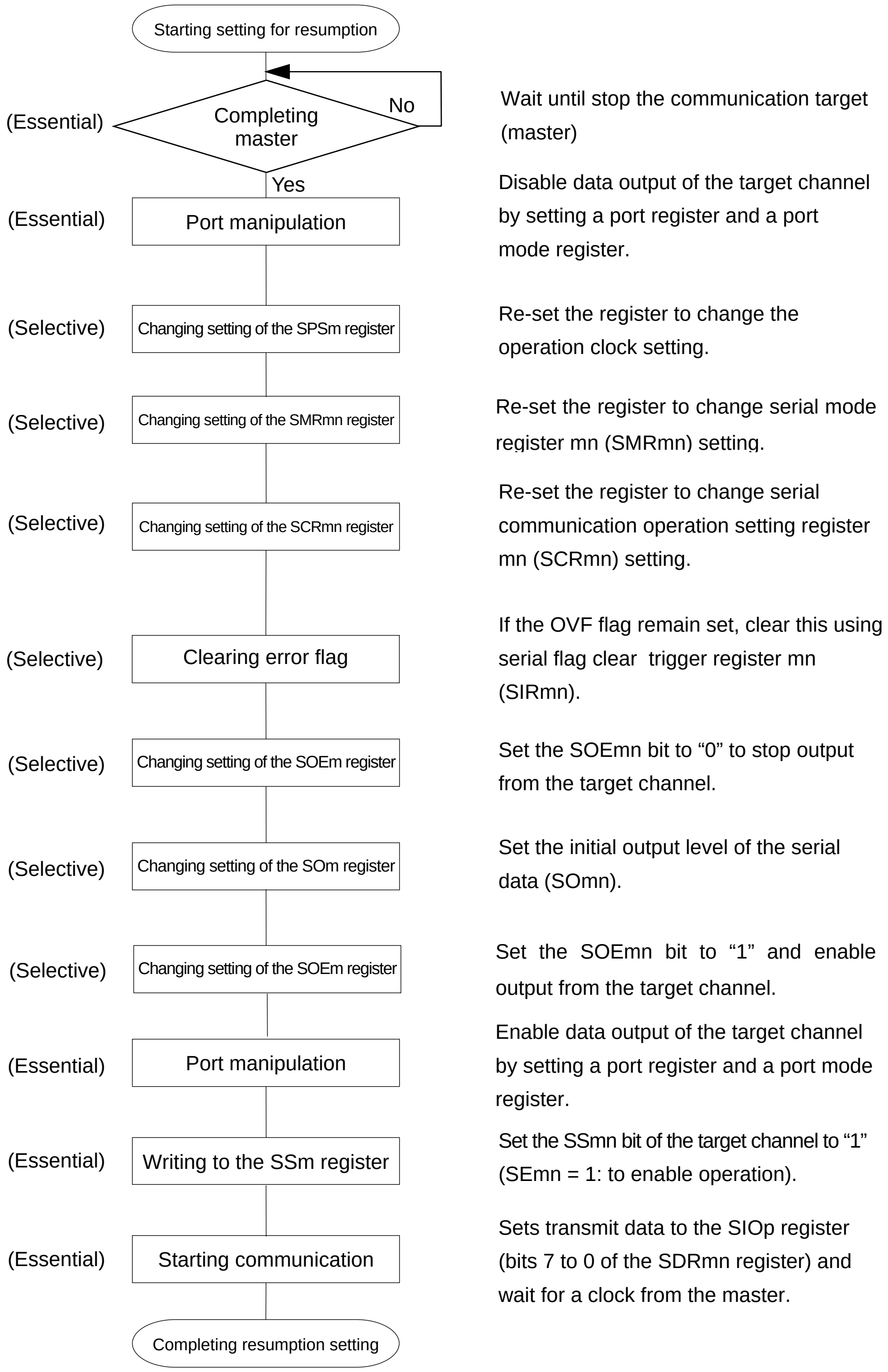
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-68. Procedure for Resuming Slave Transmission/Reception

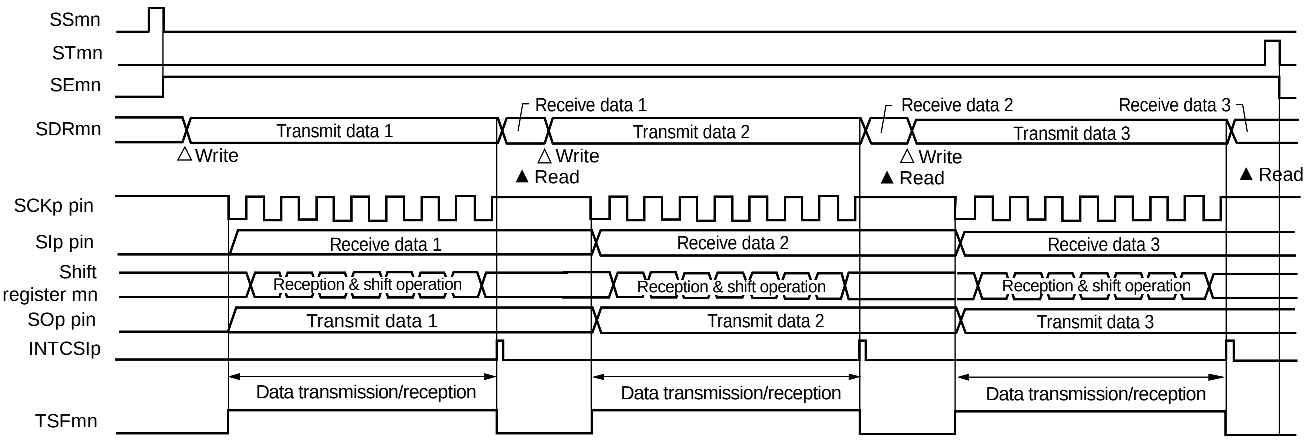

Starting setting for resumption
(Essential)
Completing master
No
Yes
Wait until stop the communication target (master)
(Essential)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).
(Selective)
Changing setting of the SOEm register
Set the SOEmn bit to "0" to stop output from the target channel.
(Selective)
Changing setting of the SOm register
Set the initial output level of the serial data (SOmn).
(Selective)
Changing setting of the SOEm register
Set the SOEmn bit to "1" and enable output from the target channel.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to "1" (SEmn = 1: to enable operation).
(Essential)
Starting communication
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.
Completing resumption setting

Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

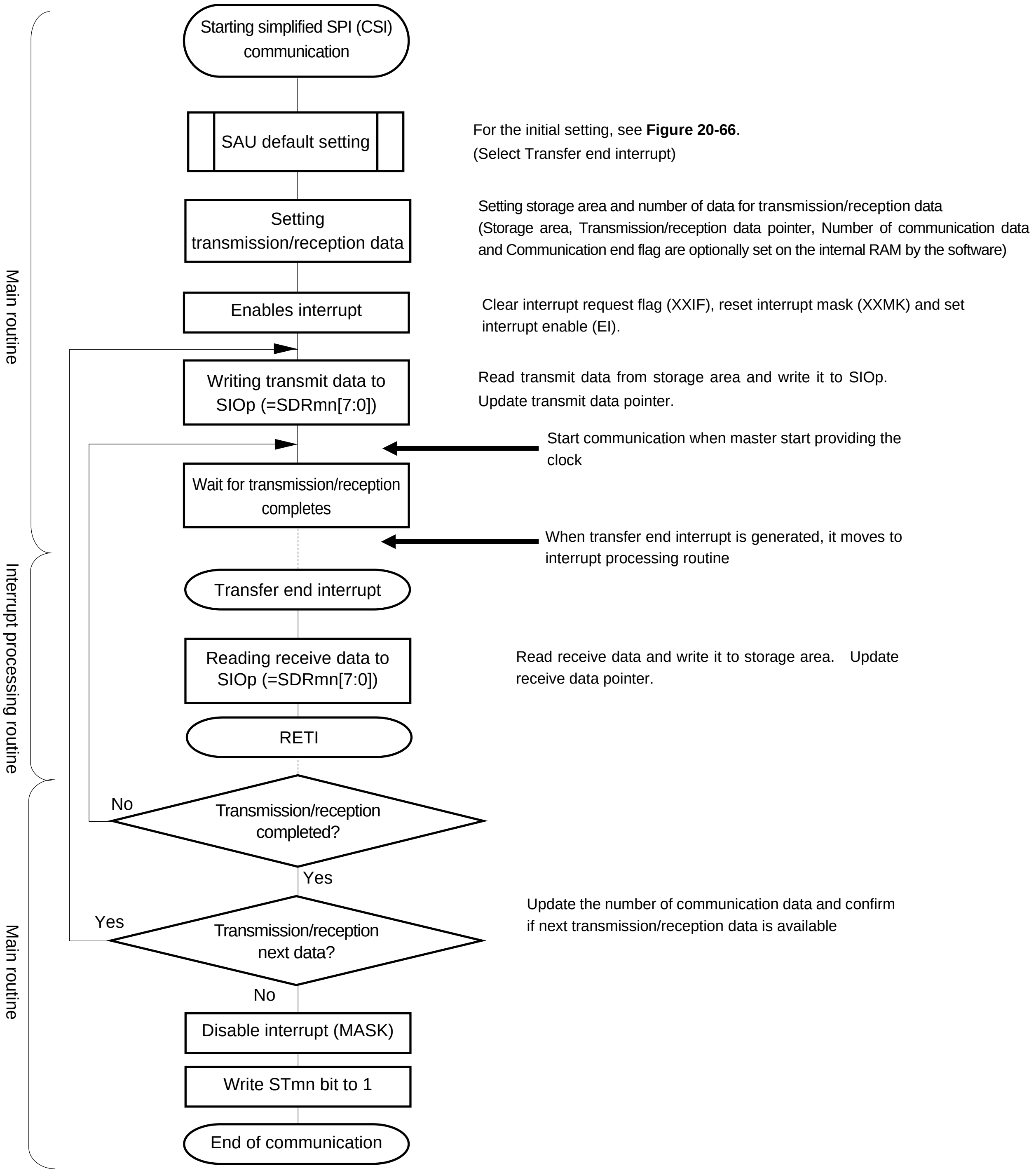
(3) Processing flow (in single-transmission/reception mode)

Figure 20-69. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

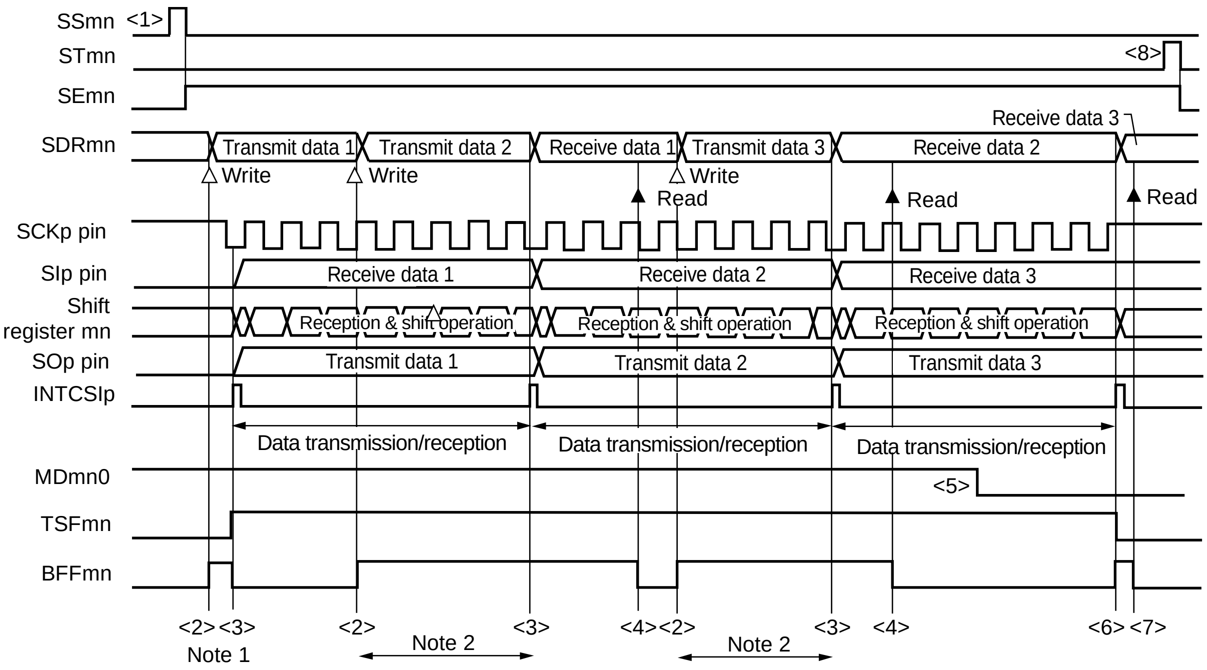
Figure 20-70. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, see **Figure 20-66**.
(Select Transfer end interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data
(Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from storage area and write it to SIOp.
Update transmit data pointer.
Start communication when master start providing the clock
Wait for transmission/reception completes
When transfer end interrupt is generated, it moves to interrupt processing routine
Interrupt processing routine
Transfer end interrupt
Reading receive data to SIOp (=SDRmn[7:0])
Read receive data and write it to storage area. Update receive data pointer.
RETI
Main routine
No
Transmission/reception completed?
Yes
Yes
Transmission/reception next data?
Update the number of communication data and confirm if next transmission/reception data is available
No
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

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 20-71. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 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.

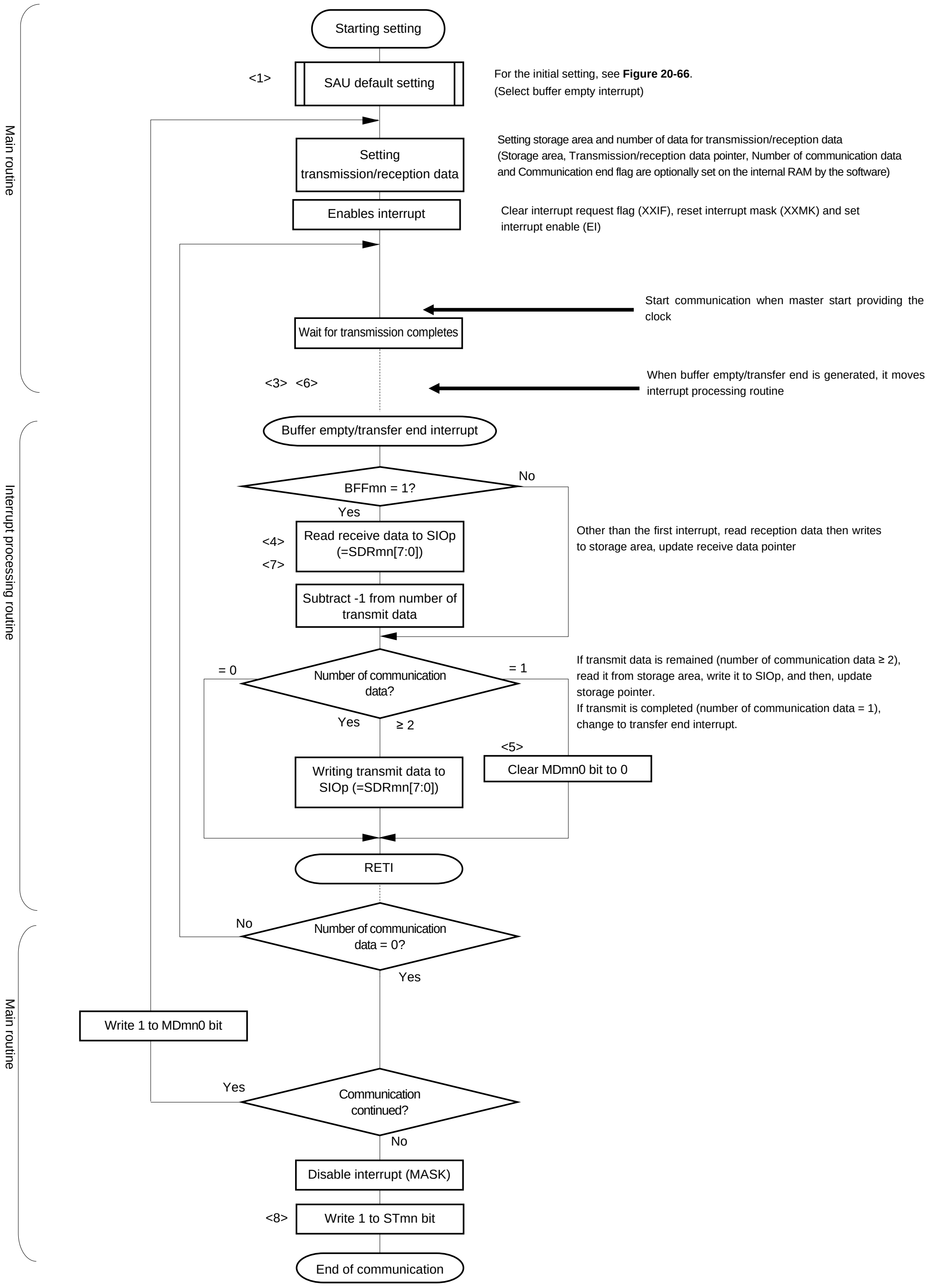
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) 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.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 20-72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 30), mn = 00, 02, 12

Figure 20-72. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)


Main routine
Starting setting
<1>
SAU default setting
For the initial setting, see **Figure 20-66**.
(Select buffer empty interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data
(Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
Start communication when master start providing the clock
Wait for transmission completes
<3> <6>
When buffer empty/transfer end is generated, it moves interrupt processing routine
Interrupt processing routine
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
<7>
Read receive data to SIOp (=SDRmn[7:0])
Other than the first interrupt, read reception data then writes to storage area, update receive data pointer
Subtract -1 from number of transmit data
= 0
Number of communication data?
= 1
If transmit data is remained (number of communication data ≥ 2), read it from storage area, write it to SIOp, and then, update storage pointer.
If transmit is completed (number of communication data = 1), change to transfer end interrupt.
Yes
≥ 2
<5>
Writing transmit data to SIOp (=SDRmn[7:0])
Clear MDmn0 bit to 0
RETI
No
Number of communication data = 0?
Yes
Main routine
Write 1 to MDmn0 bit
Yes
Communication continued?
No
Disable interrupt (MASK)
<8>
Write 1 to STmn bit
End of communication

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 20-71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

20.5.7 SNOOZE mode function

SNOOZE mode makes simplified SPI (CSI) operate reception by SCKp pin input detection while the STOP mode. Normally simplified SPI (CSI) stops communication in the STOP mode. But, using the SNOOZE mode makes reception simplified SPI (CSI) operate unless the CPU operation by detecting SCKp pin input.

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 20-74 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 20-76 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.
- After a transition to the STOP mode, it transits to SNOOZE mode upon detection of an effective edge of the SCKp pin.

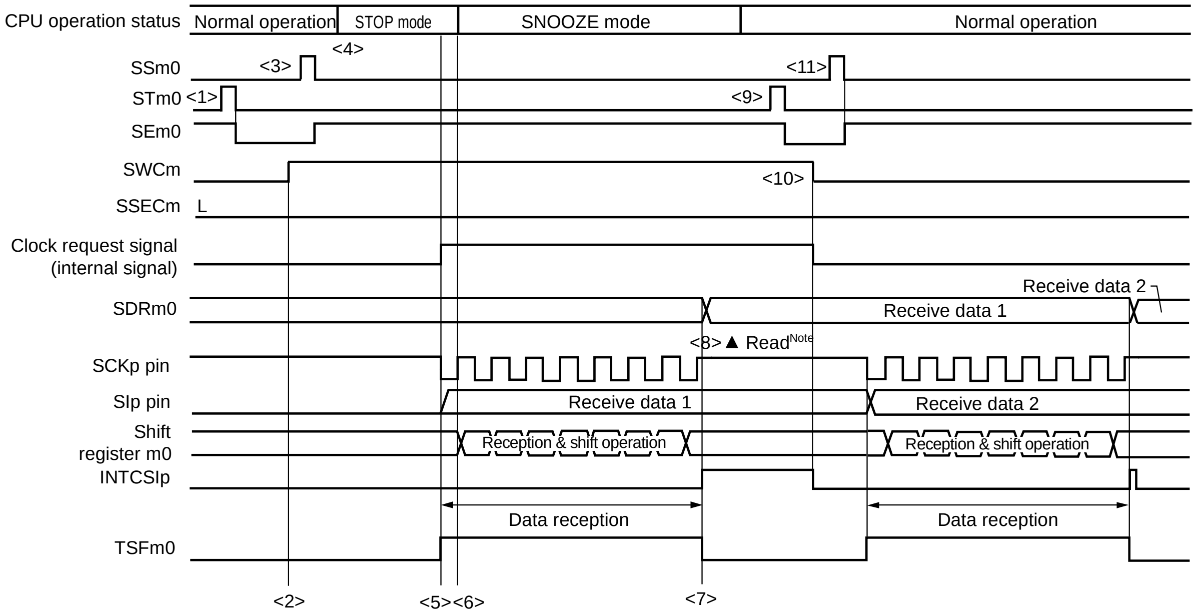
  The CSIp starts reception operations with the serial clock input of SCKp pin.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or the medium-speed on-chip oscillator clock (f_{IM}) is selected for f_{CLK}.

2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 20-73. Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)



Note Only read received data while SWCm = 1 and before the next effective edge of the SCKp pin input is detected.

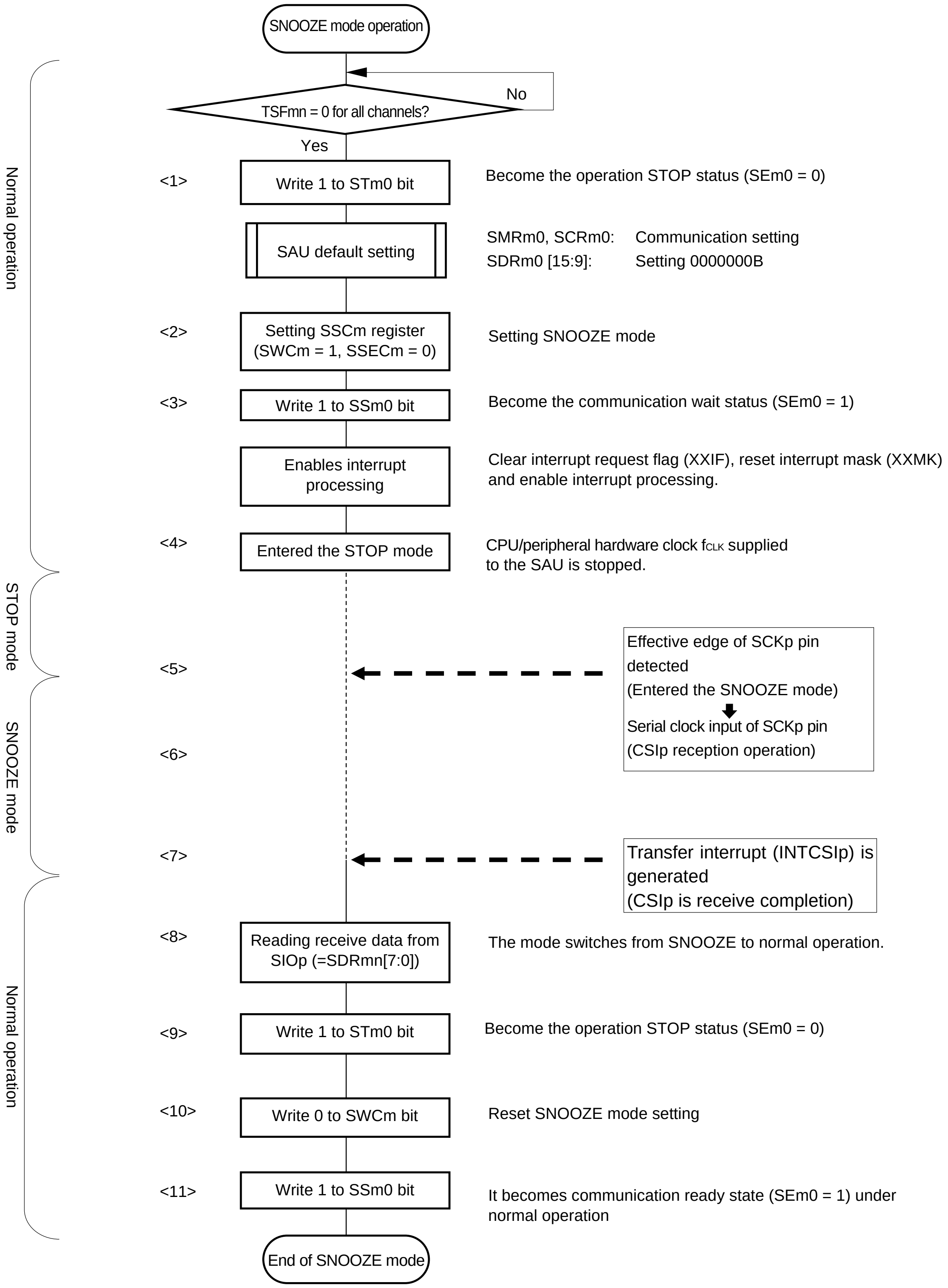
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 20-74 Flowchart of SNOOZE Mode Operation (Once Startup)**.

2. m = 0; p = 00

Figure 20-74. Flowchart of SNOOZE Mode Operation (Once Startup)

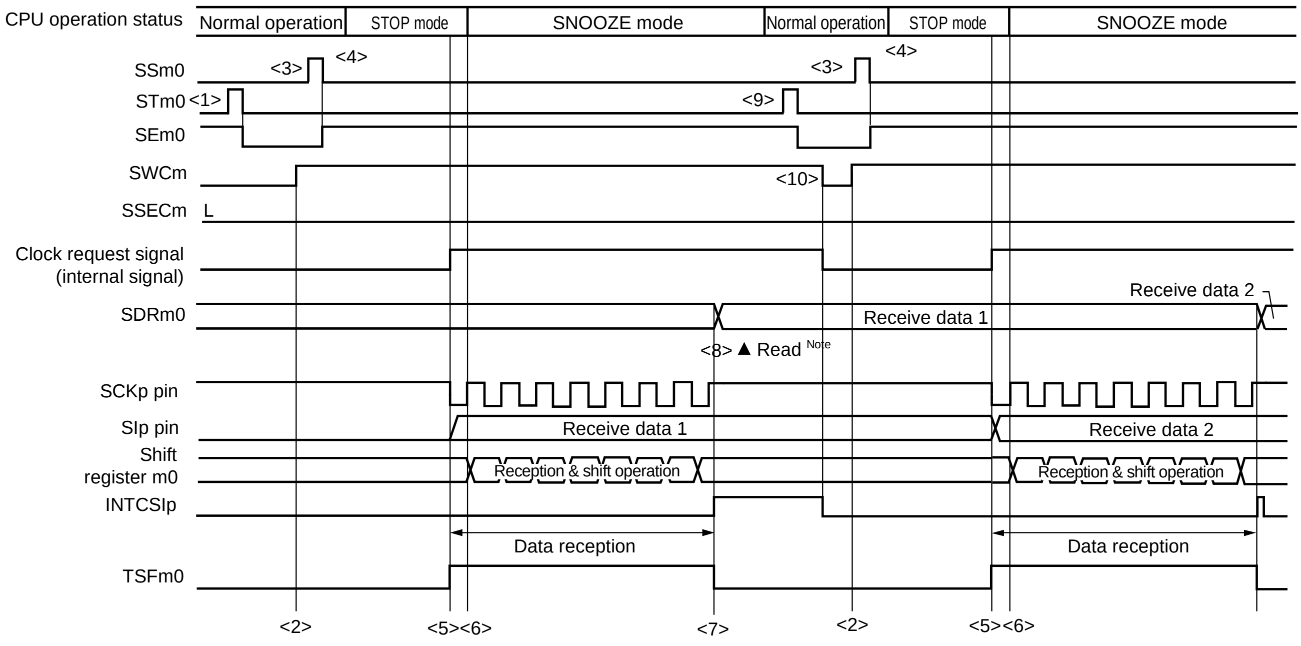

SNOOZE mode operation
TSFmn = 0 for all channels?
No
Yes
Normal operation
<1>
Write 1 to STm0 bit
Become the operation STOP status (SEm0 = 0)
SAU default setting
SMRm0, SCRm0: Communication setting
SDRm0 [15:9]: Setting 0000000B
<2>
Setting SSCm register (SWCm = 1, SSECm = 0)
Setting SNOOZE mode
<3>
Write 1 to SSm0 bit
Become the communication wait status (SEm0 = 1)
Enables interrupt processing
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and enable interrupt processing.
<4>
Entered the STOP mode
CPU/peripheral hardware clock fCLK supplied to the SAU is stopped.
STOP mode
<5>
Effective edge of SCKp pin detected
(Entered the SNOOZE mode)
SNOOZE mode
<6>
Serial clock input of SCKp pin
(CSIp reception operation)
<7>
Transfer interrupt (INTCSIp) is generated
(CSIp is receive completion)
Normal operation
<8>
Reading receive data from SIOp (=SDRmn[7:0])
The mode switches from SNOOZE to normal operation.
<9>
Write 1 to STm0 bit
Become the operation STOP status (SEm0 = 0)
<10>
Write 0 to SWCm bit
Reset SNOOZE mode setting
<11>
Write 1 to SSm0 bit
It becomes communication ready state (SEm0 = 1) under normal operation
End of SNOOZE mode

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 20-73 Timing Chart of SNOOZE Mode Operation (Once Startup)**.

2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 20-75. Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPm0 = 0, CKPm0 = 0)



Note Only read received data while SWCm = 1 and before the next effective edge of the SCKp pin input is detected.

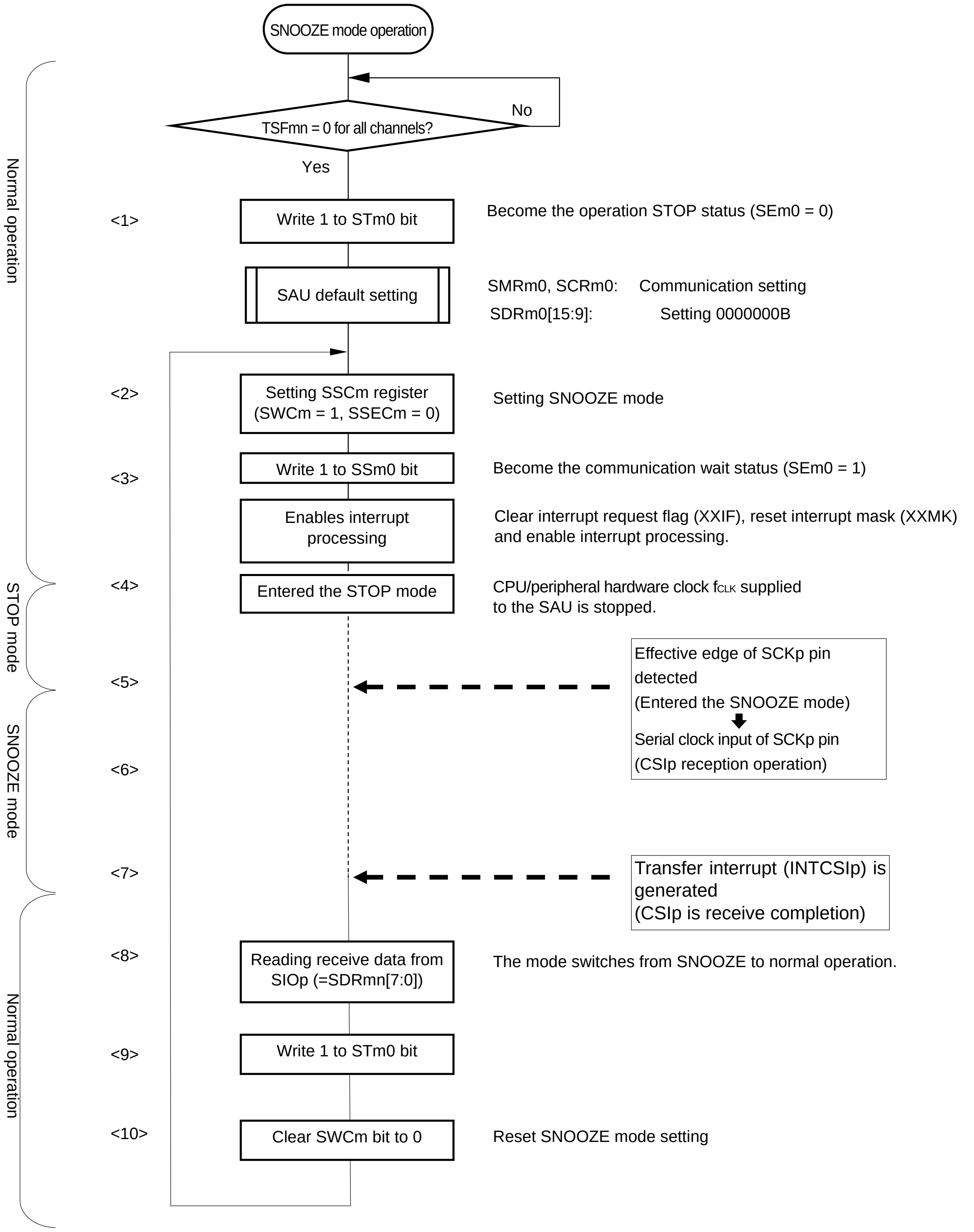
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 20-76 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.

2. m = 0; p = 00

Figure 20-76. Flowchart of SNOOZE Mode Operation (Continuous Startup)


SNOOZE mode operation
TSFmn = 0 for all channels?
No
Yes
Normal operation
<1>
Write 1 to STm0 bit
Become the operation STOP status (SEm0 = 0)
SAU default setting
SMRm0, SCRm0: Communication setting
SDRm0[15:9]: Setting 0000000B
<2>
Setting SSCm register (SWCm = 1, SSECm = 0)
Setting SNOOZE mode
<3>
Write 1 to SSm0 bit
Become the communication wait status (SEm0 = 1)
Enables interrupt processing
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and enable interrupt processing.
STOP mode
<4>
Entered the STOP mode
CPU/peripheral hardware clock fCLK supplied to the SAU is stopped.
SNOOZE mode
<5>
Effective edge of SCKp pin detected
(Entered the SNOOZE mode)
Serial clock input of SCKp pin
(CSIp reception operation)
<6>
<7>
Transfer interrupt (INTCSIp) is generated
(CSIp is receive completion)
Normal operation
<8>
Reading receive data from SIOp (=SDRmn[7:0])
The mode switches from SNOOZE to normal operation.
<9>
Write 1 to STm0 bit
<10>
Clear SWCm bit to 0
Reset SNOOZE mode setting

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 20-75 Timing Chart of SNOOZE Mode Operation (Continuous Startup)**.

2. m = 0; p = 00

20.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI10, CSI30) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/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 (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 20-2. Selection of Operation Clock For Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (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).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

20.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI10, CSI30) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI10, CSI30) communication is described in **Figure 20-77**.

Figure 20-77. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	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).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	Error flag is cleared.	Error can be cleared only during reading, 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, 2), mn = 00, 02, 12

20.6 Operation of UART (UART0 to UART4) Communication

This is a start-stop synchronization 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 UART0, timer array unit 0 (channel 7), and an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception supports the SNOOZE mode. When RxD pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only UART0 can be specified for the reception baud rate adjustment function.

The LIN-bus is accepted in UART0 (channels 0 and 1 of unit 0).

[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 0 (channel 7)

Note Only UART0 can be specified for the 9-bit data length.

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.
UART4 uses channels 0 and 1 of SAU2.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2 (supporting IrDA)	–
	1	–		–
	2	CSI30[Note]	UART3[Note]	IIC30[Note]
	3	–		–
2	0	–	UART4[Note]	–
	1	–		–

Note 100-pin product only

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as 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.

UART performs the following four types of communication operations.

- UART transmission (See **20.6.1**.)
- UART reception (See **20.6.2**.)
- LIN transmission (UART0 only) (See **20.7.1**.)
- LIN reception (UART0 only) (See **20.7.2**.)

20.6.1 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	UART4
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1	Channel 0 of SAU2
Pins used	TxD0	TxD1	TxD2	TxD3	TxD4
Interrupt	INTST0	INTST1	INTST2	INTST3	INTST4
	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	Max. $f_{MCK}/6$ [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps][Note 2]				
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 bits				
Data direction	MSB or LSB first				

Notes 1. Only UART0 can be specified for the 9-bit data length.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

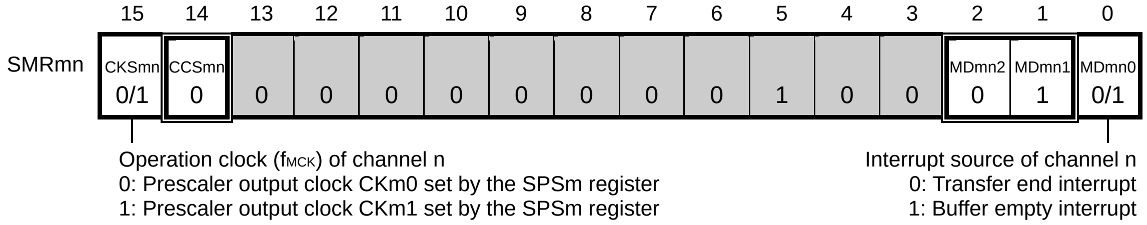
f_{CLK}: System clock frequency

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20

(1) Register setting

Figure 20-78. Example of Contents of Registers for UART Transmission of UART (UART0 to UART4) (1/2)

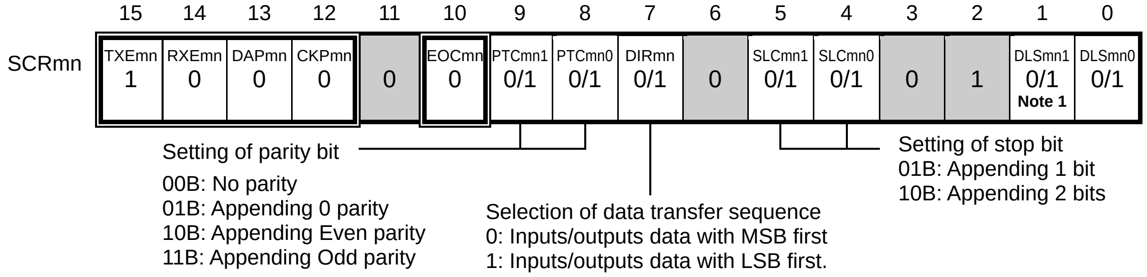
(a) Serial mode register mn (SMRmn)



Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

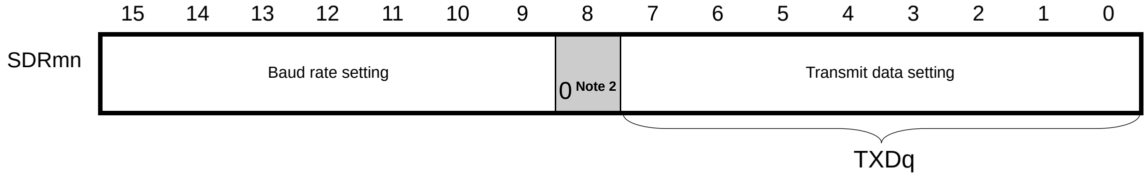


Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

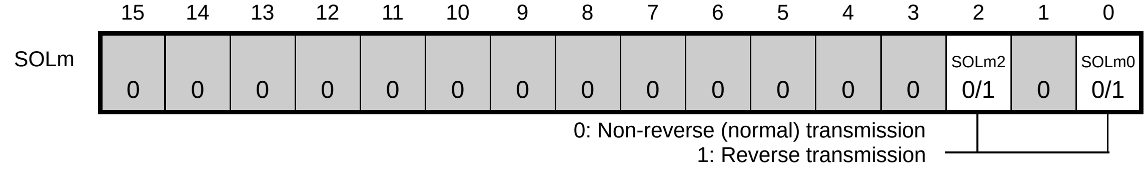
Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



0: Non-reverse (normal) transmission
1: Reverse transmission

Notes 1. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

2. When UART0 performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area. Only UART0 can be specified for the 9-bit data length.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), q: UART number (q = 0 to 4), mn = 00, 02, 10, 12, 20

2. ☐: Setting is fixed in the UART transmission mode, ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 20-78. Example of Contents of Registers for UART Transmission of UART (UART0 to UART4) (2/2)

(e) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1 Note	1	SOm0 0/1 Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

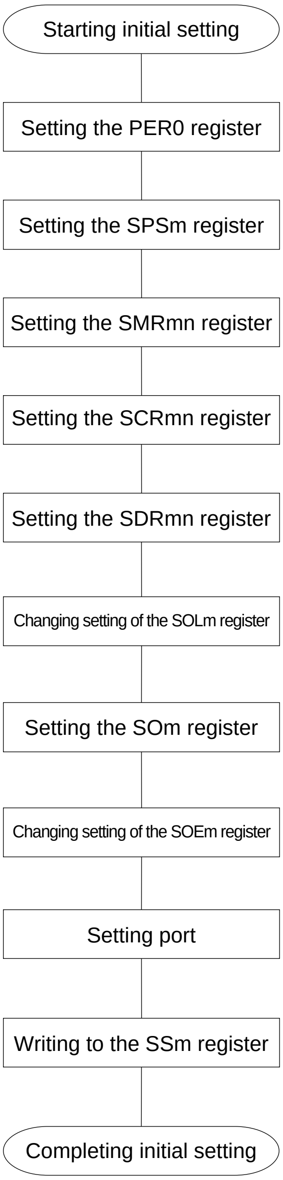
Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 0, 3), q: UART number (q = 0 to 4)
mn = 00, 02, 10, 12, 20

2. ▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-79. Initial Setting Procedure for UART Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Changing setting of the SOLm register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set an output data level.

Set the initial output level of the serial data (SOmn).

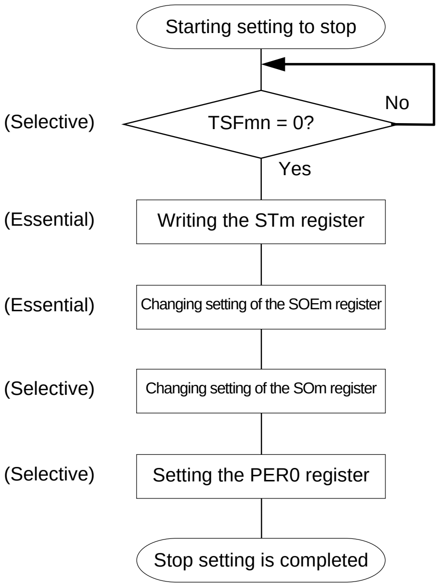
Set the SOEmn bit to “1” and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” and set the SEmn bit to 1 (to enable operation).

Initial setting is completed.
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits) and start communication.

Figure 20-80. Procedure for Stopping UART Transmission


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

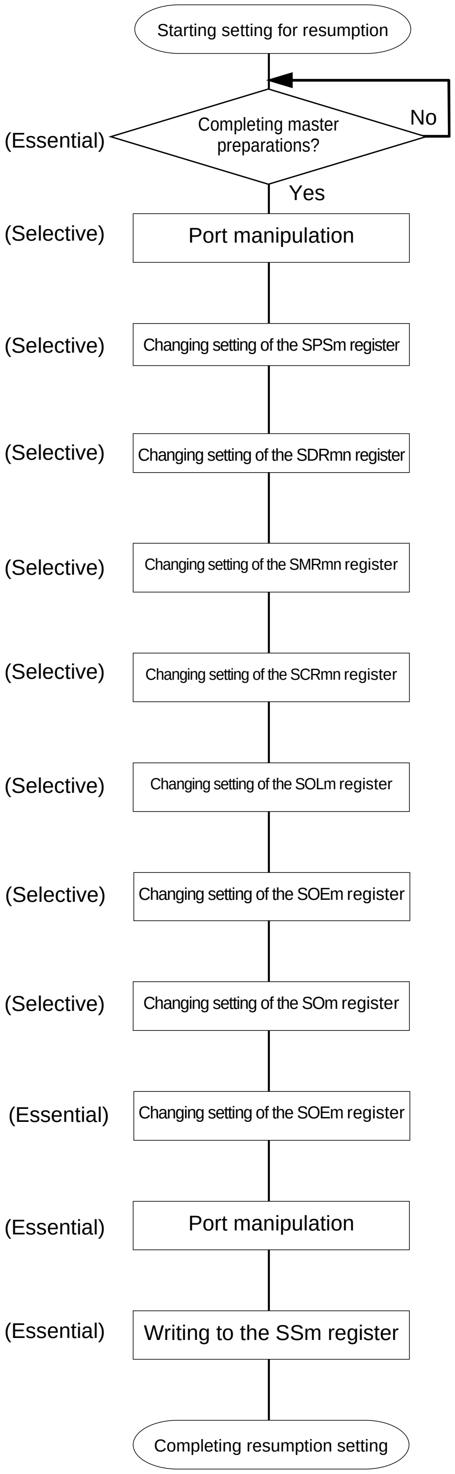
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-81. Procedure for Resuming UART Transmission


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
(Selective)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOLm register
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Essential)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to “0” and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to “1” and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

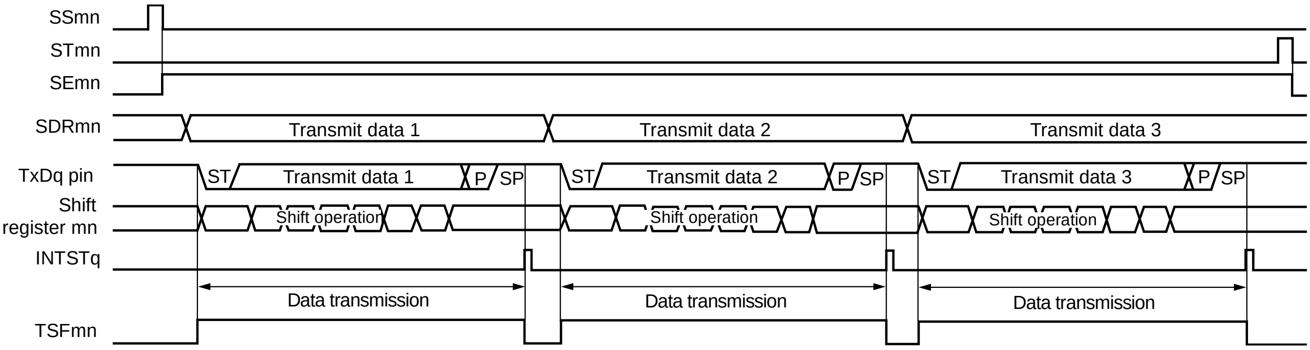
Set the SSmn bit of the target channel to “1” and set the SEmn bit to “1” (to enable operation).

Setting is completed.
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

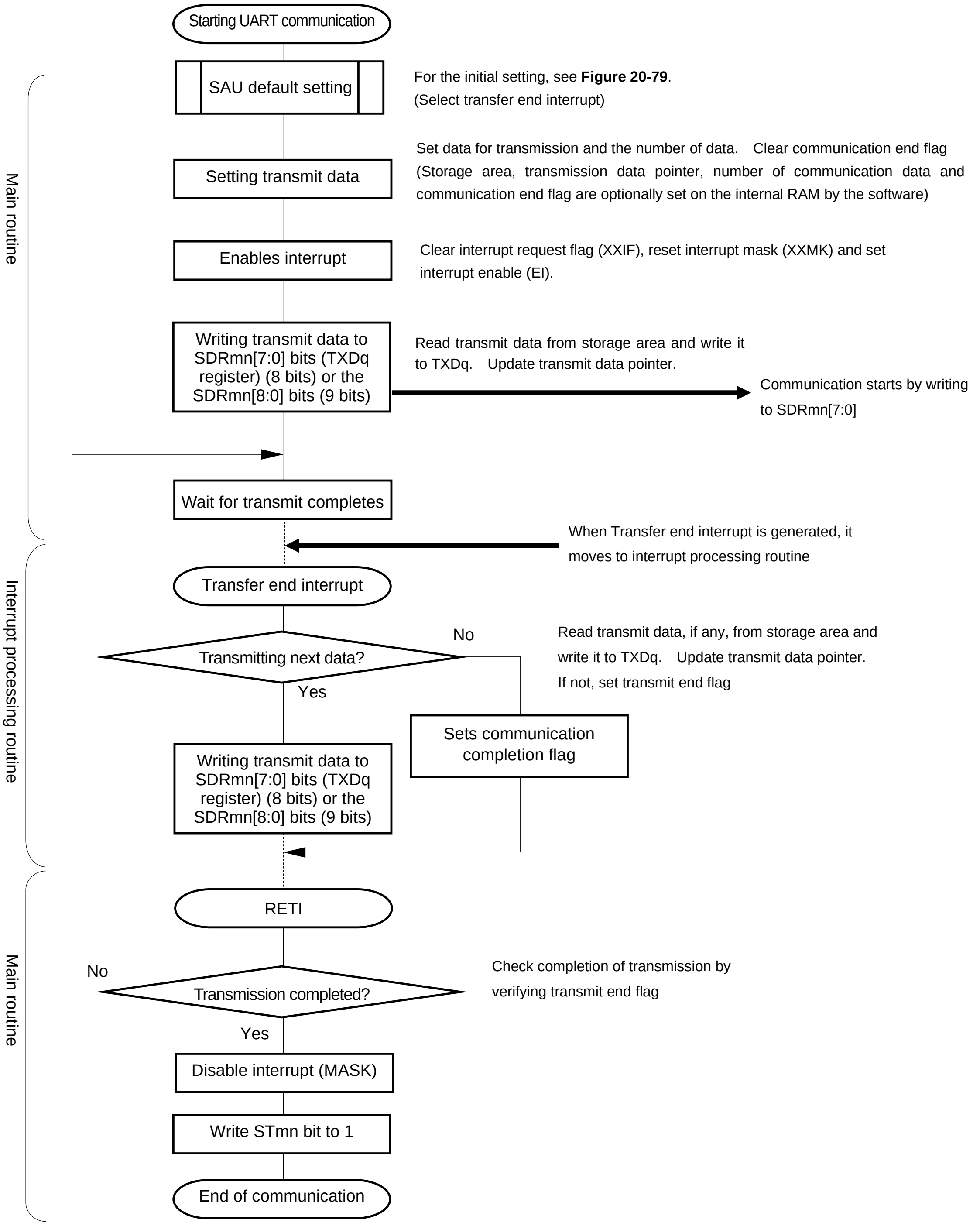
(3) Processing flow (in single-transmission mode)

Figure 20-82. Timing Chart of UART Transmission (in Single-Transmission Mode)



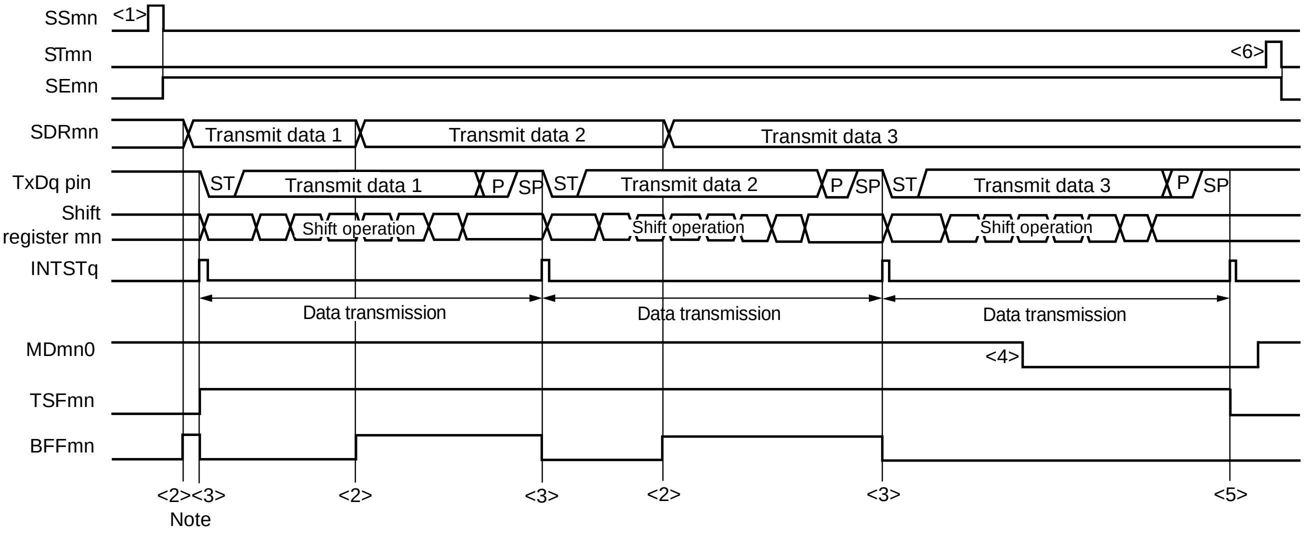
Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), q: UART number (q = 0 to 4)
mn = 00, 02, 10, 12, 20

Figure 20-83. Flowchart of UART Transmission (in Single-Transmission Mode)


Starting UART communication
SAU default setting
For the initial setting, see **Figure 20-79**.
(Select transfer end interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag (Storage area, transmission data pointer, number of communication data and communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
Read transmit data from storage area and write it to TXDq. Update transmit data pointer.
Communication starts by writing to SDRmn[7:0]
Wait for transmit completes
When Transfer end interrupt is generated, it moves to interrupt processing routine
Transfer end interrupt
Transmitting next data?
No
Yes
Read transmit data, if any, from storage area and write it to TXDq. Update transmit data pointer.
If not, set transmit end flag
Sets communication completion flag
Writing transmit data to SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
RETI
No
Transmission completed?
Yes
Check completion of transmission by verifying transmit end flag
Disable interrupt (MASK)
Write STmn bit to 1
End of communication
Main routine
Interrupt processing routine
Main routine

(4) Processing flow (in continuous transmission mode)

Figure 20-84. Timing Chart of UART Transmission (in Continuous Transmission Mode)

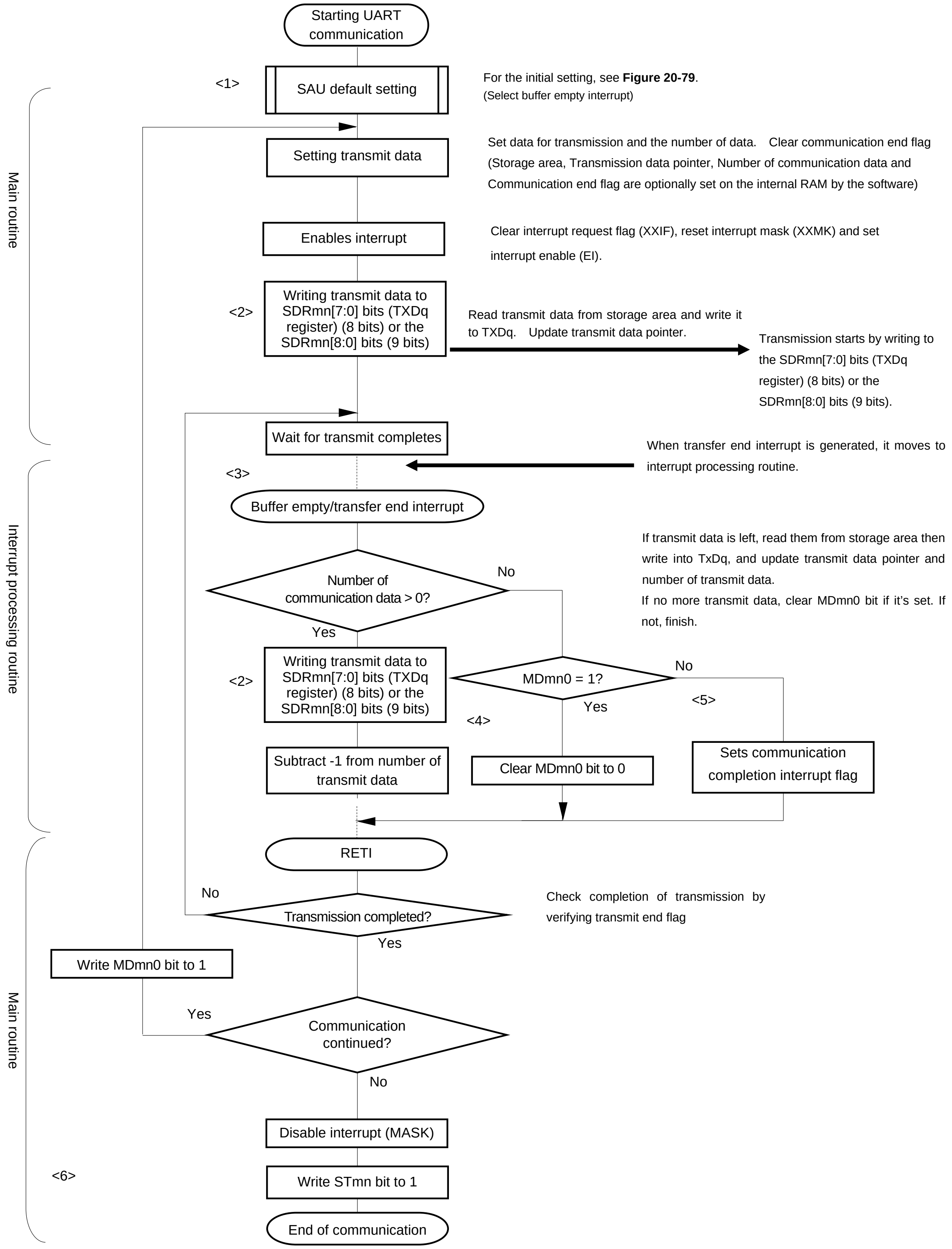


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), q: UART number (q = 0 to 4)
mn = 00, 02, 10, 12, 20

Figure 20-85. Flowchart of UART Transmission (in Continuous Transmission Mode)


Starting UART communication
<1>
SAU default setting
Main routine
Setting transmit data
Enables interrupt
<2>
Writing transmit data to SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
Wait for transmit completes
<3>
Interrupt processing routine
Buffer empty/transfer end interrupt
Number of communication data > 0?
No
Yes
<2>
Writing transmit data to SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
MDmn0 = 1?
No
<5>
Yes
<4>
Subtract -1 from number of transmit data
Clear MDmn0 bit to 0
Sets communication completion interrupt flag
RETI
No
Transmission completed?
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Main routine
Disable interrupt (MASK)
<6>
Write STmn bit to 1
End of communication

For the initial setting, see **Figure 20-79**.
(Select buffer empty interrupt)

Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to TXDq. Update transmit data pointer.

Transmission starts by writing to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits).

When transfer end interrupt is generated, it moves to interrupt processing routine.

If transmit data is left, read them from storage area then write into TxDq, and update transmit data pointer and number of transmit data.
If no more transmit data, clear MDmn0 bit if it's set. If not, finish.

Check completion of transmission by verifying transmit end flag

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 20-84 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

20.6.2 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	UART4
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1	Channel 1 of SAU2
Pins used	RxD0	RxD1	RxD2	RxD3	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3	INTSR4
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3	INTSRE4
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	Max. $f_{MCK}/6$ [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps][Note 2]				
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				

Notes 1. Only UART0 can be specified for the 9-bit data length.

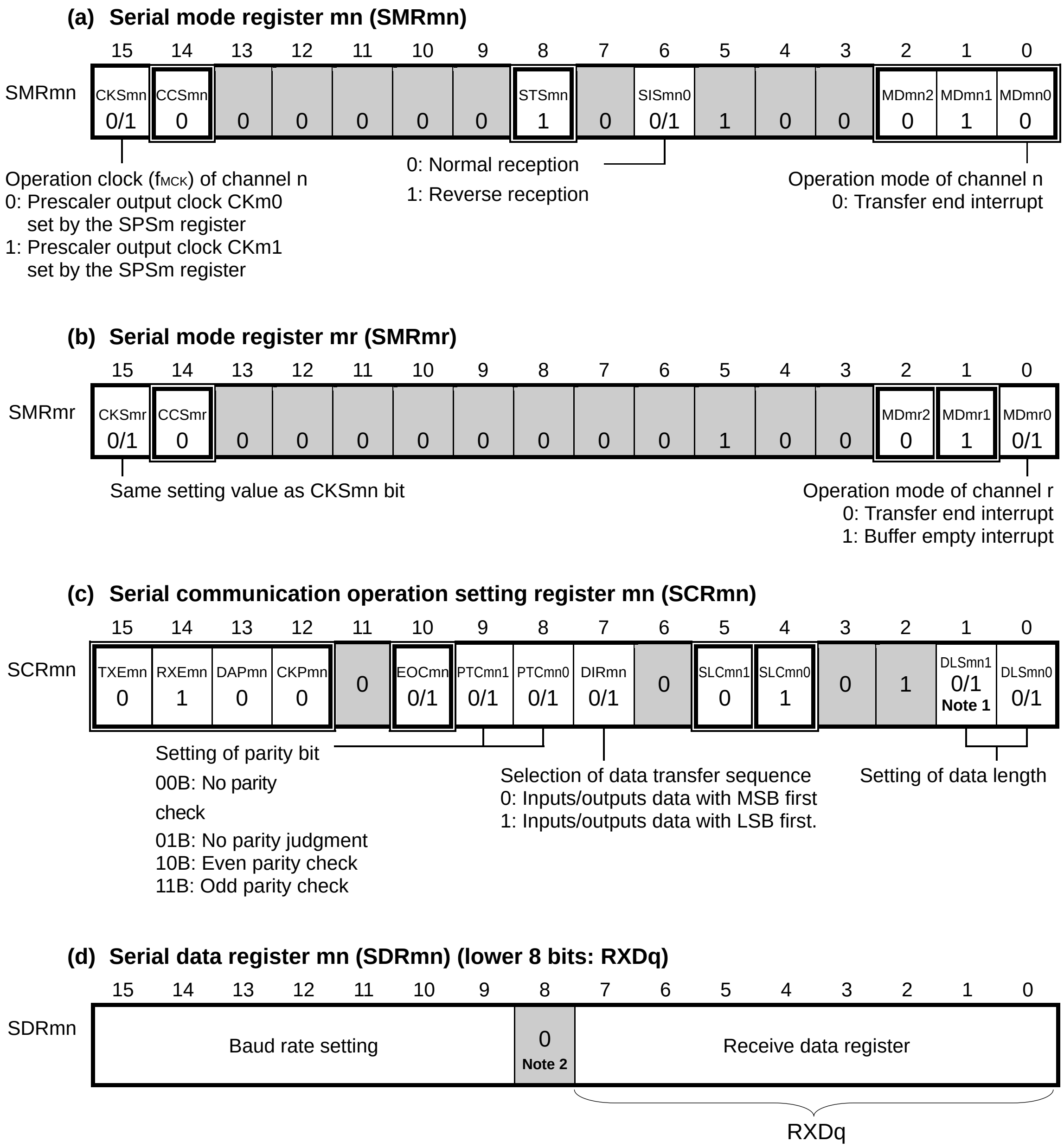
2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0 to 2), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13, 21

(1) Register setting

Figure 20-86. Example of Contents of Registers for UART Reception of UART (UART0 to UART4) (1/2)



(a) Serial mode register mn (SMRmn)

	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 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Normal reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(b) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	0	0	0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn bit

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(c) Serial communication operation setting register mn (SCRmn)

	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 0/1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 0/1 Note 1	DLSmn0 0/1

Setting of parity bit
00B: No parity check
01B: No parity judgment
10B: Even parity check
11B: Odd parity check

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length

(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15–9	8	7–0
SDRmn	Baud rate setting	0 Note 2	Receive data register

RXDq (bits 7 to 0)

Notes 1. Only provided for the SCR00 and SCR01 registers. This bit is fixed to 1 for the other registers.

2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the transmission data specification area. Only UART0 can be specified for 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.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13, 21
r: Channel number (r = n – 1), q: UART number (q = 0 to 4)

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

Figure 20-86. Example of Contents of Registers for UART Reception of UART (UART0 to UART4) (2/2)

(e) Serial output register m (SOm) … The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(f) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	0	SOEm0 ×

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel is 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 ×

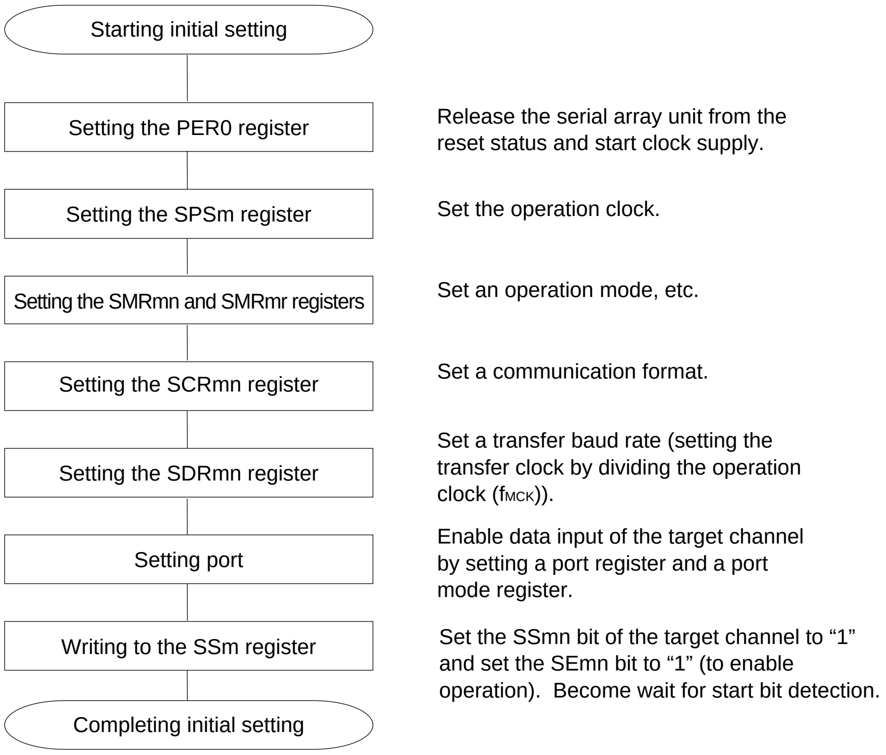
Caution For the UART reception, be sure to set the SMRmr register of channel r to UART Transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0 to 2), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13, 21
r: Channel number (r = n – 1), q: UART number (q = 0 to 4)

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 20-87. Initial Setting Procedure for UART Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn and SMRmr registers
Setting the SCRmn register
Setting the SDRmn register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

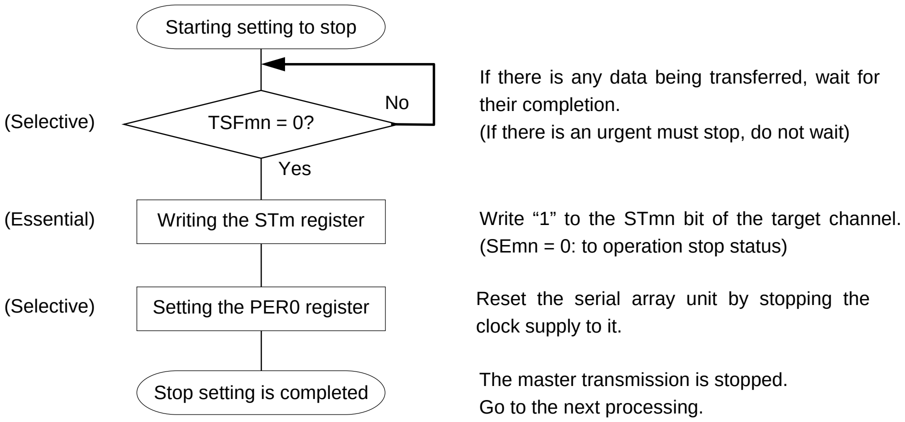
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to “1” and set the SEmn bit to “1” (to enable operation). Become wait for start bit detection.

Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 20-88. Procedure for Stopping UART Reception



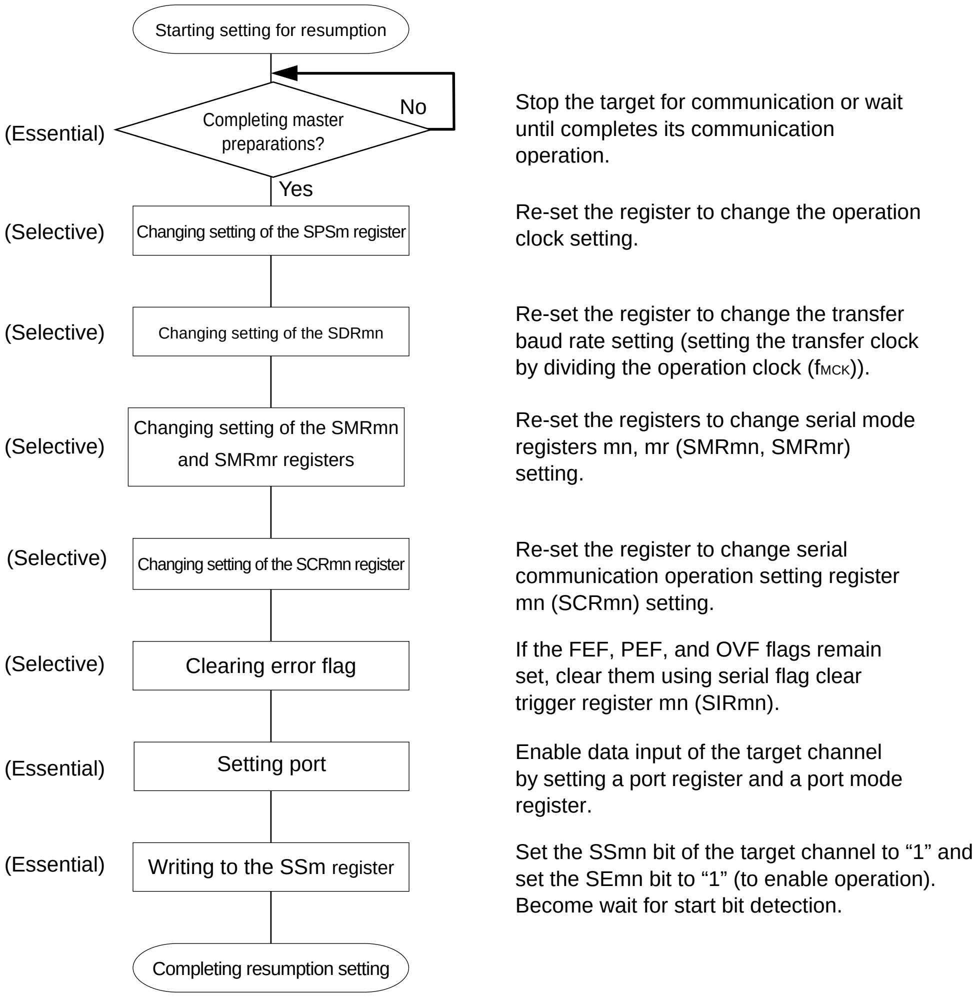
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write “1” to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 20-89. Procedure for Resuming UART Reception

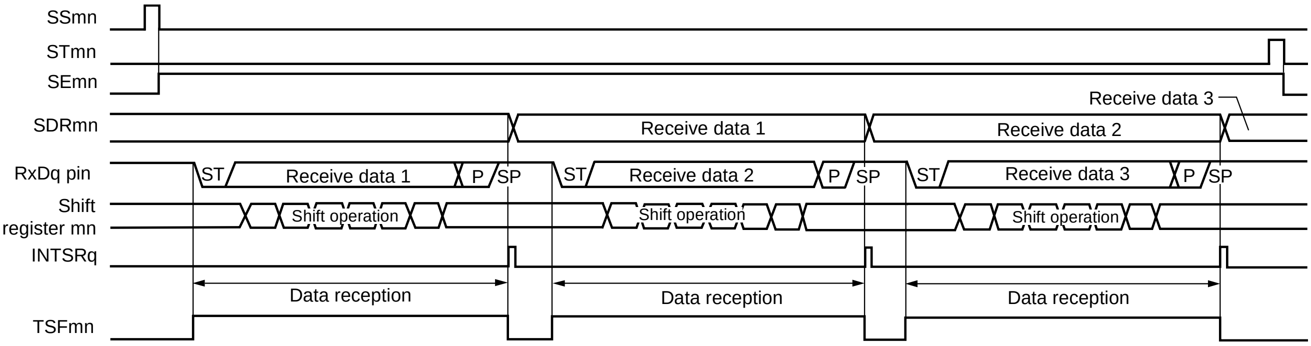

Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
Stop the target for communication or wait until completes its communication operation.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SDRmn
Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).
(Selective)
Changing setting of the SMRmn and SMRmr registers
Re-set the registers to change serial mode registers mn, mr (SMRmn, SMRmr) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).
(Essential)
Setting port
Enable data input of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to “1” and set the SEmn bit to “1” (to enable operation). Become wait for start bit detection.
Completing resumption setting

Caution After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

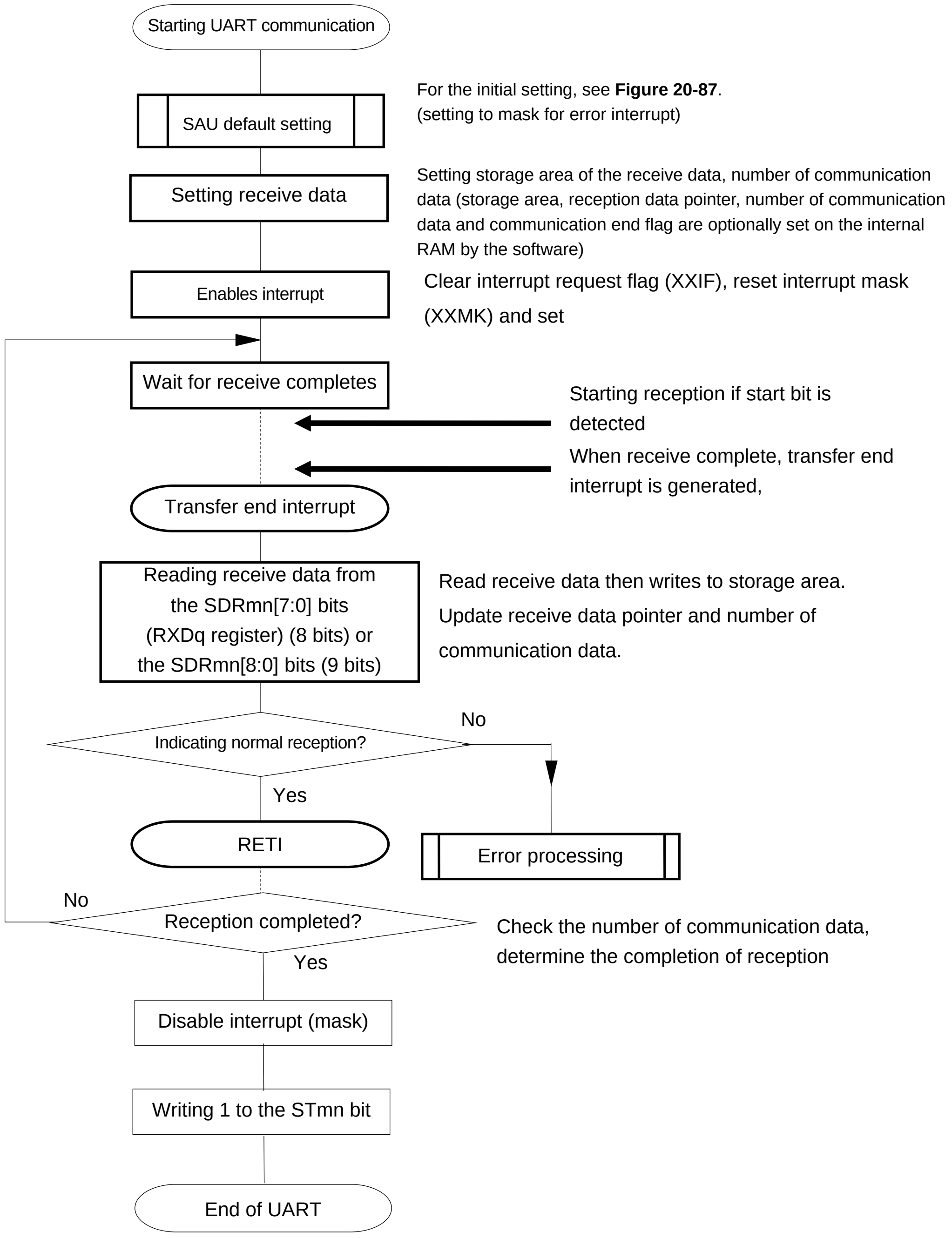
(3) Processing flow

Figure 20-90. Timing Chart of UART Reception



Remark m: Unit number (m = 0 to 2), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13, 21
r: Channel number (r = n – 1), q: UART number (q = 0 to 4)

Figure 20-91. Flowchart of UART Reception


Starting UART communication
SAU default setting
For the initial setting, see **Figure 20-87**.
(setting to mask for error interrupt)
Setting receive data
Setting storage area of the receive data, number of communication data (storage area, reception data pointer, number of communication data and communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set
Wait for receive completes
Starting reception if start bit is detected
When receive complete, transfer end interrupt is generated,
Transfer end interrupt
Reading receive data from the SDRmn[7:0] bits (RXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
Read receive data then writes to storage area. Update receive data pointer and number of communication data.
Indicating normal reception?
No
Yes
RETI
Error processing
No
Reception completed?
Yes
Check the number of communication data, determine the completion of reception
Disable interrupt (mask)
Writing 1 to the STmn bit
End of UART

20.6.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection 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 operations without CPU operation. Only UART0 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 20-94** and **Figure 20-96 Flowchart of SNOOZE Mode Operation**.)

- 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 20-3**.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- Upon detecting the start bit input of RxDq after a transition was made to the STOP mode, UARTq reception is started.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using UARTq in the SNOOZE mode is 4800 bps.

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**

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, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that the UART reception 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 reception.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01, q: UART number (q = 0)

Table 20-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 1.0% Note	$f_{CLK}/2^5$	105	2.27%	-1.53%
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	-1.57%

Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5%, the permissible range becomes smaller as shown below.

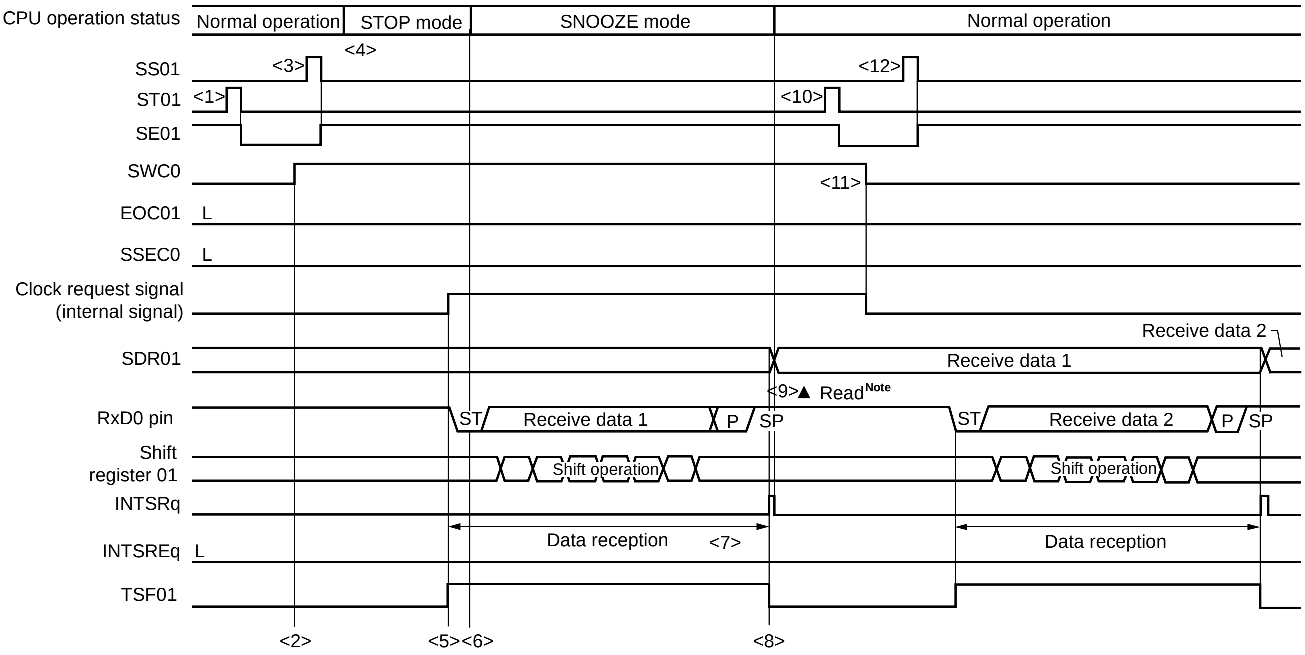
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (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 20-92. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

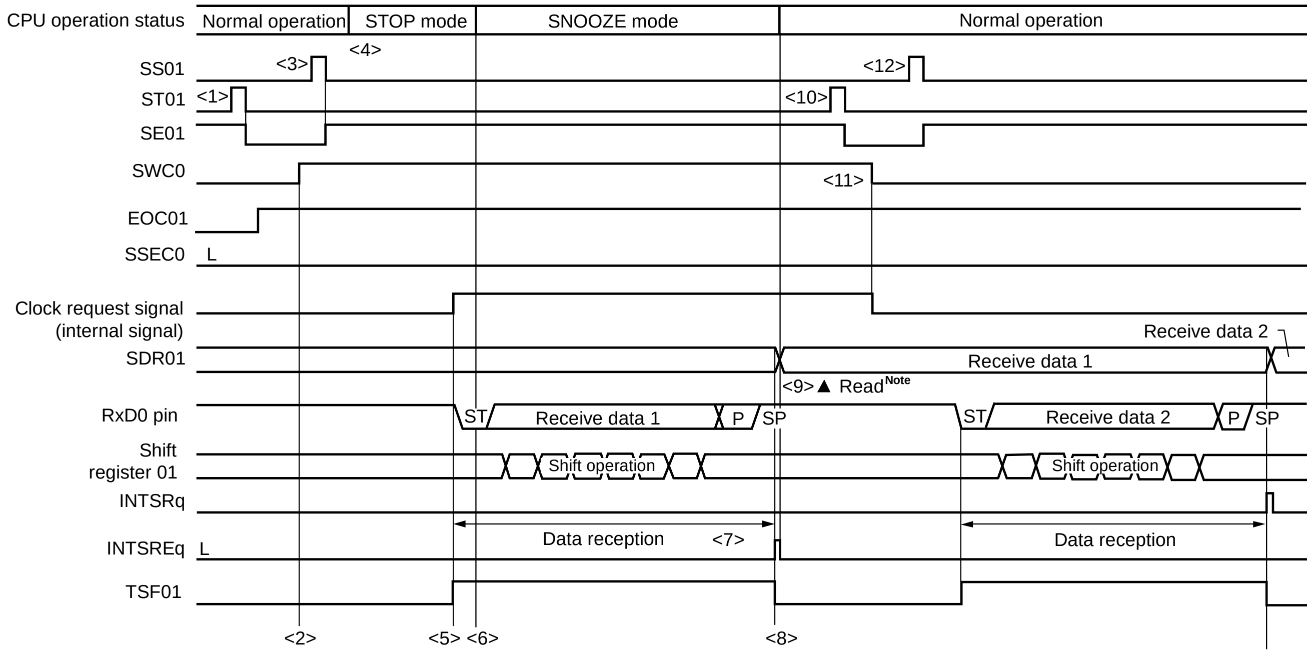
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 20-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

(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 20-93. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



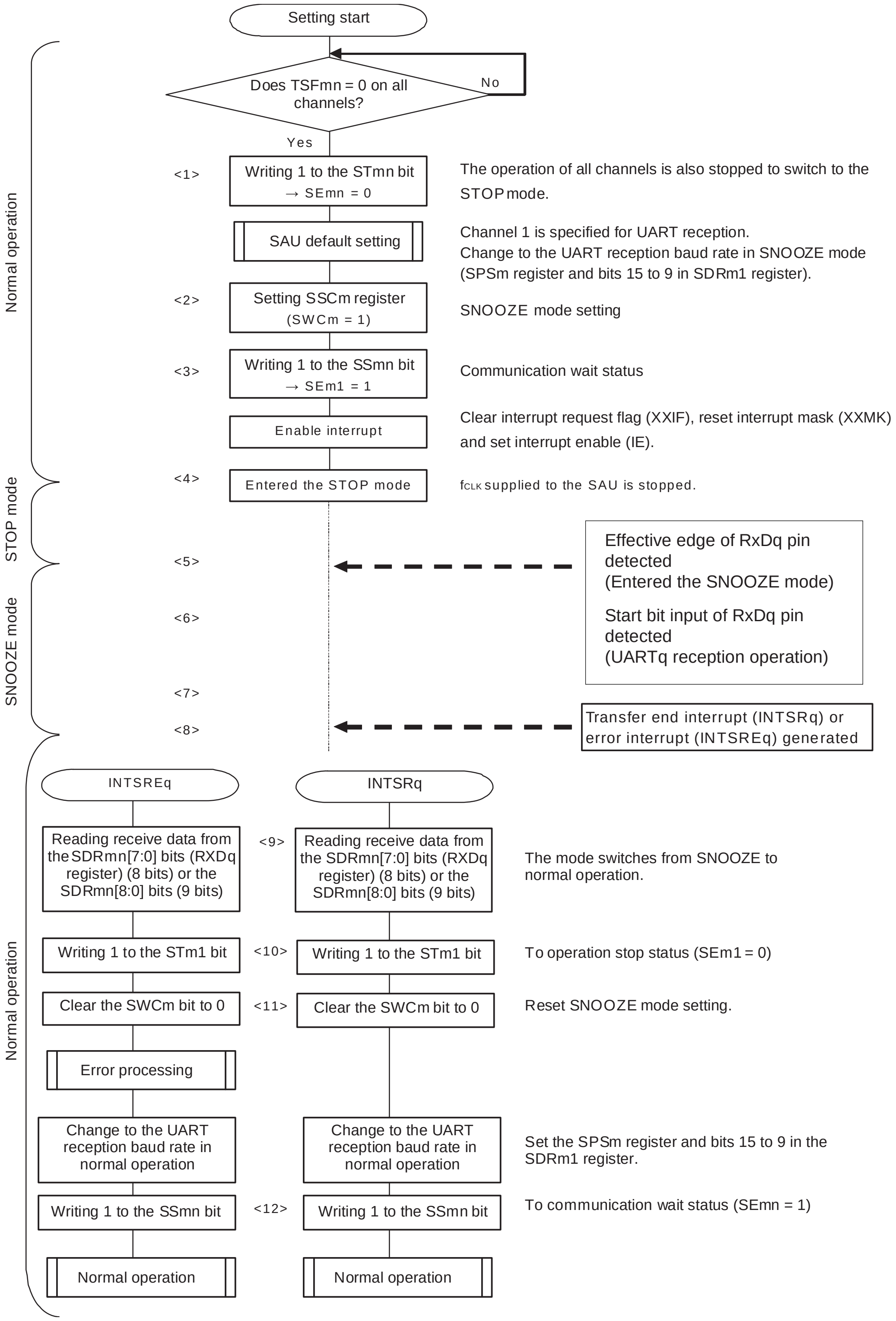
Note Read the received data when SWCm is 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 20-94 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

Figure 20-94. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)


Setting start
Does TSFmn = 0 on all channels?
No
Yes
<1>
Writing 1 to the STmn bit → SEmn = 0
The operation of all channels is also stopped to switch to the STOP mode.
SAU default setting
Channel 1 is specified for UART reception.
Change to the UART reception baud rate in SNOOZE mode (SPSm register and bits 15 to 9 in SDRm1 register).
<2>
Setting SSCm register (SWCm = 1)
SNOOZE mode setting
<3>
Writing 1 to the SSmn bit → SEm1 = 1
Communication wait status
Enable interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (IE).
Normal operation
<4>
Entered the STOP mode
fCLK supplied to the SAU is stopped.
STOP mode
<5>
Effective edge of RxDq pin detected
(Entered the SNOOZE mode)
<6>
Start bit input of RxDq pin detected
(UARTq reception operation)
SNOOZE mode
<7>
<8>
Transfer end interrupt (INTSRq) or error interrupt (INTSREq) generated
INTSREq
INTSRq
<9>
Reading receive data from the SDRmn[7:0] bits (RXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
Reading receive data from the SDRmn[7:0] bits (RXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
The mode switches from SNOOZE to normal operation.
<10>
Writing 1 to the STm1 bit
Writing 1 to the STm1 bit
To operation stop status (SEm1 = 0)
<11>
Clear the SWCm bit to 0
Clear the SWCm bit to 0
Reset SNOOZE mode setting.
Error processing
Change to the UART reception baud rate in normal operation
Change to the UART reception baud rate in normal operation
Set the SPSm register and bits 15 to 9 in the SDRm1 register.
<12>
Writing 1 to the SSmn bit
Writing 1 to the SSmn bit
To communication wait status (SEmn = 1)
Normal operation
Normal operation
Normal operation

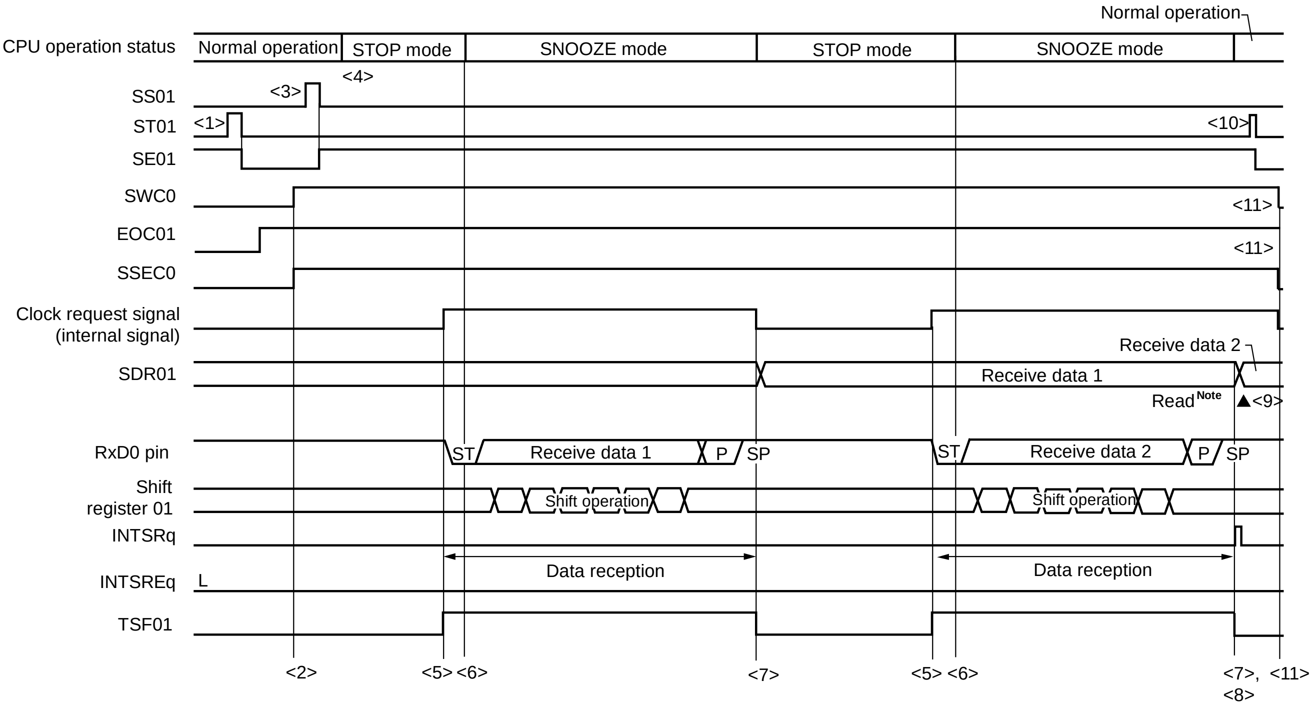
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 20-92 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 20-93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

(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 20-95. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

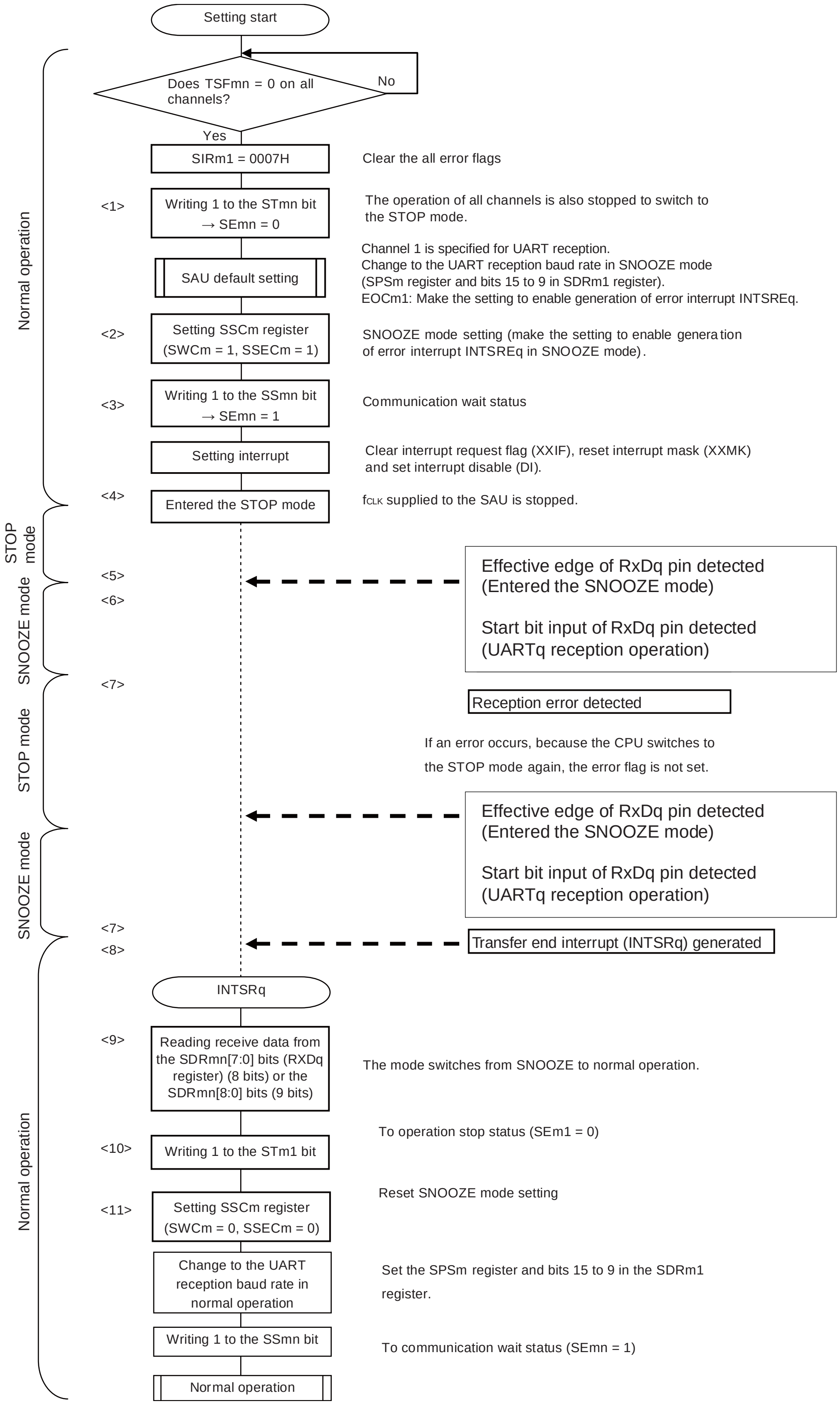
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

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, or OVFm1 flag 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).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 20-96 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0

Figure 20-96. Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)


Setting start
Does TSFmn = 0 on all channels?
No
Yes
SIRm1 = 0007H
<1>
Writing 1 to the STmn bit → SEmn = 0
SAU default setting
<2>
Setting SSCm register (SWCm = 1, SSECm = 1)
<3>
Writing 1 to the SSmn bit → SEmn = 1
Setting interrupt
<4>
Entered the STOP mode
<5>
<6>
<7>
<7>
<8>
INTSRq
<9>
Reading receive data from the SDRmn[7:0] bits (RXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits)
<10>
Writing 1 to the STm1 bit
<11>
Setting SSCm register (SWCm = 0, SSECm = 0)
Change to the UART reception baud rate in normal operation
Writing 1 to the SSmn bit
Normal operation
Normal operation
STOP mode
SNOOZE mode
STOP mode
SNOOZE mode
Normal operation
Effective edge of RxDq pin detected (Entered the SNOOZE mode)
Start bit input of RxDq pin detected (UARTq reception operation)
Reception error detected
Effective edge of RxDq pin detected (Entered the SNOOZE mode)
Start bit input of RxDq pin detected (UARTq reception operation)
Transfer end interrupt (INTSRq) generated

Clear the all error flags

The operation of all channels is also stopped to switch to the STOP mode.

Channel 1 is specified for UART reception.
Change to the UART reception baud rate in SNOOZE mode
(SPSm register and bits 15 to 9 in SDRm1 register).
EOCm1: Make the setting to enable generation of error interrupt INTSREq.

SNOOZE mode setting (make the setting to enable generation of error interrupt INTSREq in SNOOZE mode).

Communication wait status

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt disable (DI).

f_{CLK} supplied to the SAU is stopped.

If an error occurs, because the CPU switches to the STOP mode again, the error flag is not set.

The mode switches from SNOOZE to normal operation.

To operation stop status (SEm1 = 0)

Reset SNOOZE mode setting

Set the SPSm register and bits 15 to 9 in the SDRm1 register.

To communication wait status (SEmn = 1)

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

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, or OVFm1 flag 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).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 20-95 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0

20.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART4) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 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.

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 20-4. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz

Note When changing the clock selected for f_{CLK} (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).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART4) 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.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 – 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 24 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	77	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	77	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	77	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	77	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	77	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	77	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	77	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	47	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	77	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	77	76923.1 bps	+0.16%
153600 bps	f_{CLK}	77	153846 bps	+0.16%
312500 bps	f_{CLK}	37	315789 bps	+1.05%

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12, 20

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART4) 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.

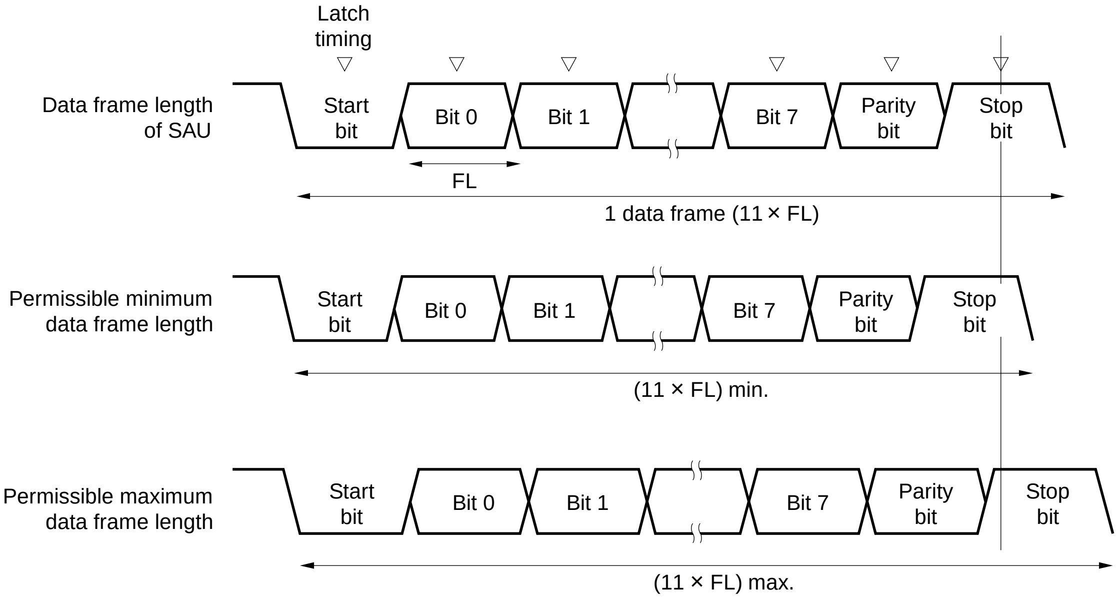
$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times \text{Brate}$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **20.6.4 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0 to 2), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13, 21

Figure 20-97. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 20-97**, 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.

20.6.5 Procedure for processing errors that occurred during UART (UART0 to UART4) communication

The procedure for processing errors that occurred during UART (UART0 to UART4) communication is described in **Figures 20-98** and **20-99**.

Figure 20-98. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	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).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 20-99. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	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).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn).	Error flag is cleared.	Error can be cleared only during reading, 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 operating.	
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 to 2), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13, 20, 21

20.7 LIN Communication Operation

20.7.1 LIN transmission

Of UART transmission, UART0 support LIN communication.

For LIN transmission, channel 0 of unit 0 is used.

UART	UART0	UART1	UART2	UART3	UART4
Support of LIN communication	Supported	Not supported	Not supported	Not supported	Not supported
Target channel	Channel 0 of SAU0	–	–	–	–
Pins used	TxD0	–	–	–	–
Interrupt	INTST0	–	–	–	–
	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	Max. f_{MCK}/6 [bps] (SDR00[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]Note				
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 in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**). In addition, LIN communication is usually 2.4/9.6/19.2 kbps is often used.

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.

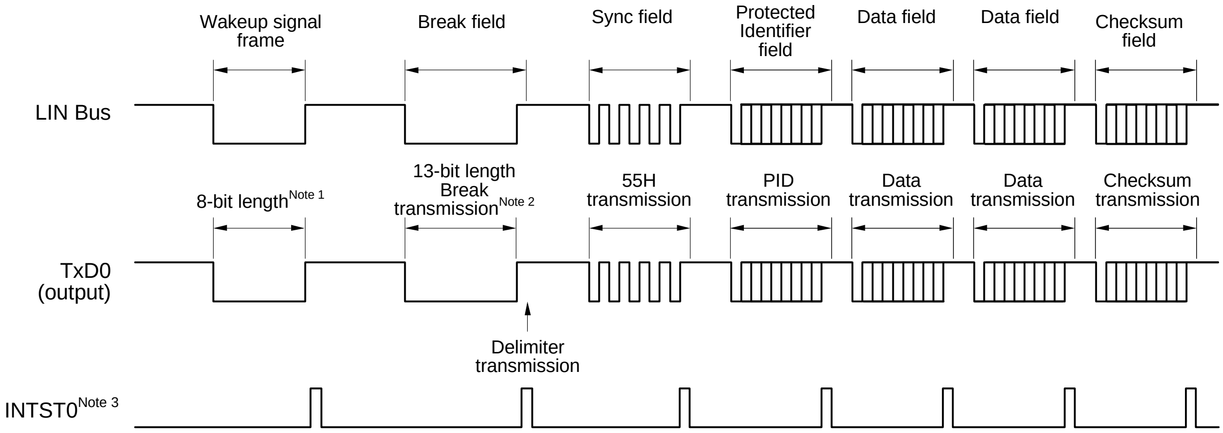
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 ±15%, communication can be established.

Figure 20-100 outlines a master transmission operation of LIN.

Figure 20-100. Transmission Operation of LIN



Notes 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

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.

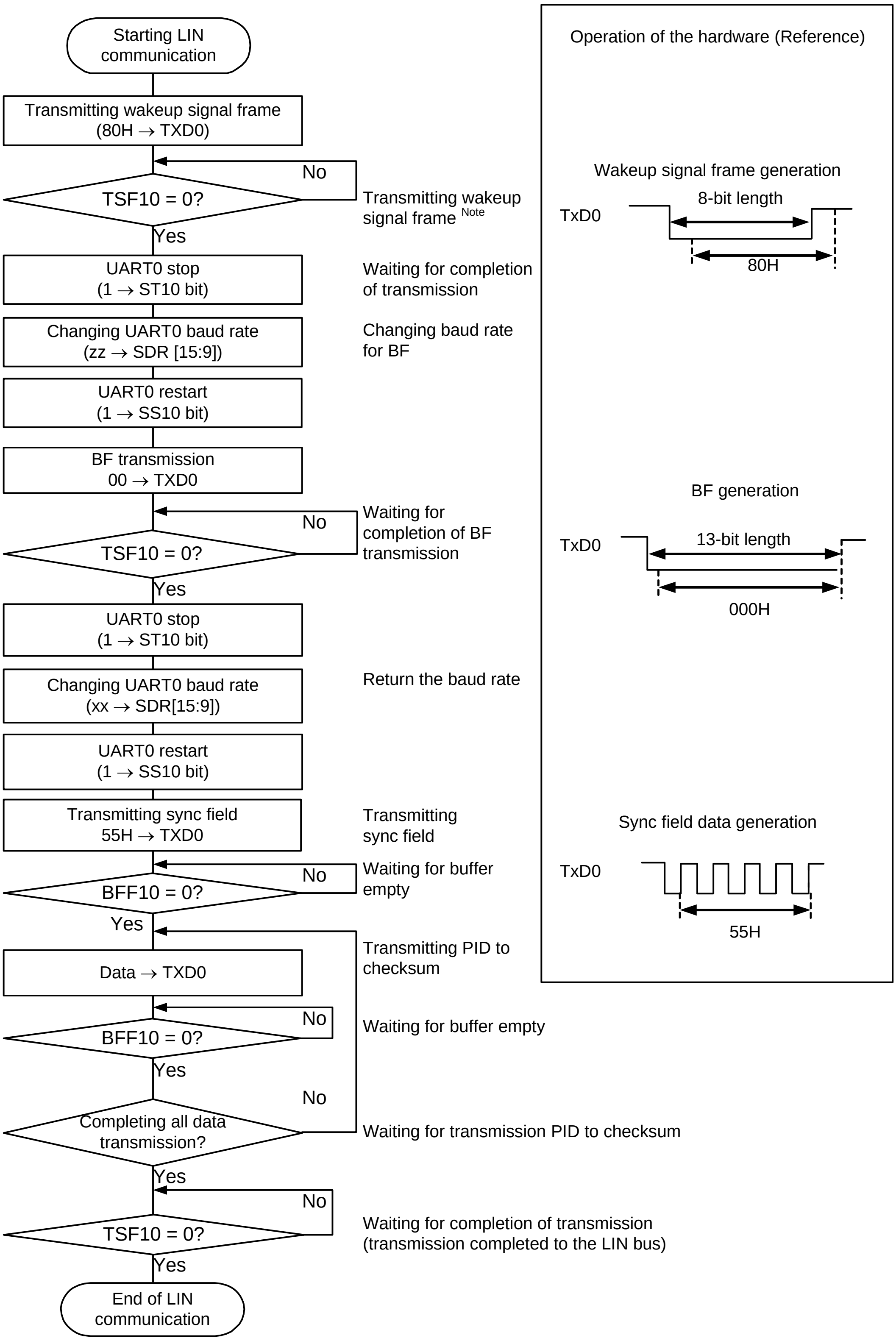
$\boxed{\text{(Baud rate of break field)} = 9/13 \times N}$

By transmitting data of 00H at this baud rate, a break field is generated.

3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 20-101. Flowchart for LIN Transmission


Starting LIN communication
Transmitting wakeup signal frame (80H → TXD0)
TSF10 = 0?
No
Yes
Transmitting wakeup signal frame Note
UART0 stop (1 → ST10 bit)
Waiting for completion of transmission
Changing UART0 baud rate (zz → SDR [15:9])
Changing baud rate for BF
UART0 restart (1 → SS10 bit)
BF transmission 00 → TXD0
TSF10 = 0?
No
Yes
Waiting for completion of BF transmission
UART0 stop (1 → ST10 bit)
Changing UART0 baud rate (xx → SDR[15:9])
Return the baud rate
UART0 restart (1 → SS10 bit)
Transmitting sync field 55H → TXD0
Transmitting sync field
BFF10 = 0?
No
Yes
Waiting for buffer empty
Data → TXD0
Transmitting PID to checksum
BFF10 = 0?
No
Yes
Waiting for buffer empty
Completing all data transmission?
No
Yes
Waiting for transmission PID to checksum
TSF10 = 0?
No
Yes
Waiting for completion of transmission (transmission completed to the LIN bus)
End of LIN communication
Operation of the hardware (Reference)
Wakeup signal frame generation
TxD0
8-bit length
80H
BF generation
TxD0
13-bit length
000H
Sync field data generation
TxD0
55H

Note When LIN-bus start from sleep status only

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

20.7.2 LIN reception

Of UART reception, UART0 support LIN communication.

For LIN reception, channel 1 of unit 0 is used.

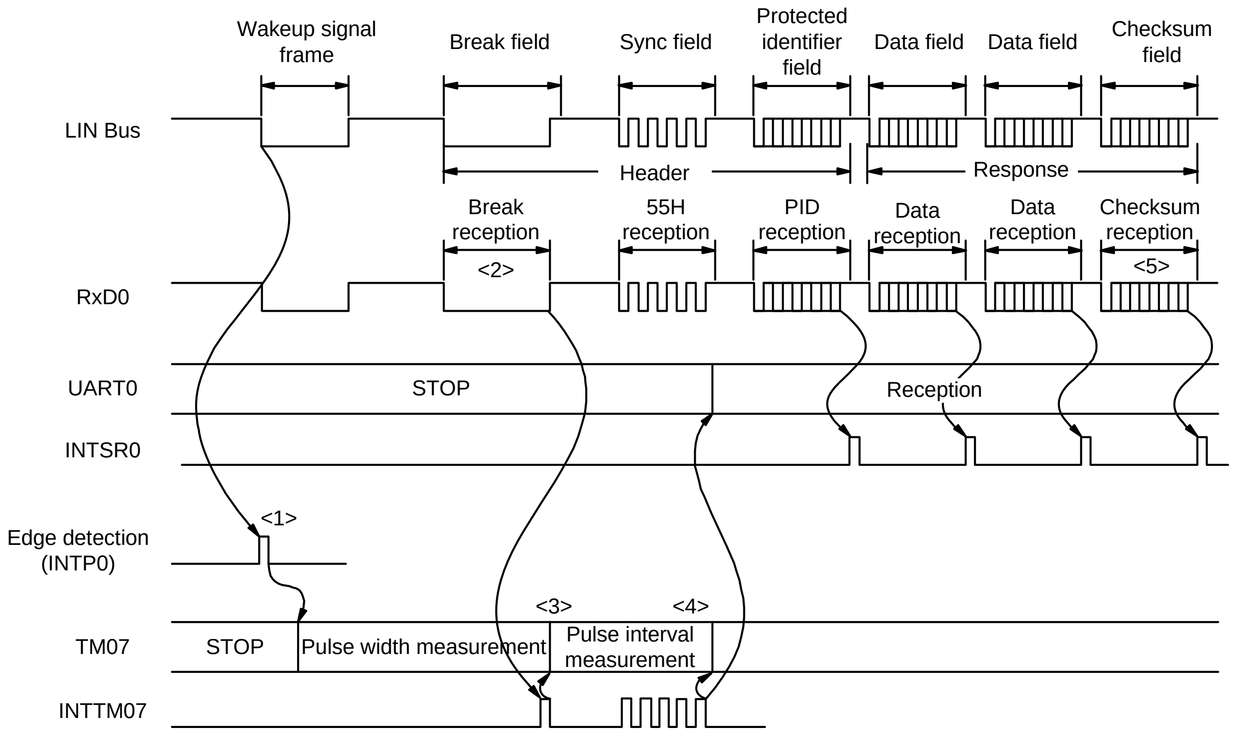
UART	UART0	UART1	UART2	UART3	UART4
Support of LIN communication	Supported	Not supported	Not supported	Not supported	Not supported
Target channel	Channel 1 of SAU0	–	–	–	–
Pins used	RxD0	–	–	–	–
Interrupt	INTSR0	–	–	–	–
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)				
Error interrupt	INTSRE0	–	–	–	–
Error detection flag	● Framing error detection flag (FEF01) ● Overrun error detection flag (OVF01)				
Transfer data length	8 bits				
Transfer rate	Max. f_{MCK}/6 [bps] (SDR01[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps][Note]				
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 in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 20-102 outlines a reception operation of LIN.

Figure 20-102. Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM07 to pulse width measurement upon detection of the wakeup signal 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.

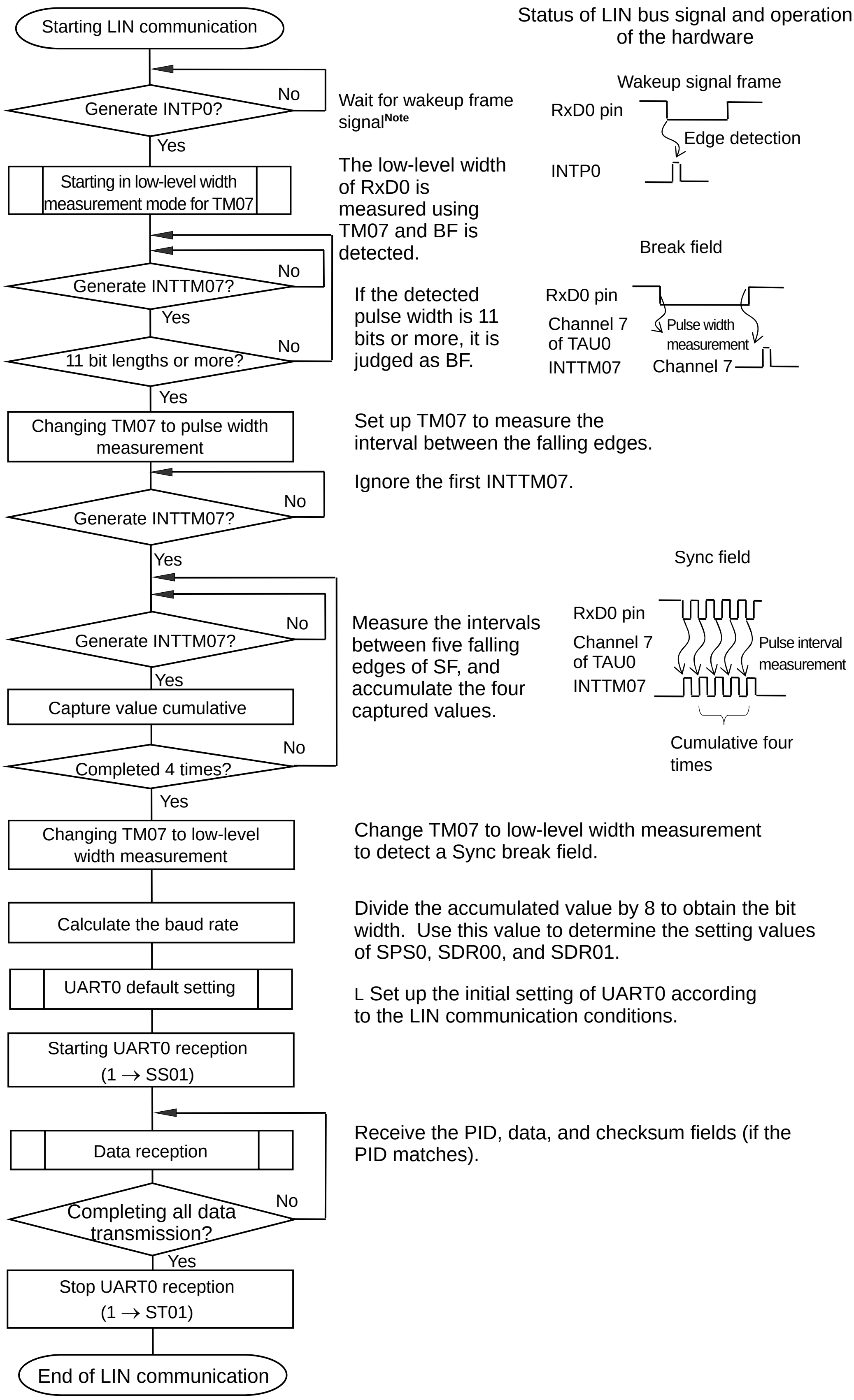
<3> When the BF signal has been received normally, change TM07 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times.

<4> When BF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **8.8.3 Operation as input pulse interval measurement**).

<5> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<6> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 20-103. Flowchart for LIN Reception


Starting LIN communication
Status of LIN bus signal and operation of the hardware
Generate INTP0?
No
Yes
Wait for wakeup frame signalNote
Wakeup signal frame
RxD0 pin
Edge detection
INTP0
Starting in low-level width measurement mode for TM07
The low-level width of RxD0 is measured using TM07 and BF is detected.
Break field
Generate INTTM07?
No
Yes
RxD0 pin
Channel 7 of TAU0
INTTM07
Pulse width measurement
Channel 7
If the detected pulse width is 11 bits or more, it is judged as BF.
11 bit lengths or more?
No
Yes
Changing TM07 to pulse width measurement
Set up TM07 to measure the interval between the falling edges.
Ignore the first INTTM07.
Generate INTTM07?
No
Yes
Sync field
Generate INTTM07?
No
Yes
Measure the intervals between five falling edges of SF, and accumulate the four captured values.
RxD0 pin
Channel 7 of TAU0
INTTM07
Pulse interval measurement
Capture value cumulative
Cumulative four times
Completed 4 times?
No
Yes
Changing TM07 to low-level width measurement
Change TM07 to low-level width measurement to detect a Sync break field.
Calculate the baud rate
Divide the accumulated value by 8 to obtain the bit width. Use this value to determine the setting values of SPS0, SDR00, and SDR01.
UART0 default setting
L Set up the initial setting of UART0 according to the LIN communication conditions.
Starting UART0 reception
(1 → SS01)
Data reception
Receive the PID, data, and checksum fields (if the PID matches).
Completing all data transmission?
No
Yes
Stop UART0 reception
(1 → ST01)
End of LIN communication

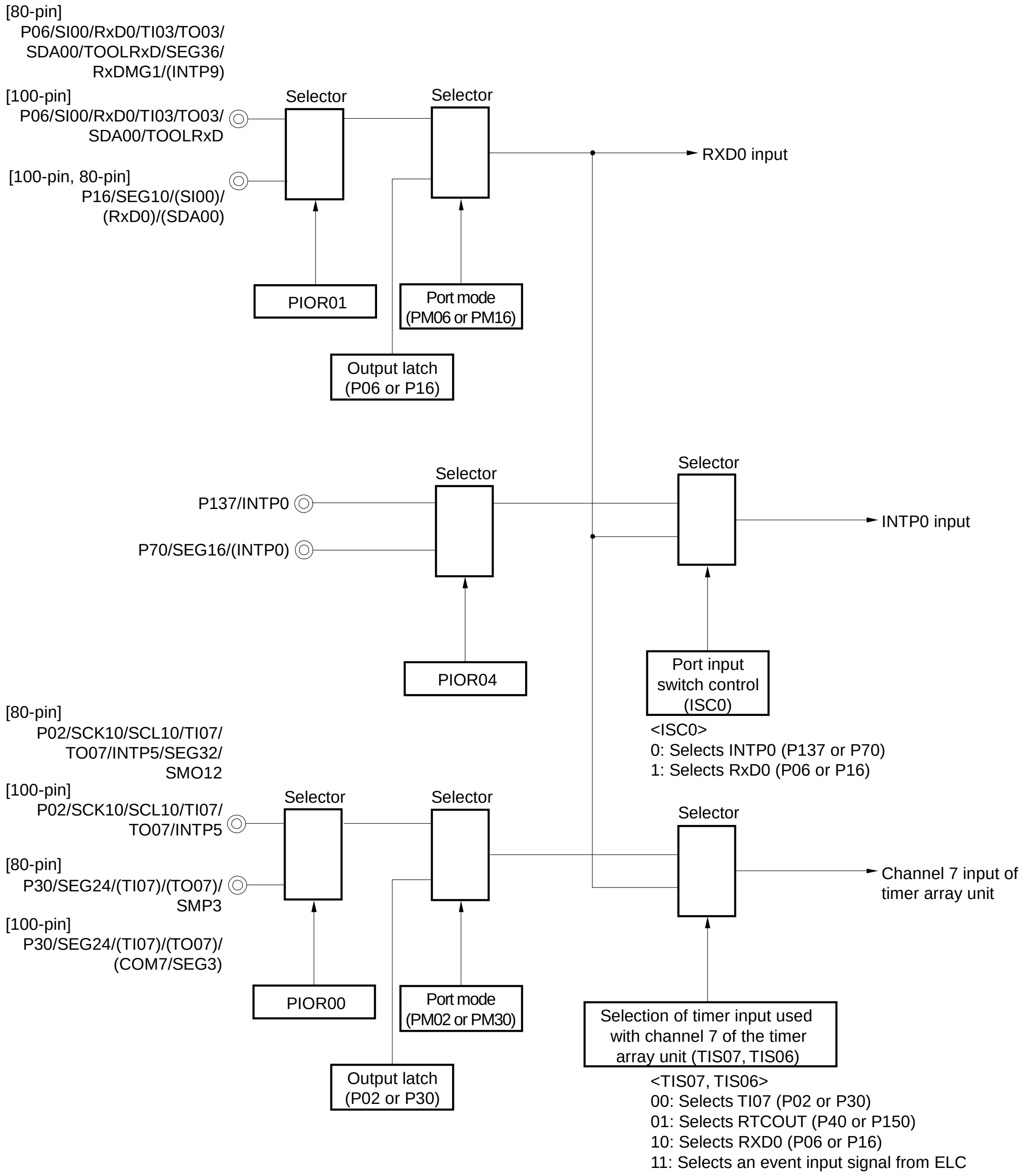
Note Required in the sleep status only.

Figure 20-104 shows the configuration of a port that manipulates reception of LIN.

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 controlling switch of port input (ISC0), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 20-104. Port Configuration for Manipulating Reception of LIN



Remarks 1. ISC0: Bit 0 of the input switch control register (ISC) (See **Figure 20-22**.)

PIOR00, PIOR01, PIOR04: Bits 0 to 4 of the peripheral I/O redirection register (PIOR0) (See **Figure 4-7**.)

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

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 an error (The interval of the edge input to RxD0 is measured in the capture mode.)
  Measured the low-level width, determine whether break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

20.8 Operation of Simplified I^{2}C (IIC00, IIC10, IIC30) 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 communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I^{2}C 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]

- Parity error (ACK error)

* [Functions not supported by simplified I^{2}C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Wait detection function

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 the processing flow in **20.8.3 (2)** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

The channels supporting simplified I^{2}C (IIC00, IIC10, IIC30) are channels 0 and 2 of SAU0 and channel 2 of SAU1.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0 (supporting LIN-bus)	IIC00
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	UART2 (supporting IrDA)	–
	1	–		–
	2	CSI30Note	UART3Note	IIC30Note
	3	–		–
2	0	–	UART4Note	–
	1	–		–

Note 100-pin product only

Simplified I^{2}C (IIC00, IIC10, IIC30) performs the following four types of communication operations.

● Address field transmission	(See **20.8.1**.)
● Data transmission	(See **20.8.2**.)
● Data reception	(See **20.8.3**.)
● Stop condition generation	(See **20.8.4**.)

20.8.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^{2}C 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	IIC10	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCL00, SDA00[Note 1]	SCL10, SDA10[Note 1]	SCL30, SDA30[Note 1]
Interrupt	INTIIC00	INTIIC10	INTIIC30
	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] (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-reversed output (default: high level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit (for ACK reception timing)		
Data direction	MSB first		

Notes 1. To perform communication via simplified I^{2}C, set the N-ch open-drain output (EV$_{DD}$ tolerance) mode for the port output mode register (POM0) (see **4.3.5 Port output mode registers (POMxx)** for details). When IIC00, IIC10, IIC30 communicating with an external device with a different potential, set the N-ch open-drain output (EV$_{DD}$ tolerance) mode also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-105. Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC00, IIC10, IIC30) (1/2)

(a) Serial mode register mn (SMRmn)

	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 0	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Operation clock (f$_{MCK}$) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Operation mode of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	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 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							
									SIOr							

(d) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(e) Serial output enable register m (SOEm)

	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	0	SOEm2 0/1	0	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

2. □ : Setting is fixed in the IIC 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

Figure 20-105. Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC00, IIC10, IIC30) (2/2)

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

SSmn = 0 until the start condition is generated, and SSmn = 1 after generation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

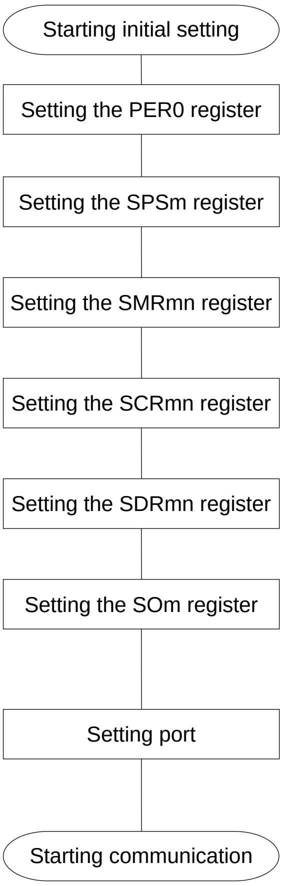
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 20-106. Initial Setting Procedure for Simplified I^{2}C


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Setting port
Starting communication

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

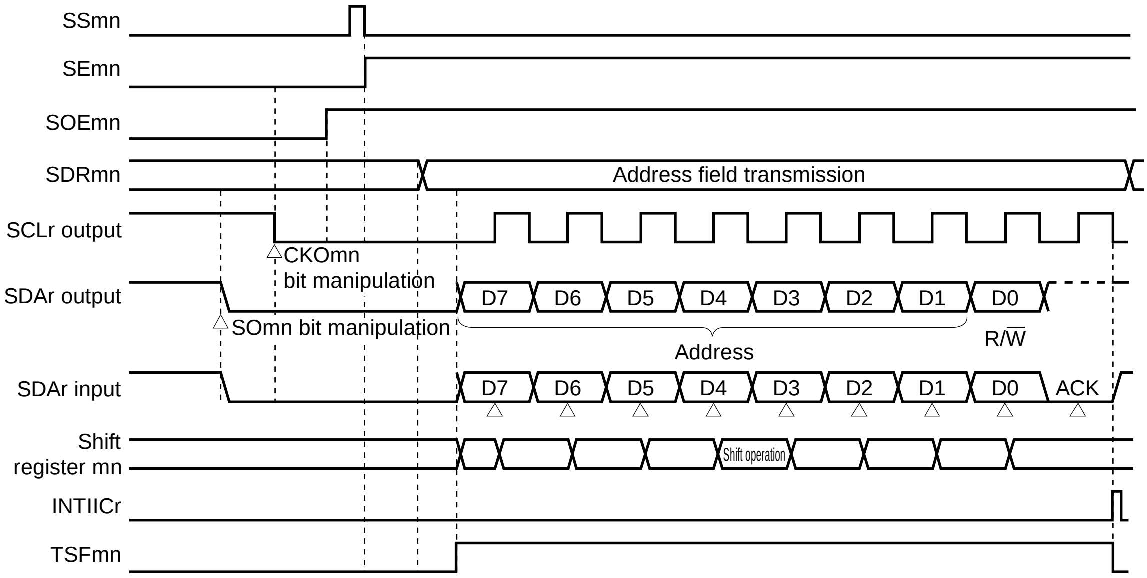
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level (1) of the serial data (SOmn) and serial clock (CKOmn).

Enable data output, clock output, and N-ch open-drain output (EV_{DD} tolerance) mode of the target channel by setting the port register, port mode register, and port output mode register.

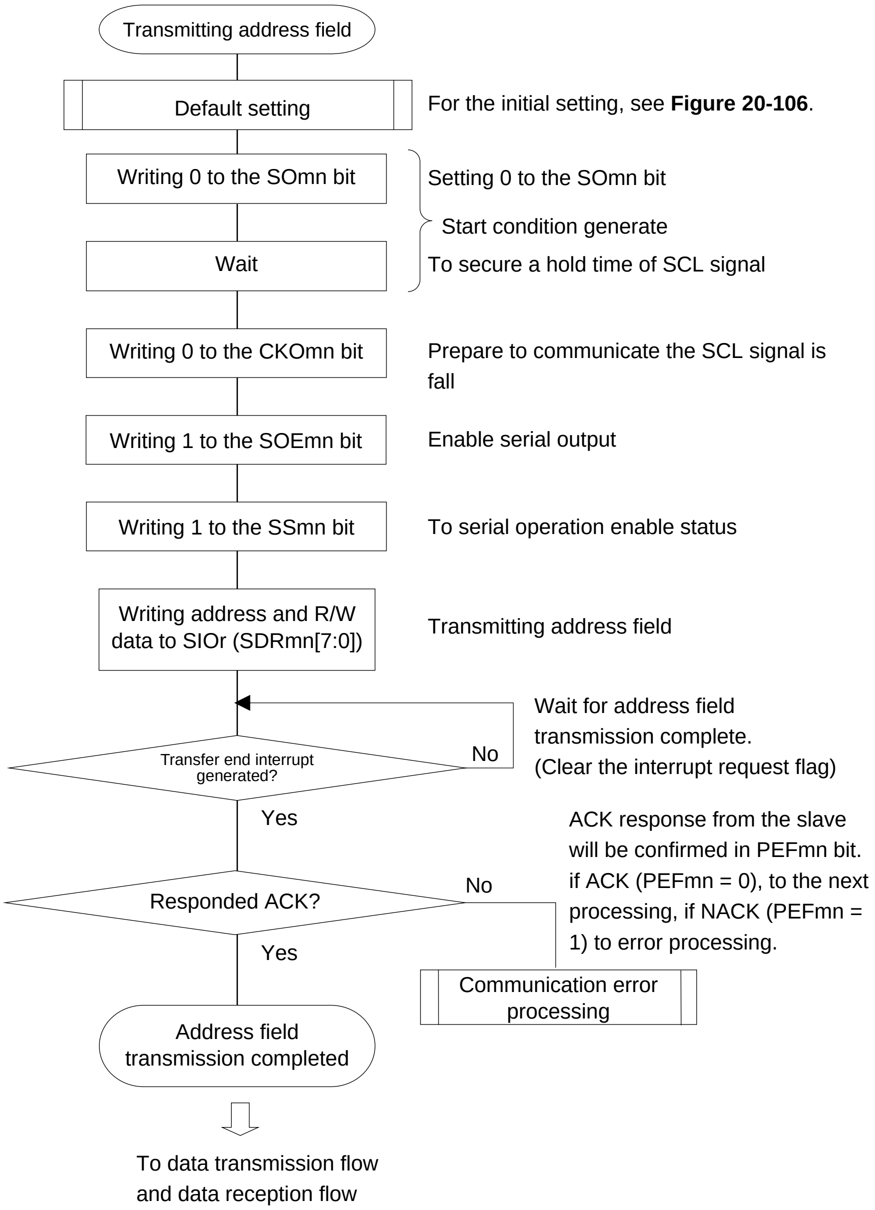
(3) Processing flow

Figure 20-107. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

Figure 20-108. Flowchart of Simplified I²C Address Field Transmission


Transmitting address field
Default setting
For the initial setting, see **Figure 20-106**.
Writing 0 to the SOmn bit
Setting 0 to the SOmn bit
Start condition generate
Wait
To secure a hold time of SCL signal
Writing 0 to the CKOmn bit
Prepare to communicate the SCL signal is fall
Writing 1 to the SOEmn bit
Enable serial output
Writing 1 to the SSmn bit
To serial operation enable status
Writing address and R/W data to SIOr (SDRmn[7:0])
Transmitting address field
Transfer end interrupt generated?
No
Wait for address field transmission complete.
(Clear the interrupt request flag)
Yes
Responded ACK?
No
ACK response from the slave will be confirmed in PEFmn bit. if ACK (PEFmn = 0), to the next processing, if NACK (PEFmn = 1) to error processing.
Yes
Communication error processing
Address field transmission completed
To data transmission flow and data reception flow

20.8.2 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^2C	IIC00	IIC10	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCL00, SDA00[Note 1]	SCL10, SDA10[Note 1]	SCL30, SDA30[Note 1]
Interrupt	INTIIC00	INTIIC10	INTIIC30
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	ACK error flag (PEFmn)		
Transfer data length	8 bits		
Transfer rate[Note 2]	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. ● Max. 1 MHz (fast mode plus) ● Max. 400 kHz (fast mode) ● Max. 100 kHz (standard mode)		
Data level	Non-reversed output (default: high level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit (for ACK reception timing)		
Data direction	MSB first		

Notes 1. To perform communication via simplified I^2C, set the N-ch open-drain output (EV_{DD} tolerance) mode for the port output mode registers (POM0) (see **4.3.5 Port output mode registers (POMxx)** for details). When IIC00, IIC10, IIC30 communicating with an external device with a different potential, set the N-ch open-drain output (EV_{DD} tolerance) mode also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-109. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC10, IIC30) (1/2)

(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 0	0	SISmn0 0	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 0	SLCmn0 1	0	1	DLSmn1 1 [Note 1]	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr) … During data transmission/reception, valid only lower 8-bits (SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting [Note 2]							0	Transmit data setting							
									SIOr							

(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	1	CKOm2 0/1 [Note3]	1	CKOm0 0/1 [Note3]	0	0	0	0	1	SOm2 0/1 [Note3]	1	SOm0 0/1 [Note3]

(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	0	SOEm2 1	0	SOEm0 1

Notes 1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

2. Because the setting is completed by address field transmission, setting is not required.

3. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

2. □ : Setting is fixed in the IIC 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

Figure 20-109. Example of Contents of Registers for Data Transmission of Simplified I^{2}C (IIC00, IIC10) (2/2)

(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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

2. ▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 20-110. Timing Chart of Data Transmission

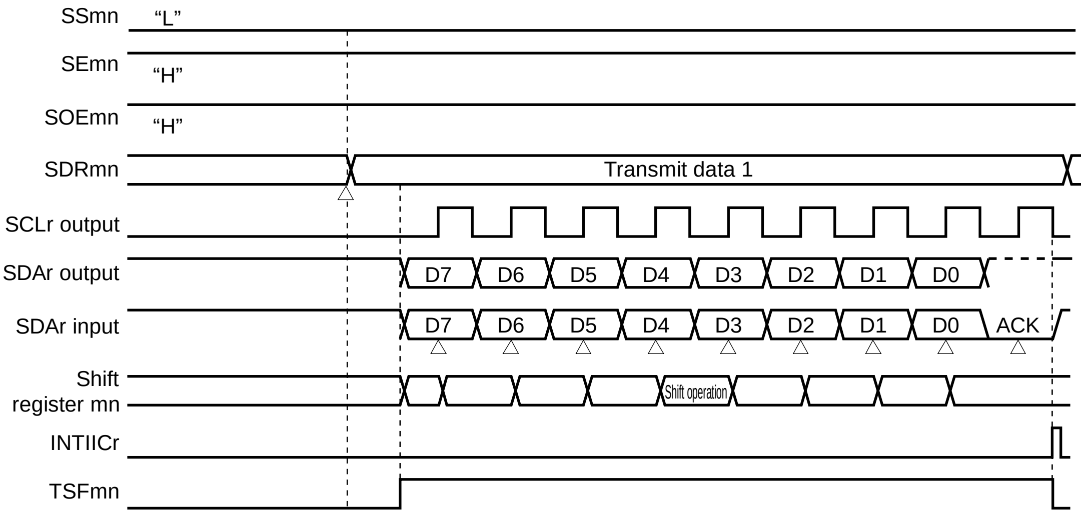
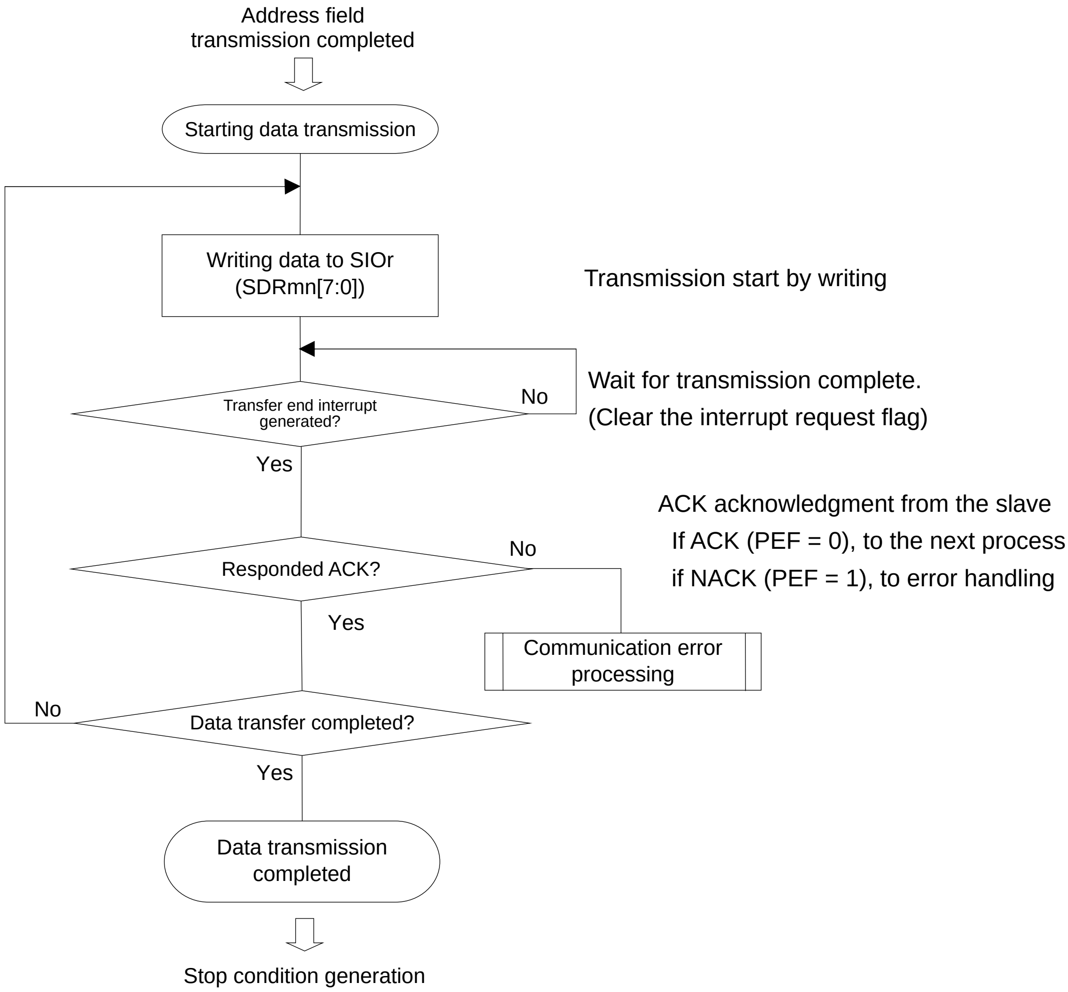


Figure 20-111. Flowchart of Simplified I²C Data Transmission


Address field
transmission completed
Starting data transmission
Writing data to SIOr
(SDRmn[7:0])
Transmission start by writing
Transfer end interrupt
generated?
No
Wait for transmission complete.
(Clear the interrupt request flag)
Yes
ACK acknowledgment from the slave
If ACK (PEF = 0), to the next process
if NACK (PEF = 1), to error handling
Responded ACK?
No
Yes
Communication error
processing
No
Data transfer completed?
Yes
Data transmission
completed
Stop condition generation

20.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC10	IIC30
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	SCL00, SDA00[Note 1]	SCL10, SDA10[Note 1]	SCL30, SDA30[Note 1]
Interrupt	INTIIC00	INTIIC10	INTIIC30
	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^2C. ● Max. 1 MHz (fast mode plus) ● Max. 400 kHz (fast mode) ● Max. 100 kHz (standard mode)		
Data level	Non-reversed output (default: high level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit (ACK transmission)		
Data direction	MSB first		

Notes 1. To perform communication via simplified I^2C, set the N-ch open-drain output (EV_{DD} tolerance) mode for the port output mode registers (POM0) (see **4.3.5 Port output mode registers (POMxx)** for details). When IIC00, IIC10, IIC30 communicating with an external device with a different potential, set the N-ch open-drain output (EV_{DD} tolerance) mode also for the clock input/output pins (SCL00, SCL10) (see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers** for details).

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 43 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(1) Register setting

Figure 20-112. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC10, IIC30) (1/2)

(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 0	0	SISmn0 0	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 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting[Note 1]							0	Dummy transmit data setting (FFH)							
									SIOr							

(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	1	CKOm2 0/1 Note 2	1	CKOm0 0/1 Note 2	0	0	0	0	1	SOm2 0/1 Note 2	1	SOm0 0/1 Note 2

(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	0	SOEm2 0/1	0	SOEm0 0/1

Notes 1. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.

2. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

2. □ : Setting is fixed in the IIC 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

Figure 20-112. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC10, IIC30) (2/2)

(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 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

2. □ : Setting is fixed in the simplified SPI (CSI) master transmission mode,

▨ : Setting disabled (set to the initial value)

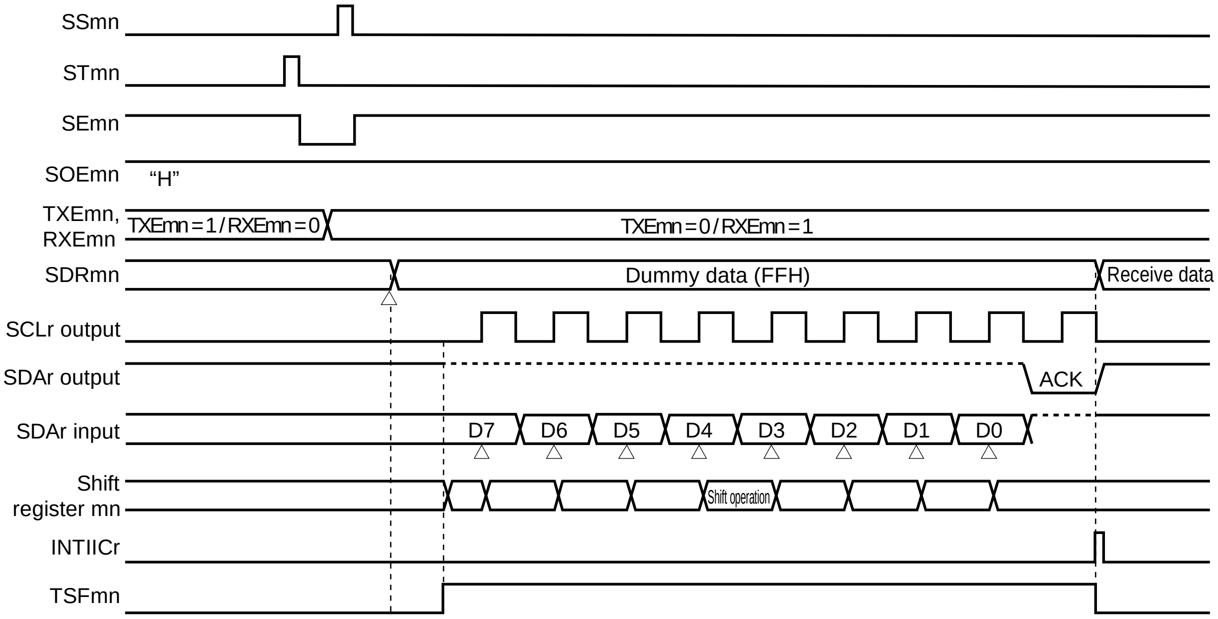
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

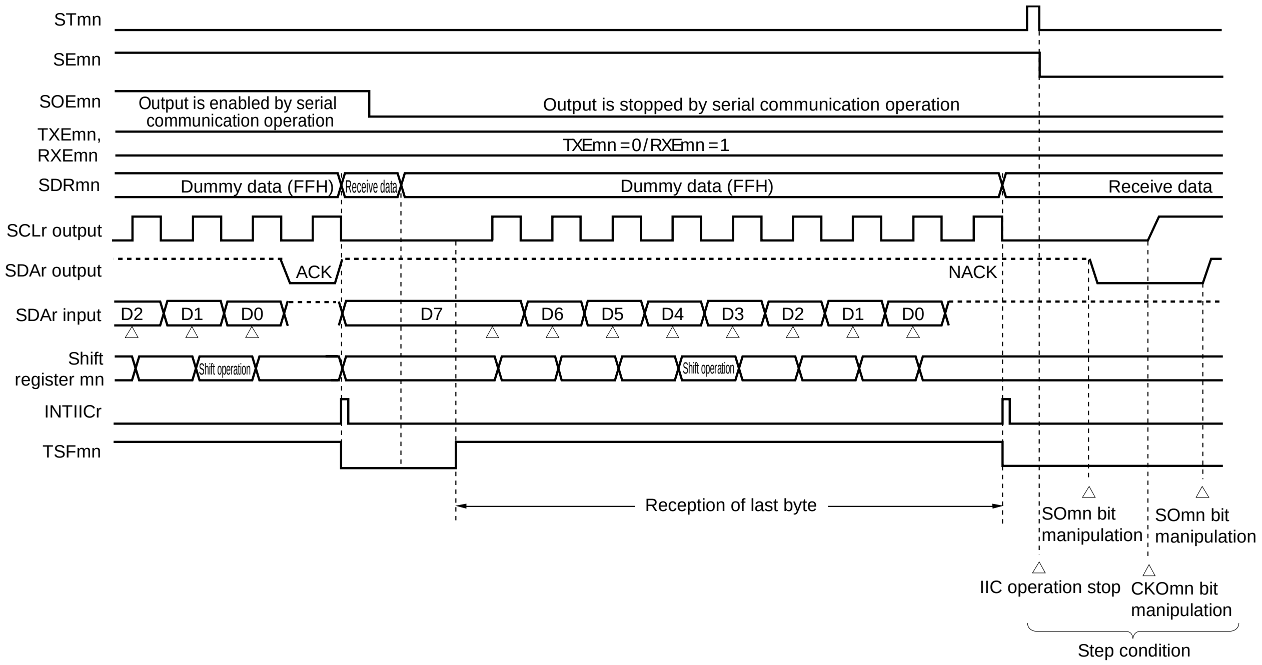
(2) Processing flow

Figure 20-113. Timing Chart of Data Reception

(a) When starting data reception

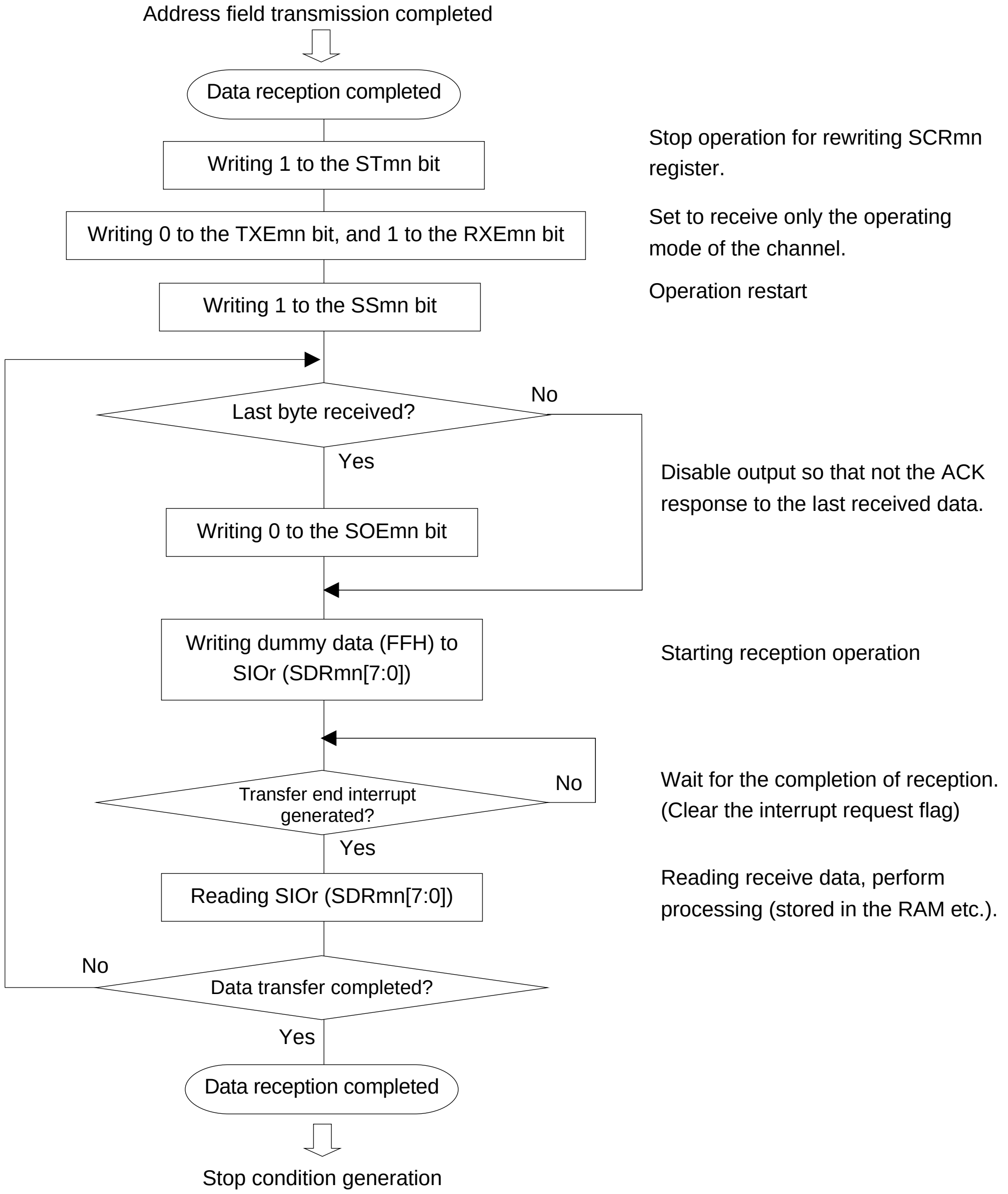


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

Figure 20-114. Flowchart of Data Reception


Address field transmission completed
Data reception completed
Writing 1 to the STmn bit
Stop operation for rewriting SCRmn register.
Writing 0 to the TXEmn bit, and 1 to the RXEmn bit
Set to receive only the operating mode of the channel.
Writing 1 to the SSmn bit
Operation restart
Last byte received?
No
Yes
Writing 0 to the SOEmn bit
Disable output so that not the ACK response to the last received data.
Writing dummy data (FFH) to SIOr (SDRmn[7:0])
Starting reception operation
Transfer end interrupt generated?
No
Wait for the completion of reception. (Clear the interrupt request flag)
Yes
Reading SIOr (SDRmn[7:0])
Reading receive data, perform processing (stored in the RAM etc.).
No
Data transfer completed?
Yes
Data reception completed
Stop condition generation

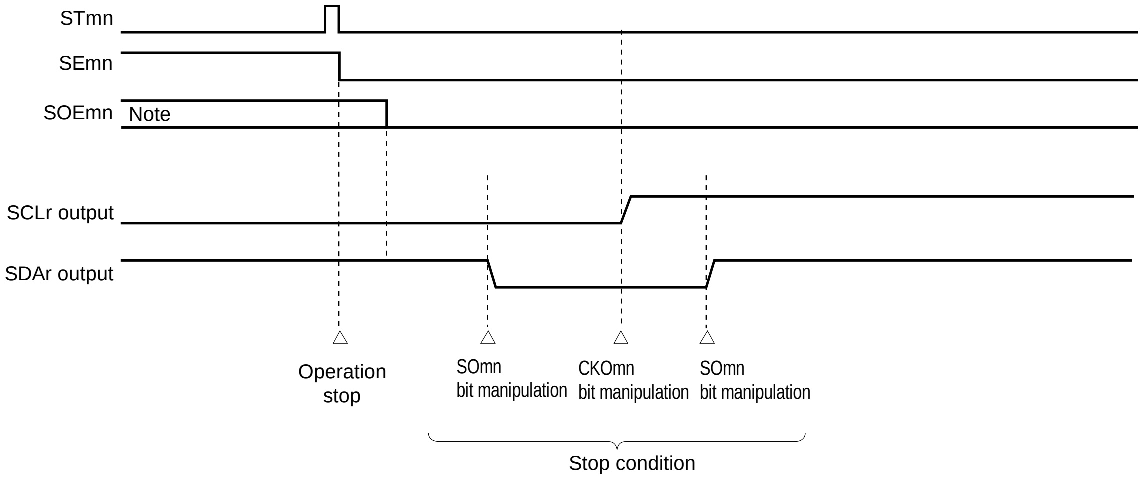
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

20.8.4 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.

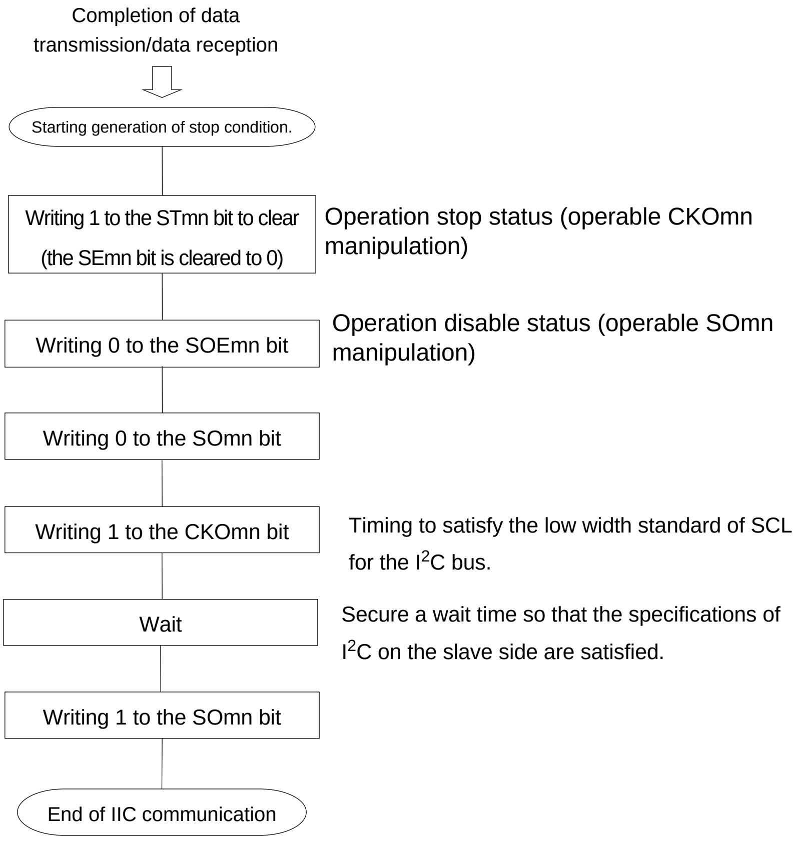
(1) Processing flow

Figure 20-115. 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 20-116. Flowchart of Stop Condition Generation


Completion of data transmission/data reception
Starting generation of stop condition.
Writing 1 to the STmn bit to clear (the SEmn bit is cleared to 0)
Operation stop status (operable CKOmn manipulation)
Writing 0 to the SOEmn bit
Operation disable status (operable SOmn manipulation)
Writing 0 to the SOmn bit
Writing 1 to the CKOmn bit
Timing to satisfy the low width standard of SCL for the I2C bus.
Wait
Secure a wait time so that the specifications of I2C on the slave side are satisfied.
Writing 1 to the SOmn bit
End of IIC communication

20.8.5 Calculating transfer rate

The transfer rate for simplified I^{2}C (IIC00, IIC10, IIC30) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 00000000B. Be sure to set a value of 00000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I^{2}C is 50%. The I^{2}C 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 I^{2}C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I^{2}C bus specifications.

Remarks 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.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 20-5. Selection of Operation Clock For Simplified I^2C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (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).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK}$ = 24 MHz.

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	29	380 kHz	5.0%Note
1 MHz	f_{CLK}	5	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%.

20.8.6 Procedure for processing errors that occurred during simplified I^{2}C (IIC00, IIC10, IIC30) communication

The procedure for processing errors that occurred during simplified I^{2}C (IIC00, IIC10, IIC30) communication is described in **Figures 20-117** and **20-118**.

Figure 20-117. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	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.

Figure 20-118. Processing Procedure in Case of ACK Error in Simplified I^{2}C Mode

Software Manipulation	Hardware Status	Remark
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.
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, 2), r: IIC number (r = 00, 10, 30), mn = 00, 02, 12

CHAPTER 21 SERIAL INTERFACE IICA

21.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^{2}C bus mode (multimaster 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 I^{2}C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^{2}C 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 a local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1).

Figure 21-1 shows a block diagram of serial interface IICA.

Remark n = 0

Figure 21-1. Block Diagram of Serial Interface IICA0

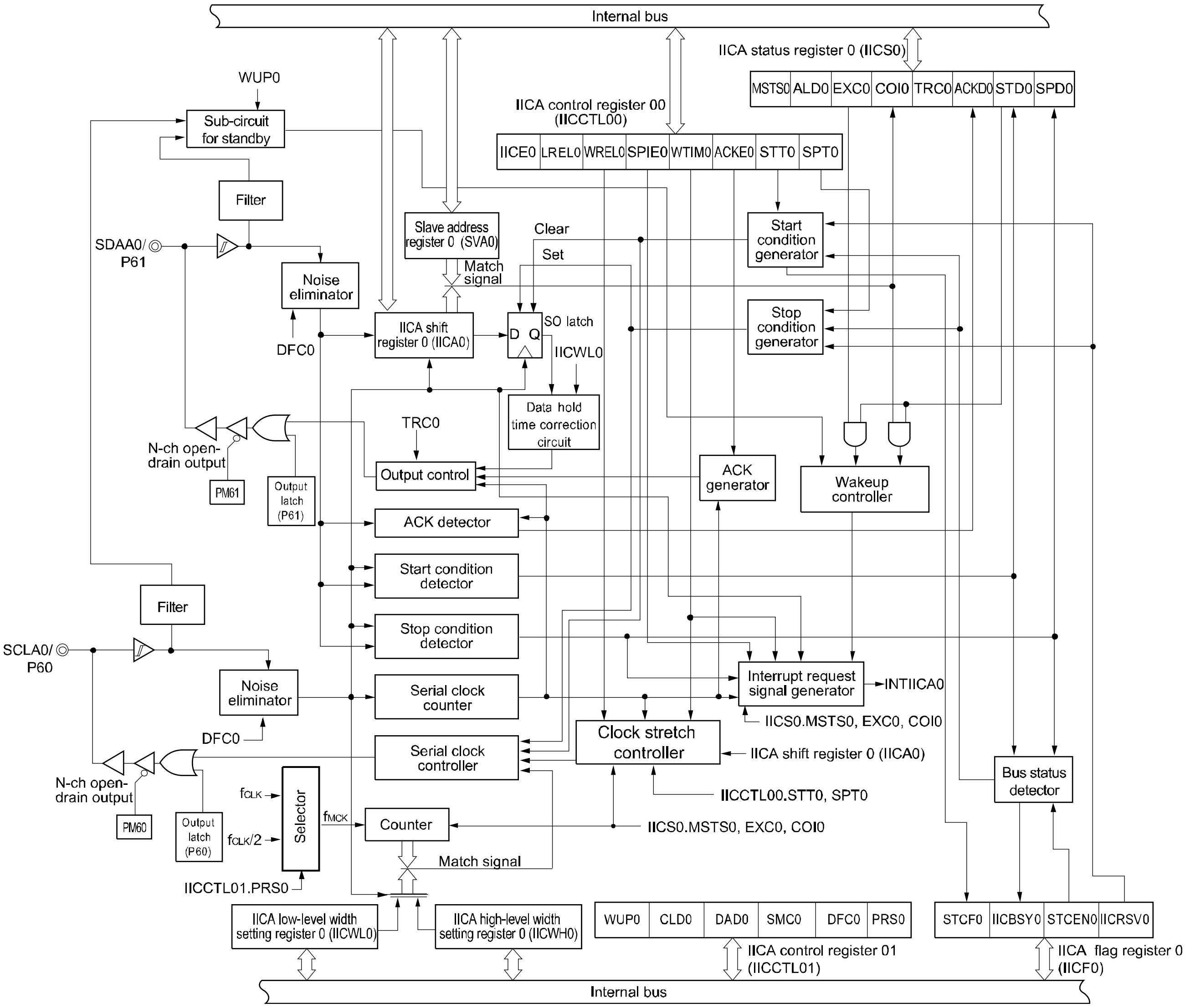
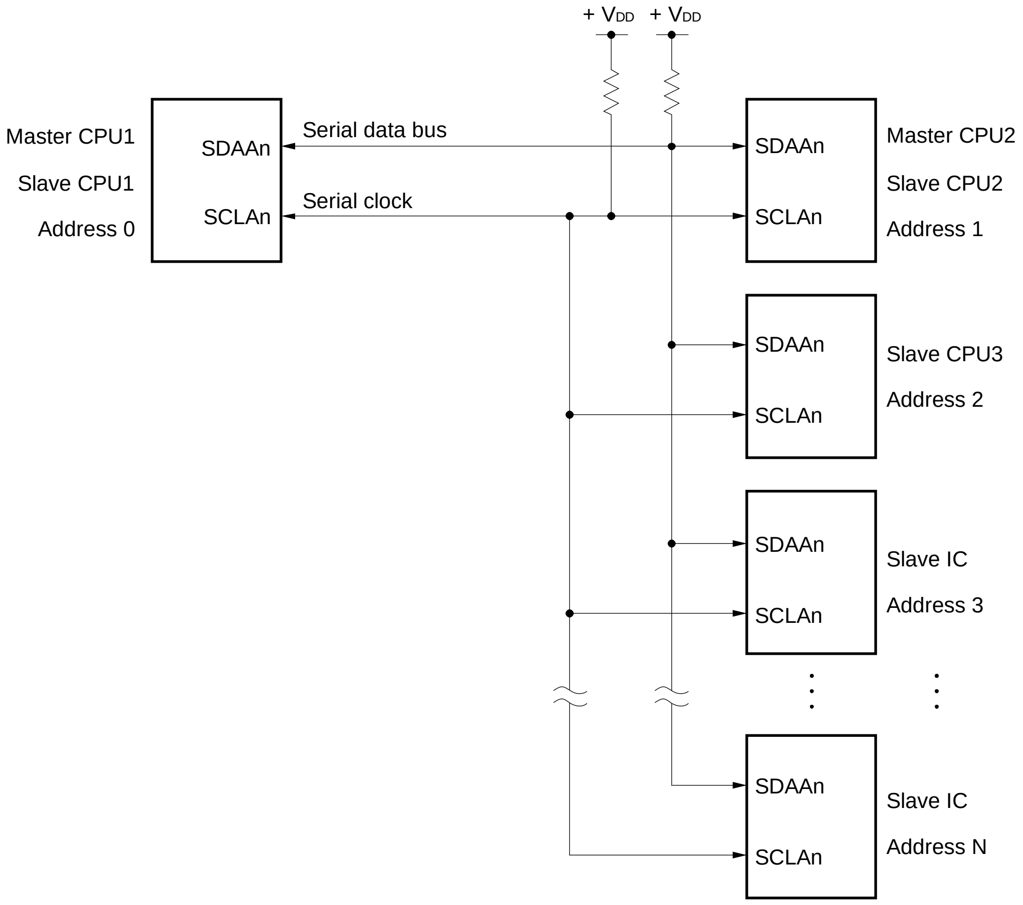


Figure 21-2 shows a serial bus configuration example.

Figure 21-2. Serial Bus Configuration Example Using I^{2}C Bus



Remark n = 0

21.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 21-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register n (IICAn) Slave address register n (SVAn)
Control registers	Peripheral enable register 0 (PER0) Peripheral Reset Control Register 0 (PRR0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (IICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode register 6 (PM6) Port register 6 (P6)

Remark n = 0

(1) IICA shift register n (IICAn)

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICAn register.

Cancel 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.

Reset signal generation clears IICAn to 00H.

Figure 21-3. Format of IICA Shift Register n (IICAn)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

Cautions 1. Do not write data to the IICAn register during data transfer.

2. Write or read 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.

3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0

(2) Slave address register n (SVAn)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} 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).

Reset signal generation clears the SVAn register to 00H.

Figure 21-4. Format of Slave Address Register n (SVAn)

Address: F0234H After reset: 00H 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 (INTIICAn) when the address received by this register matches the address value set to the slave address register n (SVAn) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks 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 I^{2}C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

Remark WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)
SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

(8) Clock stretch controller

This circuit controls the clock stretch timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

Remark n = 0

(11) Start condition generator

This circuit generates a start condition when the STTn bit is set to 1.

However, in the communication reservation disabled status (IICRSVn bit = 1), when the bus is not released (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 status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCENn bit.

Remarks 1.

STTn bit:	Bit 1 of IICA control register n0 (IICCTLn0)
SPTn bit:	Bit 0 of IICA control register n0 (IICCTLn0)
IICRSVn bit:	Bit 0 of IICA flag register n (IICFn)
IICBSYn bit:	Bit 6 of IICA flag register n (IICFn)
STCFn bit:	Bit 7 of IICA flag register n (IICFn)
STCENn bit:	Bit 1 of IICA flag register n (IICFn)

2. n = 0

21.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following ten registers.

- Peripheral enable register 0 (PER0)
- Peripheral Reset Control Register 0 (PRR0)
- 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)
- Port mode register 6 (PM6)
- Port register 6 (P6)

Remark n = 0

21.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the serial interface IICA0 is to be used, be sure to set bit 4 (IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 21-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. ● SFR used by the serial interface IICA0 cannot be written. The read value is 0H. However, the SFR is not initialized. [Note]
1	Enables input clock supply. ● SFR used by the serial interface IICA0 can be read and written.

Note To initialize the serial interface IICA0 and the SFR used by the serial interface IICA0, use bit 4 (IICA0RES) of PRR0.

Cautions 1. When setting serial interface IICAn, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 0, writing to the control registers of serial interface IICA is ignored (except for port mode register 6 (PM6) and port register 6 (P6)).

- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**

2. Be sure to clear bit 7 to “0”.

Remark n = 0

21.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the serial interface IICA0 in the reset state, be sure to set bit 4 (IICA0RES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears the PRR0 register to 00H.

Figure 21-6. Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

IICA0RES	Control resetting of the serial interface IICA0
0	The serial interface IICA0 is released from the reset state.
1	The serial interface IICA0 is in the reset state.

21.3.3 IICA control register n0 (IICCTLn0)

This register is used to enable/stop I^{2}C operations, set clock stretch timing, and set other I^{2}C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIEn, WTIMn, and ACKEn bits while IICEn = 0 or during the clock stretch period. These bits can be set at the same time when the IICEn bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Remark n = 0

Figure 21-7. Format of IICA Control Register n0 (IICCTLn0) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I^2C operation enable
0	Stop operation. Reset the IICA status register n (IICSn)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
● Cleared by instruction ● Reset	● Set by instruction

LRELn[Notes 2, 3]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. ● After a stop condition is detected, restart is in master mode. ● An address match or extension code reception occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
● Automatically cleared after execution ● Reset	● Set by instruction

WRELn[Notes 2, 3]	Clock stretch cancellation
0	Do not cancel clock stretch
1	Cancel clock stretch. This setting is automatically cleared after clock stretch is canceled.
When the WRELn bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
● Automatically cleared after execution ● Reset	● Set by instruction

Notes 1. 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.

2. The signal of this bit is invalid while IICEn is 0.

3. When the LRELn and WRELn bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICEn = 1).

Remark n = 0

Figure 21-7. Format of IICA Control Register n0 (IICCTLn0) (2/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

SPIEn[Note 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.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
● Cleared by instruction ● Reset	● Set by instruction

WTIMn[Note 1]	Control of clock stretch and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of eight clocks, the clock is set to low level and clock stretch is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of nine clocks, the clock is set to low level and clock stretch is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a clock stretch is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a clock stretch is inserted at the falling edge of the ninth clock after an acknowledge (ACK) is issued. However, when the slave device has received an extension code, a clock stretch is inserted at the falling edge of the eighth clock.	
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

Notes 1. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Remark n = 0

Figure 21-7. Format of IICA Control Register n0 (IICCTLn0) (3/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

STTn[Notes 1, 2]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSYn = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: ● When communication reservation function is enabled (IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. ● When communication reservation function is disabled (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated. In the clock stretch state (when master device): Generates a restart condition after releasing the clock stretch.

Cautions concerning set timing
● For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception. ● For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock. ● Cannot be set to 1 at the same time as stop condition trigger (SPTn). ● Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.

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

Notes 1. The signal of this bit is invalid while IICEn is 0.

2. The STTn bit is always read as 0.

Remarks 1. IICRSVn: Bit 0 of IIC flag register n (IICFn)
STCFn: Bit 7 of IIC flag register n (IICFn)

2. n = 0

Figure 21-7. Format of IICA Control Register n0 (IICCTLn0) (4/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

<table>
<tr><th>SPTn[Note]</th><th colspan="2">Stop condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device’s transfer).</td></tr>
<tr><td colspan="3">Cautions concerning set timing
● For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in 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 clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clocks, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
● Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.</td></tr>
<tr><td colspan="2">Condition for clearing (SPTn = 0)</td><td>Condition for setting (SPTn = 1)</td></tr>
<tr><td colspan="2">● Cleared by loss in arbitration
● Automatically cleared after stop condition is detected
● Cleared by LRELn = 1 (exit from communications)
● When IICEn = 0 (operation stop)
● Reset</td><td>● Set by instruction</td></tr>
</table>

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 status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

Remark n = 0

21.3.4 IICA status register n (IICSn)

This register indicates the status of I^2C.

The IICSn register is read by a 1-bit or 8-bit memory manipulation instruction only when STTn = 1 and during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICSn register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn interrupt request, the change in status 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 IICSn register after the interrupt has been detected.

Remark STTn: bit 1 of IICA control register n0 (IICCTLn0)
WUPn: bit 7 of IICA control register n1 (IICCTLn1)

Figure 21-8. Format of IICA Status Register n (IICSn) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status
Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
● When a stop condition is detected ● When ALDn = 1 (arbitration loss) ● Cleared by LRELn = 1 (exit from communications) ● When the IICEn bit changes from 1 to 0 (operation stop) ● Reset	● When a start condition is generated

ALDn	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTSn bit is cleared.
Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
● Automatically cleared after the IICSn register is read[Note] ● When the IICEn bit changes from 1 to 0 (operation stop) ● Reset	● When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICSn register. Therefore, when using the ALDn bit, read the data of this bit before the data of the other bits.

Remarks 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

2. n = 0

Figure 21-8. Format of IICA Status Register n (IICSn) (2/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

<table>
<tr><th>EXCn</th><th colspan="2">Detection of extension code reception</th></tr>
<tr><td>0</td><td colspan="2">Extension code was not received.</td></tr>
<tr><td>1</td><td colspan="2">Extension code was received.</td></tr>
<tr><td colspan="2">Condition for clearing (EXCn = 0)</td><td>Condition for setting (EXCn = 1)</td></tr>
<tr><td colspan="2">● 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</td><td>● When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).</td></tr>
</table>

<table>
<tr><th>COIn</th><th colspan="2">Detection of matching addresses</th></tr>
<tr><td>0</td><td colspan="2">Addresses do not match.</td></tr>
<tr><td>1</td><td colspan="2">Addresses match.</td></tr>
<tr><td colspan="2">Condition for clearing (COIn = 0)</td><td>Condition for setting (COIn = 1)</td></tr>
<tr><td colspan="2">● 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</td><td>● When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).</td></tr>
</table>

<table>
<tr><th>TRCn</th><th colspan="2">Detection of transmit/receive status</th></tr>
<tr><td>0</td><td colspan="2">Receive status (other than transmit status). The SDAAn line is set for high impedance.</td></tr>
<tr><td>1</td><td colspan="2">Transmit status. 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).</td></tr>
<tr><td colspan="2">Condition for clearing (TRCn = 0)</td><td>Condition for setting (TRCn = 1)</td></tr>
<tr><td colspan="2">&lt;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} (clock stretch cancel)
● When the ALDn bit changes from 0 to 1 (arbitration loss)
● Reset
● When not used for communication (MSTSn, EXCn, 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)</td><td>&lt;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)</td></tr>
</table>

(**Note** and **Remarks** are listed on the next page.)

Note When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

Remarks 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
2. n = 0

Figure 21-8. Format of IICA Status Register n (IICSn) (3/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

ACKDn	Detection of acknowledge (ACK)	
0	Acknowledge was not detected.	
1	Acknowledge was detected.	
Condition for clearing (ACKDn = 0)		Condition for setting (ACKDn = 1)
● When a stop condition is detected ● At the rising edge of the next byte’s first clock ● Cleared by LRELn = 1 (exit from communications) ● When the IICEn bit changes from 1 to 0 (operation stop) ● Reset		● After the SDAAn line is set to low level at the rising edge of SCLAn line’s ninth clock

STDn	Detection of start condition	
0	Start condition was not detected.	
1	Start condition was detected. This indicates that the address transfer period is in effect.	
Condition for clearing (STDn = 0)		Condition for setting (STDn = 1)
● When a stop condition is detected ● At the rising edge of the next byte’s first clock following address transfer ● Cleared by LRELn = 1 (exit from communications) ● When the IICEn bit changes from 1 to 0 (operation stop) ● Reset		● When a start condition is detected

SPDn	Detection of stop condition	
0	Stop condition was not detected.	
1	Stop condition was detected. The master device’s communication is terminated and the bus is released.	
Condition for clearing (SPDn = 0)		Condition for setting (SPDn = 1)
● At the rising edge of the address transfer byte’s first clock following setting of this bit and detection of a start condition ● When the WUPn bit changes from 1 to 0 ● When the IICEn bit changes from 1 to 0 (operation stop) ● Reset		● When a stop condition is detected

Remarks 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)

IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

2. n = 0

21.3.5 IICA flag register n (IICFn)

This register sets the operation mode of I^{2}C and indicates the status of the I^{2}C bus.

The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I^{2}C bus status flag (IICBSYn) bits are read-only.

The IICRSVn bit can be used to enable/disable the communication reservation function.

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 I^{2}C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). When operation is enabled, the IICFn register can be read.

Reset signal generation clears this register to 00H.

Figure 21-9. Format of IICA Flag Register n (IICFn)

Address: FFF52H After reset: 00H R/W[Note]

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)	Condition for setting (STCFn = 1)
● Cleared by STTn = 1 ● When IICEn = 0 (operation stop) ● Reset	● 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 release status (communication initial status when STCENn = 1)
1	Bus communication status (communication initial status when STCENn = 0)
Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
● Detection of stop condition ● When IICEn = 0 (operation stop) ● Reset	● Detection of start condition ● 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)	Condition for setting (STCENn = 1)
● Cleared by instruction ● Detection of start condition ● Reset	● Set by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation
Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
● Cleared by instruction ● Reset	● Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

2. As the bus release status (IICBSYn = 0) is recognized regardless of the actual bus status 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.

3. Write to IICRSVn only when the operation is stopped (IICEn = 0).

Remarks 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

2. n = 0

21.3.6 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I^{2}C and detect the statuses 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 I^{2}C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

Figure 21-10. Format of IICA Control Register n1 (IICCTLn1) (1/2)

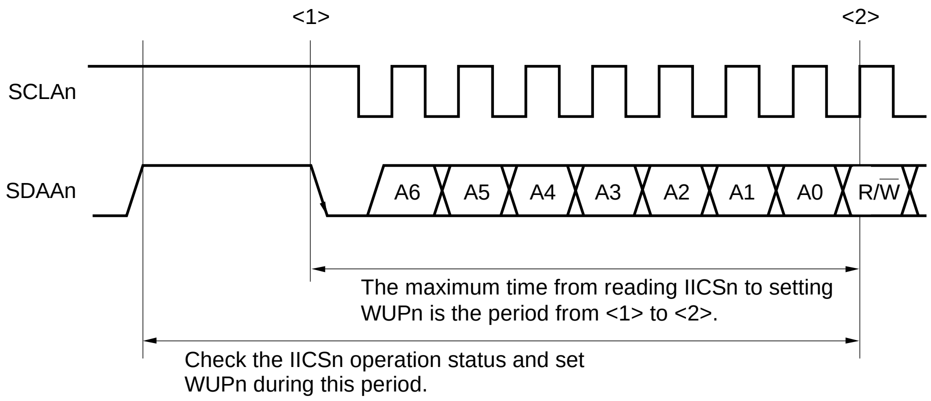
Address: F0231H After reset: 00H R/W[Note 1]

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

<table>
<tr><th>WUPn</th><th colspan="2">Control of address match wakeup</th></tr>
<tr><td>0</td><td colspan="2">Stops operation of address match wakeup function in STOP mode.</td></tr>
<tr><td>1</td><td colspan="2">Enables operation of address match wakeup function in STOP mode.</td></tr>
<tr><td colspan="3">To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three clocks of f_{MCK} after setting (1) the WUPn bit (see <b>Figure 21-23 Flow When Setting WUPn = 1</b>).
Clear (0) the WUPn bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. (The clock stretch must be released and transmit data must be written after the WUPn bit has been cleared (0).)
The interrupt timing when the address has matched 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.</td></tr>
<tr><td colspan="2">Condition for clearing (WUPn = 0)</td><td>Condition for setting (WUPn = 1)</td></tr>
<tr><td colspan="2">●Cleared by instruction (after address match 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>

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



The maximum time from reading IICSn to setting WUPn is the period from <1> to <2>.

Check the IICSn operation status and set WUPn during this period.

Remark n = 0

Figure 21-10. Format of IICA Control Register n1 (IICCTLn1) (2/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
● When the SCLAn pin is at low level ● When IICEn = 0 (operation stop) ● Reset	● When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
● When the SDAAn pin is at low level ● When IICEn = 0 (operation stop) ● Reset	● When the SDAAn pin is at high level

SMCn	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set (1) or cleared (0).	

PRSn	IICA operation clock (f_{MCK}) control
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz)
1	Selects f_{CLK}/2 (20 MHz < f_{CLK})

Cautions 1. The maximum operating frequency of the IICA operating clock (f_{MCK}) is 20 MHz (Max.). Set the IICA control register n1 (IICCTLn1) bit 0 (PRSn) to "1" only when f_{CLK} exceeds 20 MHz.

2. Note the minimum f_{CLK} operation frequency when setting the transfer clock.

The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode:	**f_{CLK} = 3.5 MHz (MIN.)**
Fast mode plus:	**f_{CLK} = 10 MHz (MIN.)**
Normal mode:	**f_{CLK} = 1 MHz (MIN.)**

(**Remarks** are listed on the next page.)

Remarks 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)
2. n = 0

21.3.7 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 I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **21.4.2 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 21-11. Format of IICA Low-Level Width Setting Register n (IICWLn)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

21.3.8 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 I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

Figure 21-12. Format of IICA High-Level Width Setting Register n (IICWHn)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remarks 1. For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **21.4.2 (1)** and **21.4.2 (2)**, respectively.
2. n = 0

21.3.9 Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICEn bit (bit 7 of IICA control register n0 (IICCTLn0)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICEn bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 21-13. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	PM62	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0 to 2)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

21.4 I²C Bus Mode Functions

21.4.1 Pin configuration

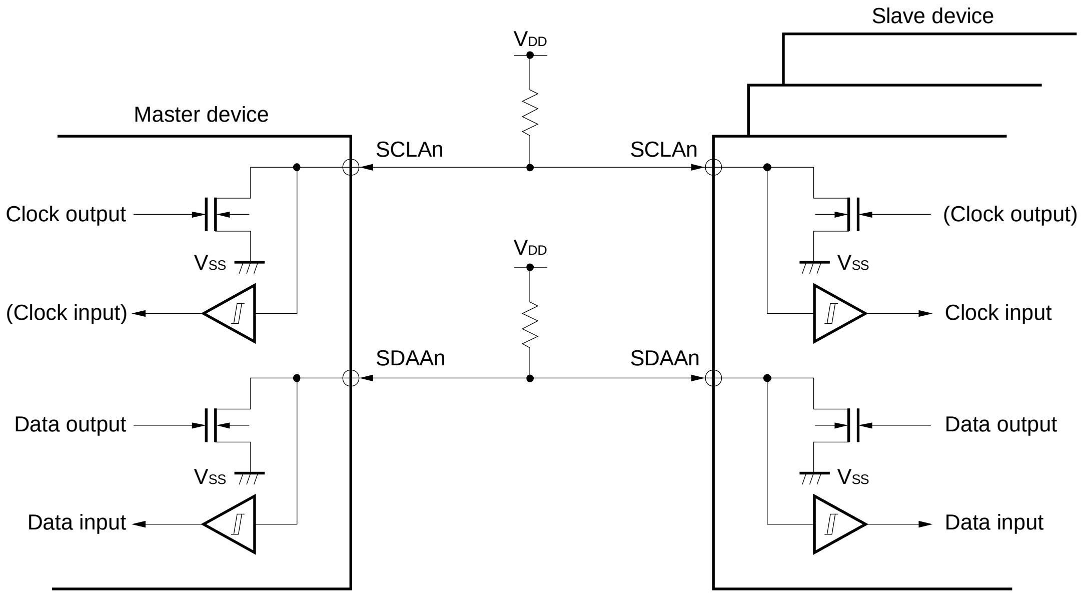
The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

(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 21-14. Pin Configuration Diagram



Remark n = 0

21.4.2 Setting transfer clock by using IICWLn and IICWHn registers

(1) Setting transfer clock on master side

$$\boxed{\text{Transfer clock} = \frac{f_{MCK}}{\text{IICWL0} + \text{IICWH0} + f_{MCK}\,(t_R + t_F)}}$$

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

$$\text{IICWLn} = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWHn} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$\text{IICWLn} = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWHn} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$\text{IICWLn} = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWHn} = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWLn and IICWHn registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$\text{IICWLn} = 1.3\ \mu\text{s} \times f_{MCK}$$
$$\text{IICWHn} = (1.2\ \mu\text{s} - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$\text{IICWLn} = 4.7\ \mu\text{s} \times f_{MCK}$$
$$\text{IICWHn} = (5.3\ \mu\text{s} - t_R - t_F) \times f_{MCK}$$

- When the fast mode plus

$$\text{IICWLn} = 0.50\ \mu\text{s} \times f_{MCK}$$
$$\text{IICWHn} = (0.50\ \mu\text{s} - t_R - t_F) \times f_{MCK}$$

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

Cautions 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (Max.).
Set bit 0 (PRSn) of the IICA control register n1 (IICCTLn1) to “1” only when the f_{CLK} exceeds 20 MHz.

2. Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode:	**f_{CLK} = 3.5 MHz (MIN.)**
Fast mode plus:	**f_{CLK} = 10 MHz (MIN.)**
Normal mode:	**f_{CLK} = 1 MHz (MIN.)**

Remarks 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 resistance and wire load.

2.

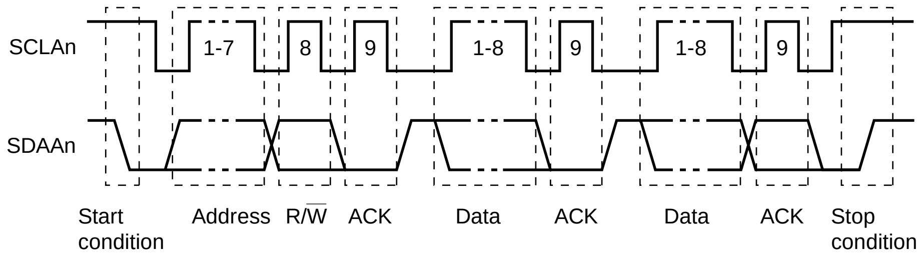
IICWLn:	IICA low-level width setting register n
IICWHn:	IICA high-level width setting register n
t_F:	SDAAn and SCLAn signal falling times
t_R:	SDAAn and SCLAn signal rising times
f_{MCK}:	IICA operation clock frequency

3. n = 0

21.5 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 21-15** shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 21-15. I²C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

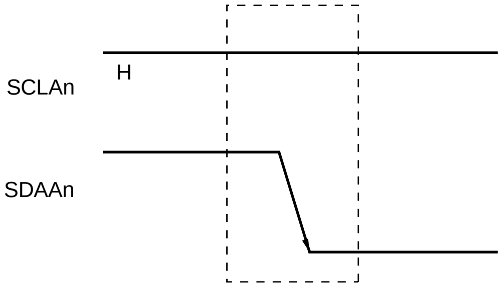
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a clock stretch can be inserted.

21.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn pin are signals 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 21-16. Start Conditions



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

Remark n = 0

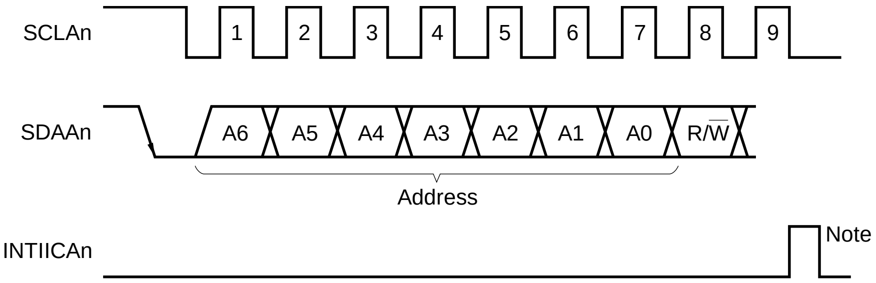
21.5.2 Addresses

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 21-17. Address



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **21.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

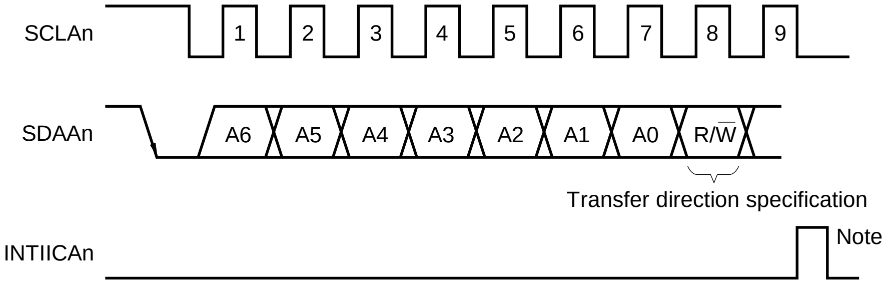
The slave address is assigned to the higher 7 bits of the IICAn register.

21.5.3 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 21-18. Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Remark n = 0

21.5.4 Acknowledge (ACK)

ACK is used to check the status 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.

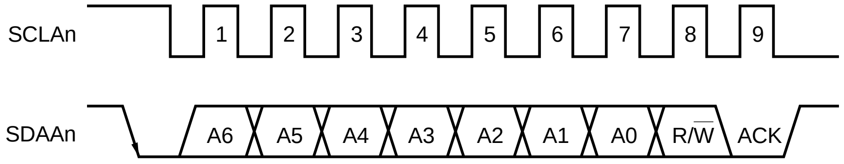
To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).

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 by the data 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 21-19. 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, ACK is generated if the ACKEn bit is set to 1 in advance.

How ACK is generated when data is received differs as follows depending on the setting of the clock stretch timing.

- When 8-clock clock stretching state is selected (bit 3 (WTIMn) of IICCTLn0 register = 0):
  By setting the ACKEn bit to 1 before releasing the clock stretch state, ACK is generated at the falling edge of the eighth clock of the SCLAn pin.
- When 9-clock clock stretch state is selected (bit 3 (WTIMn) of IICCTLn0 register = 1):
  ACK is generated by setting the ACKEn bit to 1 in advance.

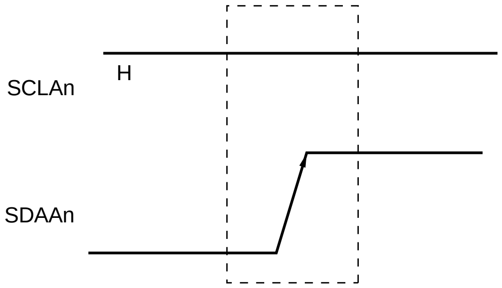
Remark n = 0

21.5.5 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 21-20. 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

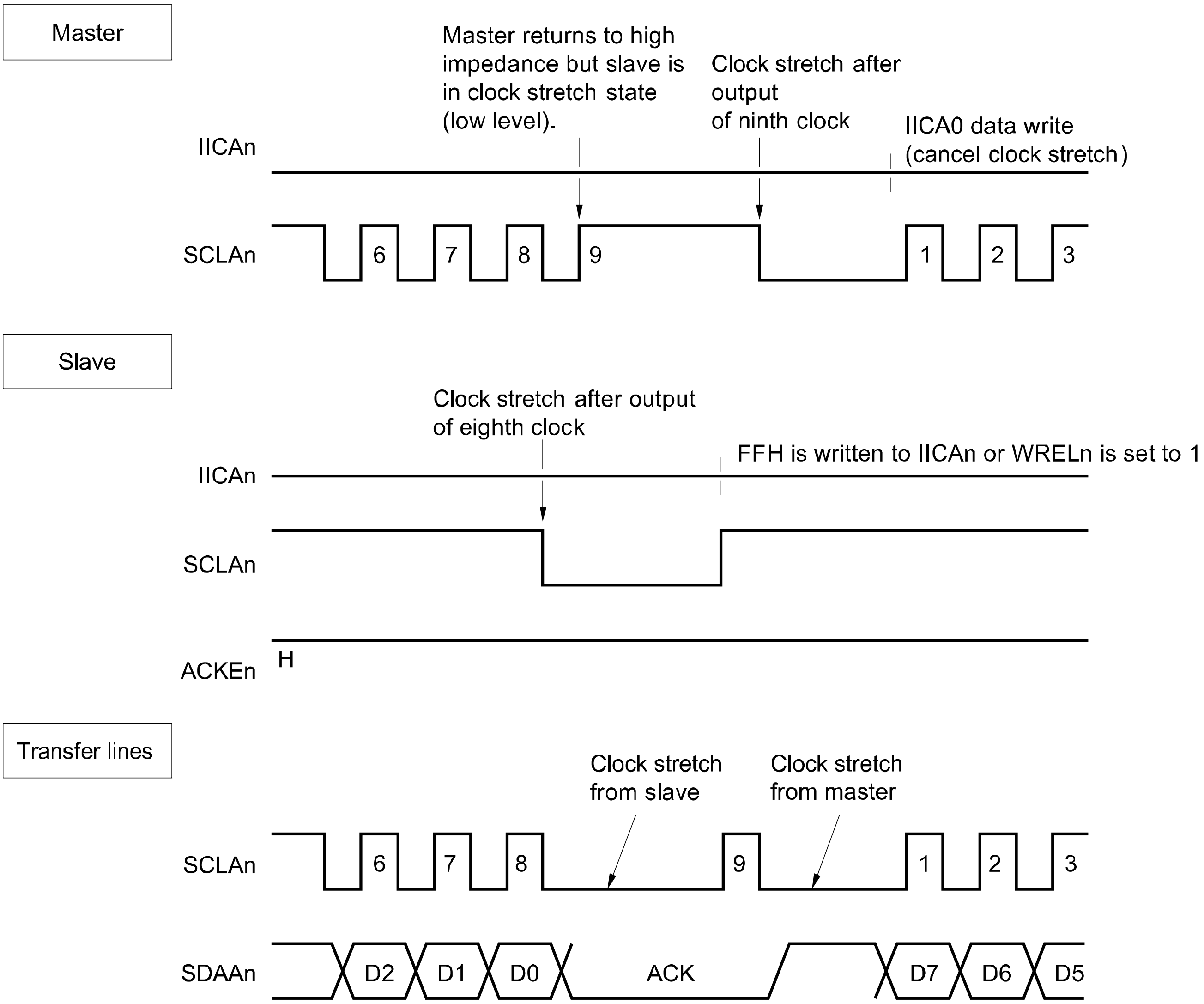
21.5.6 Clock stretching

The clock stretching is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a clock stretch state).

Setting the SCLAn pin to low level notifies the communication partner of the clock stretch state. When clock stretching state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 21-21. Clock stretching (1/2)

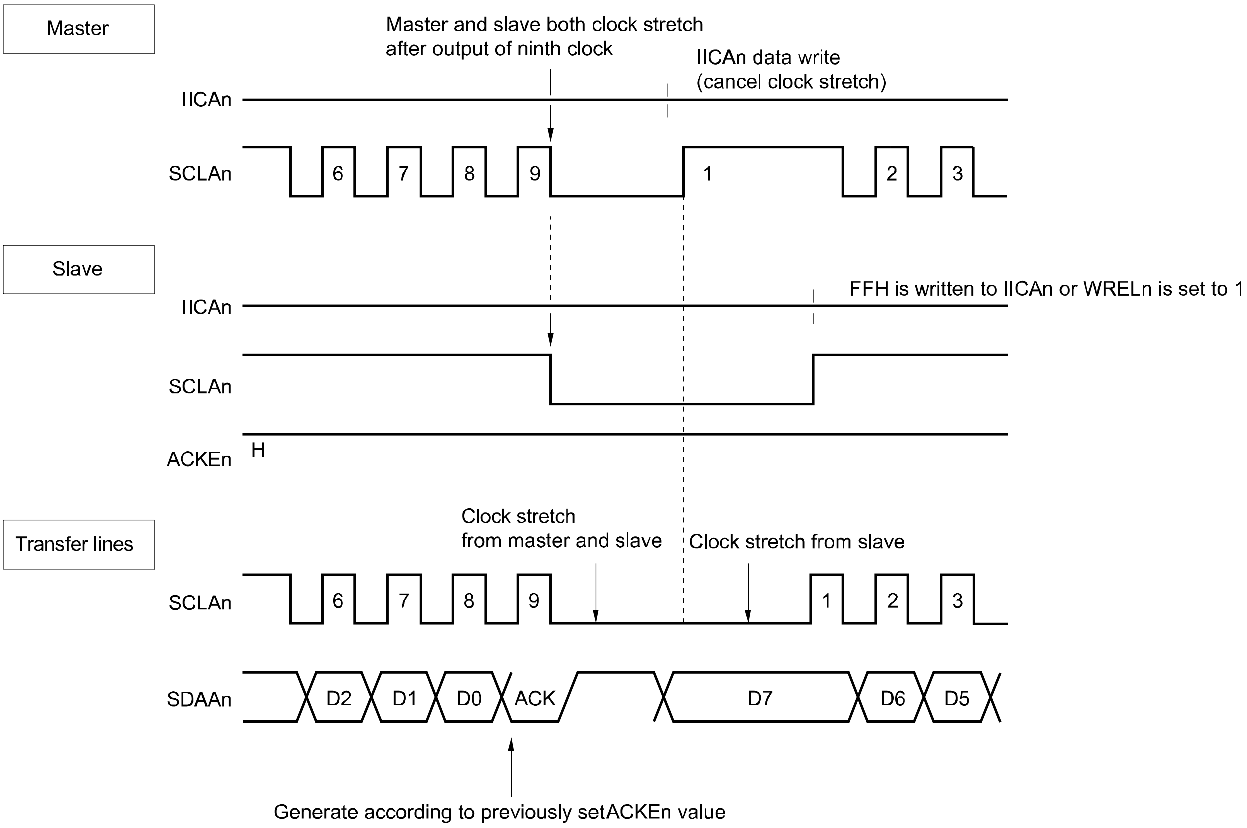
(1) When master device has a nine-clock clock stretching and slave device has an eight-clock clock stretching (master transmits, slave receives, and ACKEn = 1)


Master
Master returns to high impedance but slave is in clock stretch state (low level).
Clock stretch after output of ninth clock
IICA0 data write (cancel clock stretch)
IICAn
SCLAn
6
7
8
9
1
2
3
Slave
Clock stretch after output of eighth clock
FFH is written to IICAn or WRELn is set to 1
IICAn
SCLAn
ACKEn
H
Transfer lines
Clock stretch from slave
Clock stretch from master
SCLAn
6
7
8
9
1
2
3
SDAAn
D2
D1
D0
ACK
D7
D6
D5

Remark n = 0

Figure 21-21. Clock stretching (2/2)

(2) When master and slave devices both have a nine-clock Clock stretching (master transmits, slave receives, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

A clock stretch may be automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0).

Normally, the receiving side cancels 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 cancels the clock stretch state when data is written to the IICAn register.

The master device can also cancel 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

21.5.7 Canceling clock stretch

The I^{2}C usually cancels a 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) (canceling clock stretch)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition)Note
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition)Note

Note Master only

When the above clock stretch canceling processing is executed, the I^{2}C cancels the clock stretch state and communication is resumed.

To cancel a clock stretch state and transmit data (including addresses), write the data to the IICAn register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.

To generate a restart condition after canceling a clock stretch state, set bit 1 (STTn) of the IICCTLn0 register to 1.

To generate a stop condition after canceling a clock stretch state, set bit n (SPTn) of the IICCTLn0 register to 1.

Execute the canceling processing only once for one clock stretch state.

If, for example, data is written to the IICAn register after canceling a 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, communication is stopped if the IICEn bit is cleared to 0 when communication has been aborted, so that the clock stretch state can be canceled.

If the I^{2}C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LRELn) of the IICCTLn0 register, so that the clock stretch state can be canceled.

Caution **If a processing to cancel a clock stretch state is executed when WUPn = 1, the clock stretch state will not be canceled.**

Remark n = 0

21.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and the corresponding clock stretch control, as shown in **Table 21-2**.

Table 21-2. INTIICAn Generation Timing and Clock Stretch Control

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9[Notes 1, 2]	8[Note 2]	8[Note 2]	9	8	8
1	9[Notes 1, 2]	9[Note 2]	9[Note 2]	9	9	9

Notes 1. The slave device’s INTIICAn signal and clock stretch period occurs at the falling edge of the ninth clock 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, 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 9th clock, but clock stretch does not occur.

2. If the received address does not match the contents of the slave address register n (SVAn) and extension code is not received, neither INTIICAn nor a clock stretch occurs.

Remark The numbers in the table indicate the number of the serial clock’s clock signals. Interrupt requests and clock stretch control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and clock stretch timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIMn bit.
- Master device operation: Interrupt and clock stretch timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

(3) During data transmission

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

(4) Clock stretch cancellation method

The four clock stretch cancellation methods are as follows.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling clock stretch)
- Setting bit 1 (STTn) of IICCTLn0 register (generating start condition)[Note]
- Setting bit 0 (SPTn) of IICCTLn0 register (generating stop condition)[Note]

Note Master only.

When an 8-clock clock stretching has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined prior to clock stretch cancellation.

(5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

Remark n = 0

21.5.9 Address match detection method

In I^{2}C 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 (INTIICAn) occurs when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, or when an extension code has been received.

21.5.10 Error detection

In I^{2}C bus mode, the status 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

21.5.11 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 (INTIICAn) is issued at the falling edge of the eighth clock. 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 when the SVAn register is set to 11110xx0. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXCn = 1
- Seven bits of data match: 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 while a slave device is operating, 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 21-3. 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)

Remarks 1. See the I^{2}C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.
2. n = 0

21.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

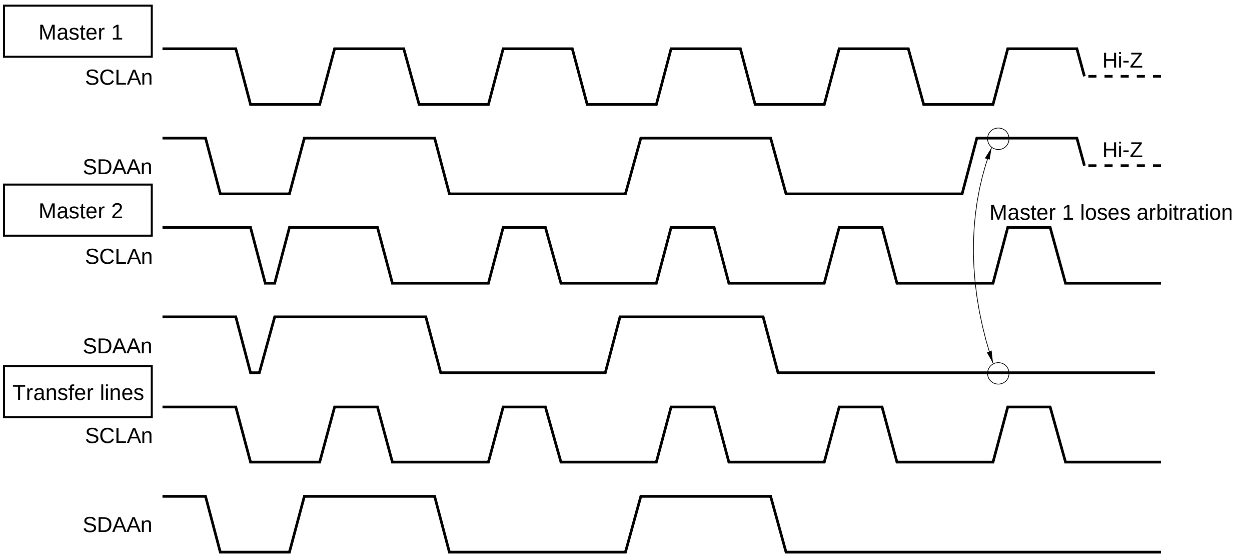
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALDn = 1 setting that has been made by software.

For details of interrupt request timing, see **21.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control**.

Remark STDn: Bit 1 of IICA status register n (IICSn)
STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Figure 21-22. Arbitration Timing Example



Remark n = 0

Table 21-4. Status During Arbitration and Interrupt Request Generation Timing

Status 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	

Notes 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 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

Remarks 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

2. n = 0

21.5.13 Wakeup function

The I^{2}C bus slave function is a function that generates an interrupt request signal (INTIICAn) when a local address and extension code have been received.

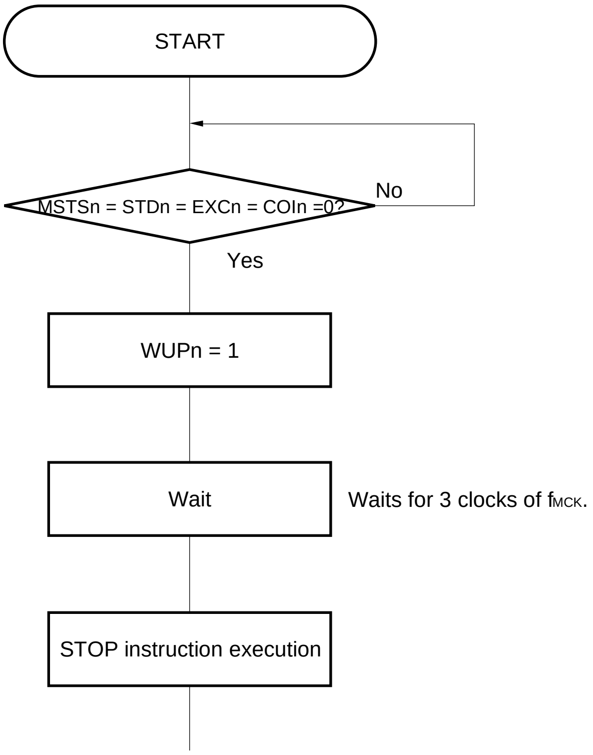
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

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 a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUPn bit after this interrupt has been generated.

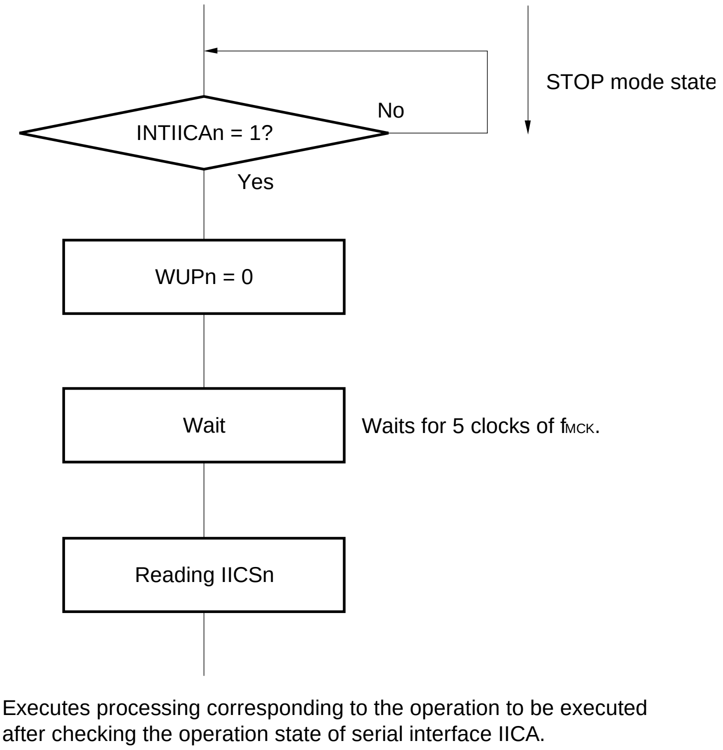
Figure 21-23 shows the flow for setting WUPn = 1 and **Figure 21-24** shows the flow for setting WUPn = 0 upon an address match.

Figure 21-23. Flow When Setting WUPn = 1


START
MSTSn = STDn = EXCn = COIn =0?
No
Yes
WUPn = 1
Wait
Waits for 3 clocks of f_{MCK}.
STOP instruction execution

Remark n = 0

Figure 21-24. Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)


STOP mode state
INTIICAn = 1?
No
Yes
WUPn = 0
Wait
Waits for 5 clocks of fMCK.
Reading IICSn
Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

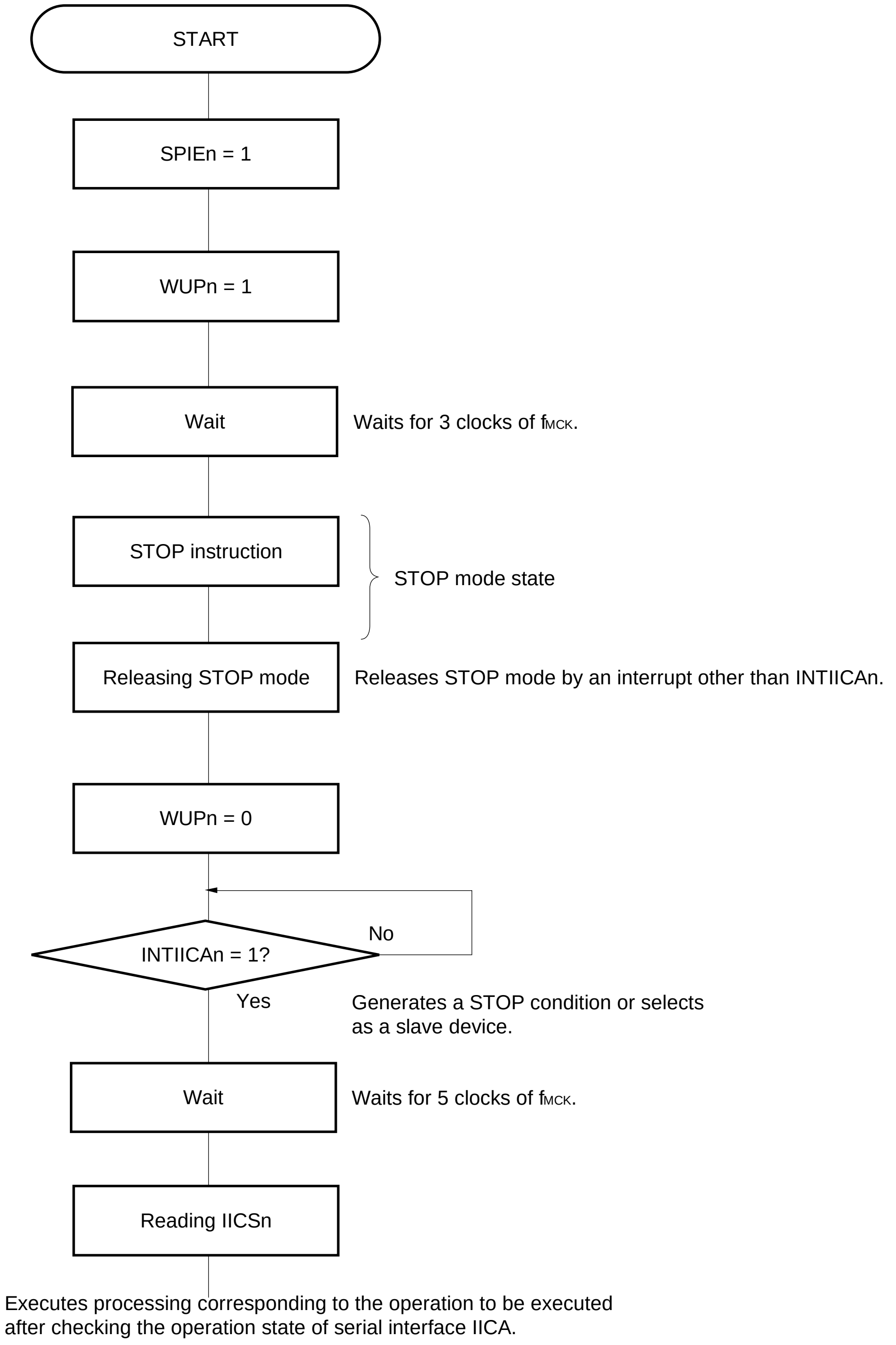
- When operating next IIC communication as master: Flow shown in **Figure 21-25**
- When operating next IIC communication as slave:

  When restored by INTIICAn interrupt: Same as the flow in **Figure 21-23**

  When restored by other than INTIICAn interrupt: Until the INTIICAn interrupt occurs, continue operating with WUPn left set to 1

Remark n = 0

Figure 21-25. When Operating as Master Device After Releasing STOP Mode Other than by INTIICAn


START
SPIEn = 1
WUPn = 1
Wait
Waits for 3 clocks of fMCK.
STOP instruction
STOP mode state
Releasing STOP mode
Releases STOP mode by an interrupt other than INTIICAn.
WUPn = 0
INTIICAn = 1?
No
Yes
Generates a STOP condition or selects as a slave device.
Wait
Waits for 5 clocks of fMCK.
Reading IICSn
Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Remark n = 0

21.5.14 Communication reservation

(1) When communication reservation function is enabled (bit n (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
- 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 saving communication).

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

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 bus status.

- 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.

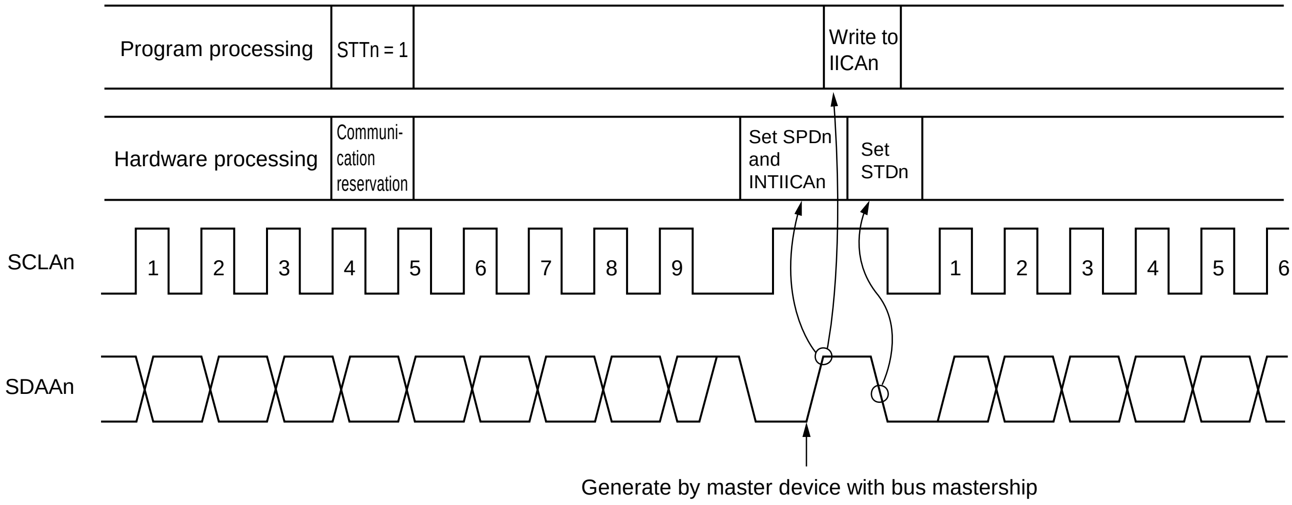
Use software to secure the wait time calculated by the following expression.

Wait time from setting STTn = 1 to checking the MSTSn flag:
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

Remarks	**1.**	IICWLn:	IICA low-level width setting register n
		IICWHn:	IICA high-level width setting register n
		t_F:	SDAAn and SCLAn signal falling times
		f_{MCK}:	IICA operation clock frequency
	2.	n = 0	

Figure 21-26 shows the communication reservation timing.

Figure 21-26. 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 21-27**. 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 21-27. Timing for Accepting Communication Reservations

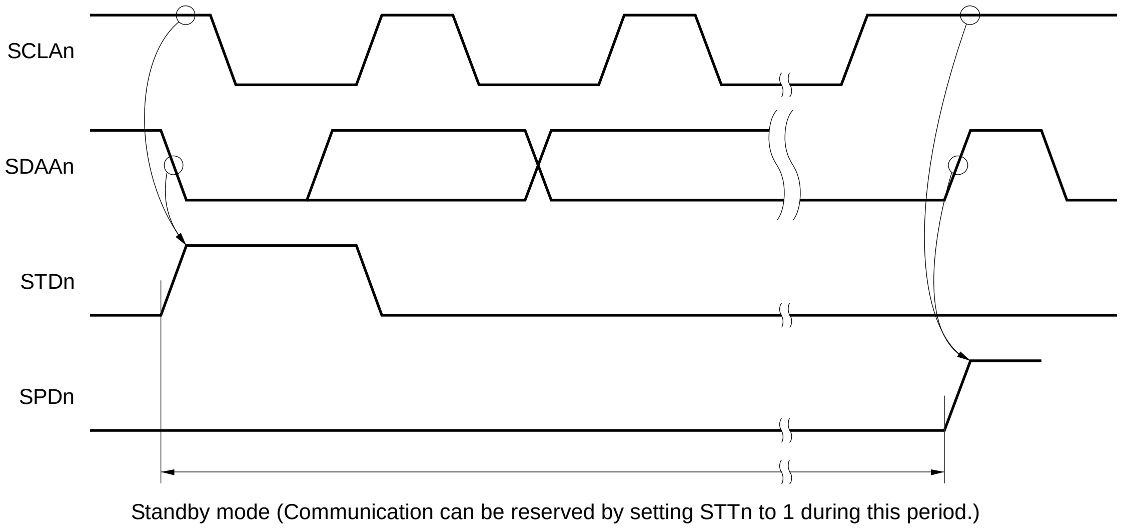
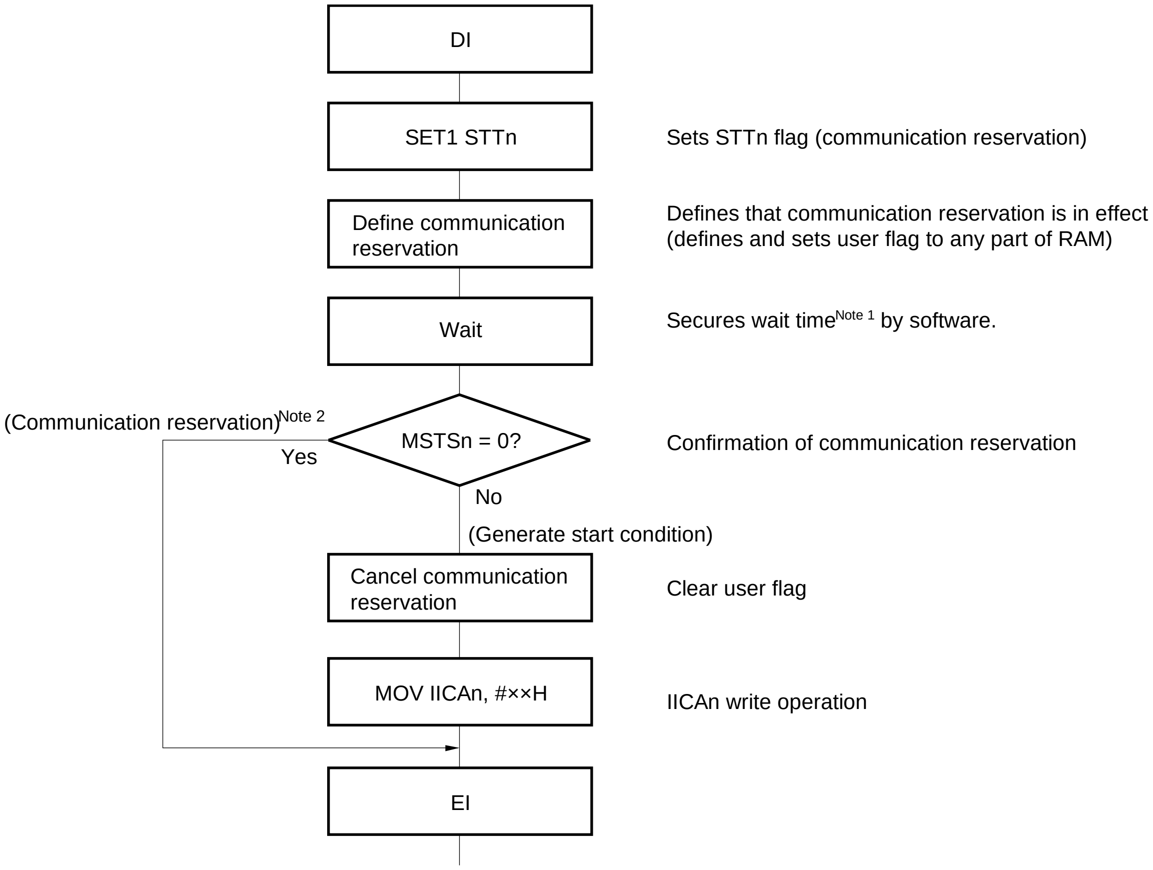

SCLAn
SDAAn
STDn
SPDn
Standby mode (Communication can be reserved by setting STTn to 1 during this period.)

Figure 21-28 shows the communication reservation protocol.

Remark n = 0

Figure 21-28. Communication Reservation Protocol


DI
SET1 STTn
Sets STTn flag (communication reservation)
Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)
Wait
Secures wait time[Note 1] by software.
(Communication reservation)[Note 2]
Yes
MSTSn = 0?
Confirmation of communication reservation
No
(Generate start condition)
Cancel communication reservation
Clear user flag
MOV IICAn, #××H
IICAn write operation
EI

Notes 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remarks 1.

STTn:	Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn:	Bit 7 of IICA status register n (IICSn)
IICAn:	IICA shift register n
IICWLn:	IICA low-level width setting register n
IICWHn:	IICA high-level width setting register n
t_F:	SDAAn and SCLAn signal falling times
f_{MCK}:	IICA operation clock frequency

2. n = 0

(2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- 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 saving 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 clocks of f_{MCK} until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure the time by software.

Remark n = 0

21.5.15 Cautions

(1) When STCENn = 0

Immediately after I^{2}C operation is enabled (IICEn = 1), the bus communication status (IICBSYn = 1) is recognized regardless of the actual bus status. 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 I^{2}C operation is enabled (IICEn = 1), the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status. 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 I^{2}C communications are already in progress

If I^{2}C 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 macro of I^{2}C 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, ACK is returned, but this interferes with other I^{2}C communications. To avoid this, start I^{2}C 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 I^{2}C.
<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 clocks 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

21.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/I1C (512 KB) 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 multimaster system

In the I^{2}C bus multimaster system, whether the bus is released or used cannot be judged by the I^{2}C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/I1C (512 KB) 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/I1C (512 KB) looses 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/I1C (512 KB) is used as the I^{2}C bus slave is shown below.

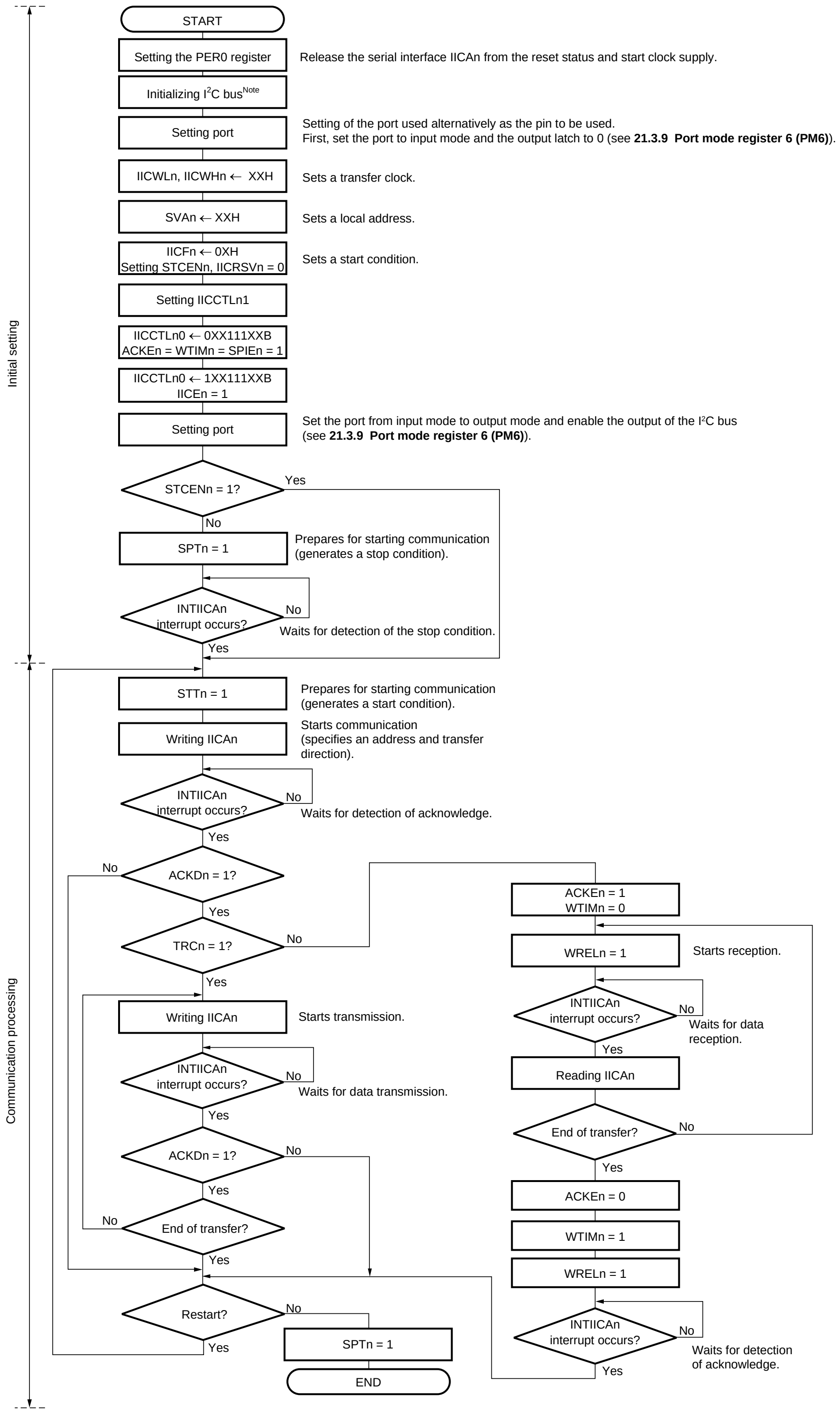
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark n = 0

(1) Master operation in single-master system

Figure 21-29. Master Operation in Single-Master System



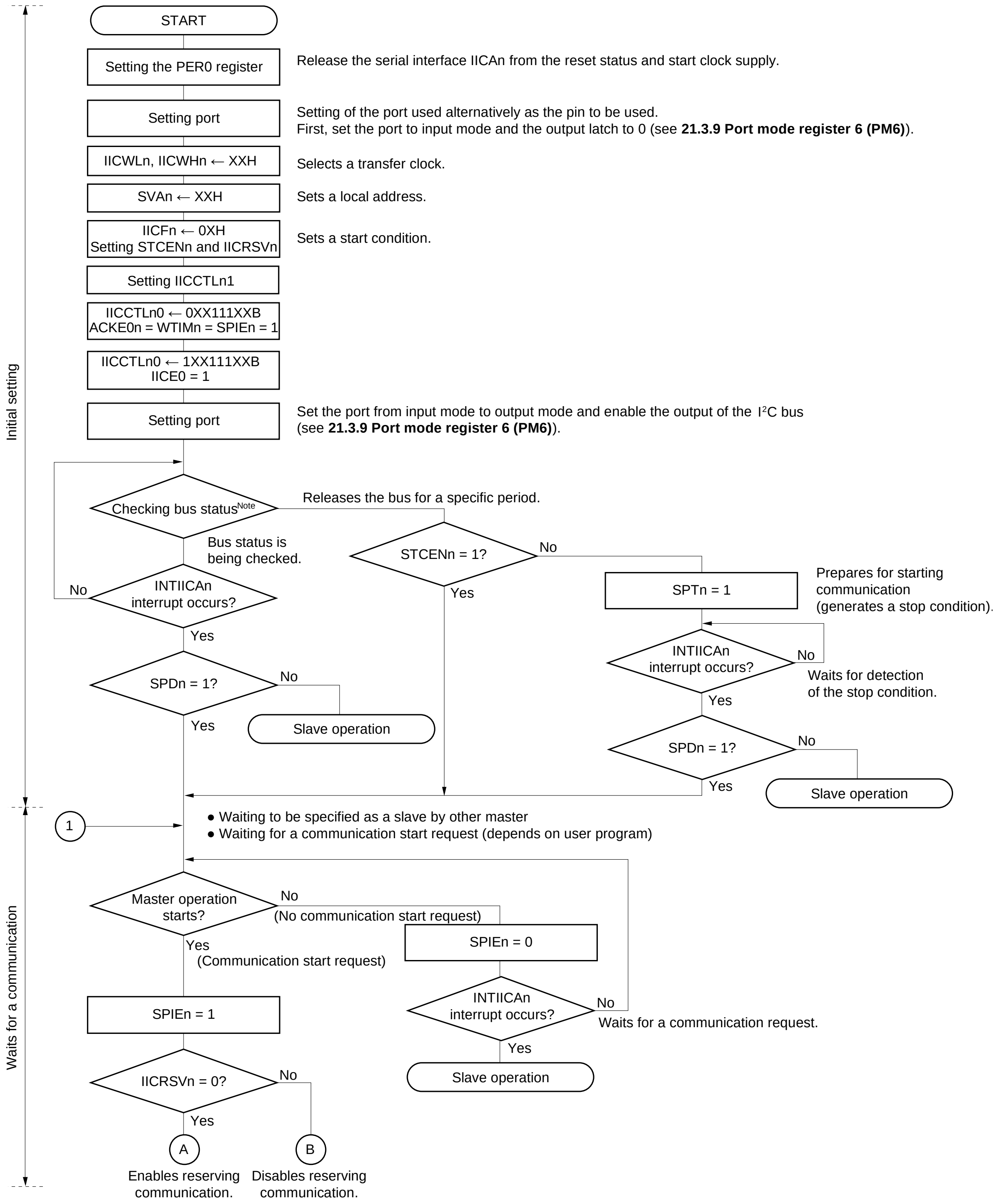
Note Release (SCLAn and SDAAn pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. n = 0

(2) Master operation in multi-master system

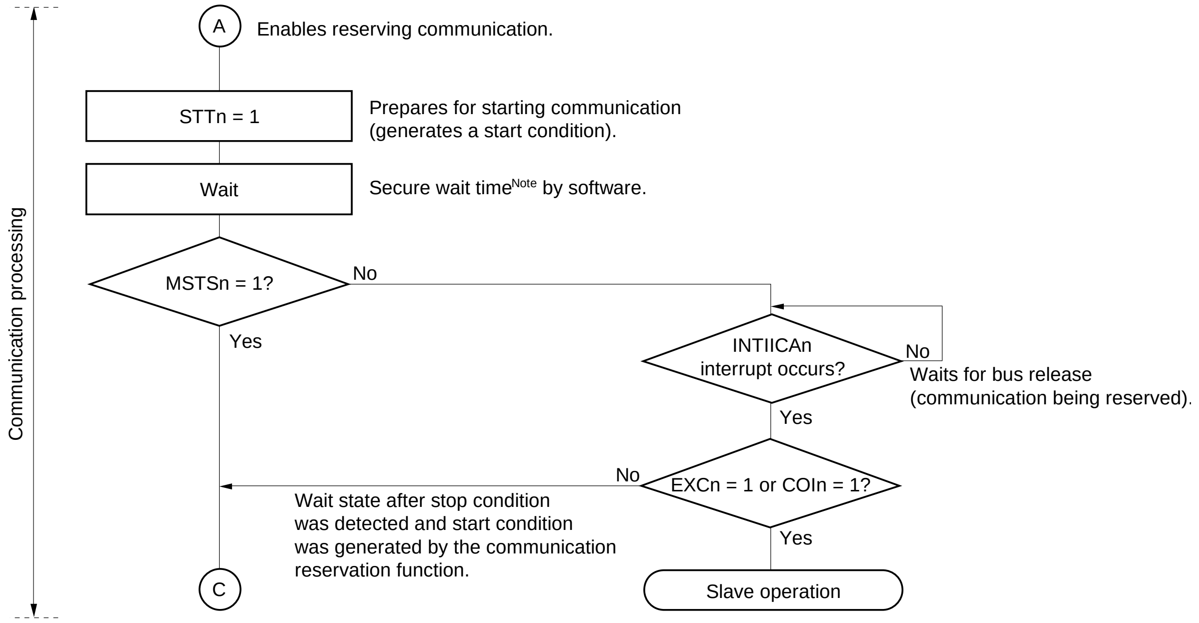
Figure 21-30. Master Operation in Multi-Master System (1/3)


Initial setting
START
Setting the PER0 register
Release the serial interface IICAn from the reset status and start clock supply.
Setting port
Setting of the port used alternatively as the pin to be used.
First, set the port to input mode and the output latch to 0 (see **21.3.9 Port mode register 6 (PM6)**).
IICWLn, IICWHn ← XXH
Selects a transfer clock.
SVAn ← XXH
Sets a local address.
IICFn ← 0XH
Setting STCENn and IICRSVn
Sets a start condition.
Setting IICCTLn1
IICCTLn0 ← 0XX111XXB
ACKE0n = WTIMn = SPIEn = 1
IICCTLn0 ← 1XX111XXB
IICE0 = 1
Setting port
Set the port from input mode to output mode and enable the output of the I²C bus (see **21.3.9 Port mode register 6 (PM6)**).
Checking bus statusNote
Releases the bus for a specific period.
Bus status is being checked.
STCENn = 1?
No
Yes
SPTn = 1
Prepares for starting communication (generates a stop condition).
INTIICAn interrupt occurs?
No
Yes
SPDn = 1?
No
Yes
Slave operation
INTIICAn interrupt occurs?
No
Waits for detection of the stop condition.
Yes
SPDn = 1?
No
Yes
Slave operation
Waits for a communication
1
● Waiting to be specified as a slave by other master
● Waiting for a communication start request (depends on user program)
Master operation starts?
No
(No communication start request)
Yes
(Communication start request)
SPIEn = 0
INTIICAn interrupt occurs?
No
Waits for a communication request.
Yes
Slave operation
SPIEn = 1
IICRSVn = 0?
No
Yes
A
B
Enables reserving communication.
Disables reserving communication.

Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I²C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

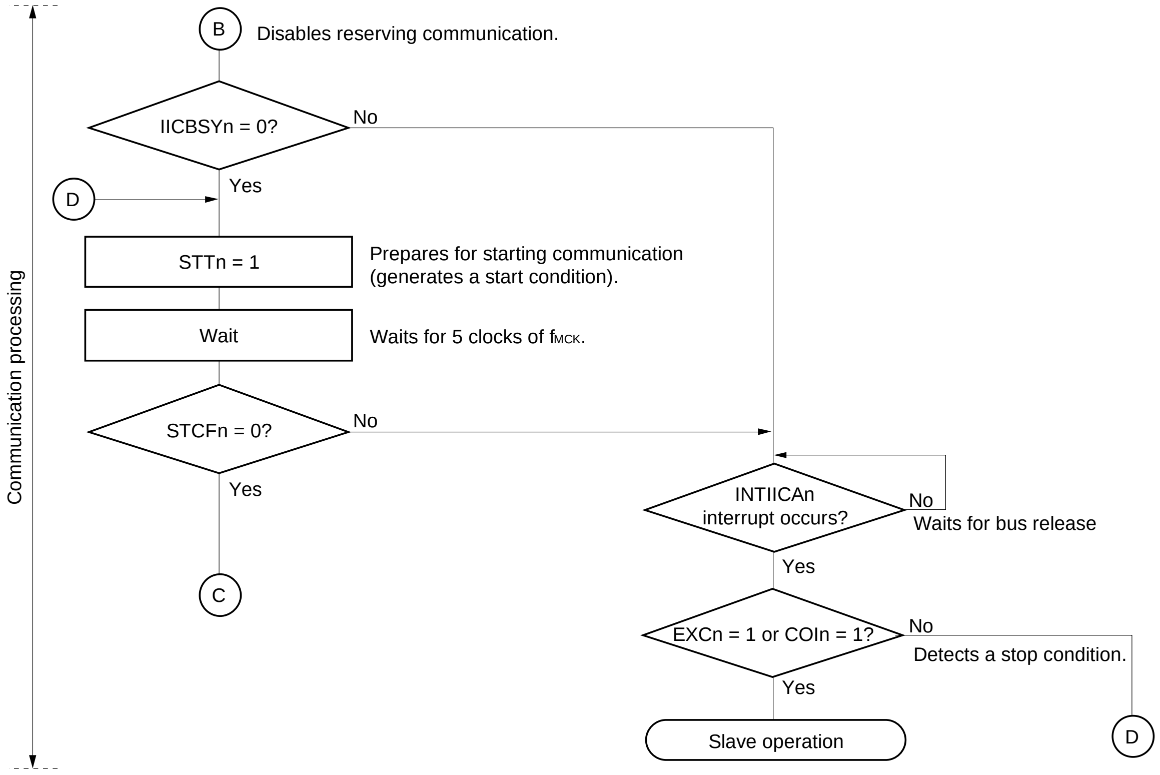
Remark n = 0

Figure 21-30. Master Operation in Multi-Master System (2/3)



Communication processing
A
Enables reserving communication.
STTn = 1
Prepares for starting communication (generates a start condition).
Wait
Secure wait time[Note] by software.
MSTSn = 1?
No
Yes
INTIICAn interrupt occurs?
No
Waits for bus release (communication being reserved).
Yes
EXCn = 1 or COIn = 1?
No
Yes
Wait state after stop condition was detected and start condition was generated by the communication reservation function.
C
Slave operation

Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

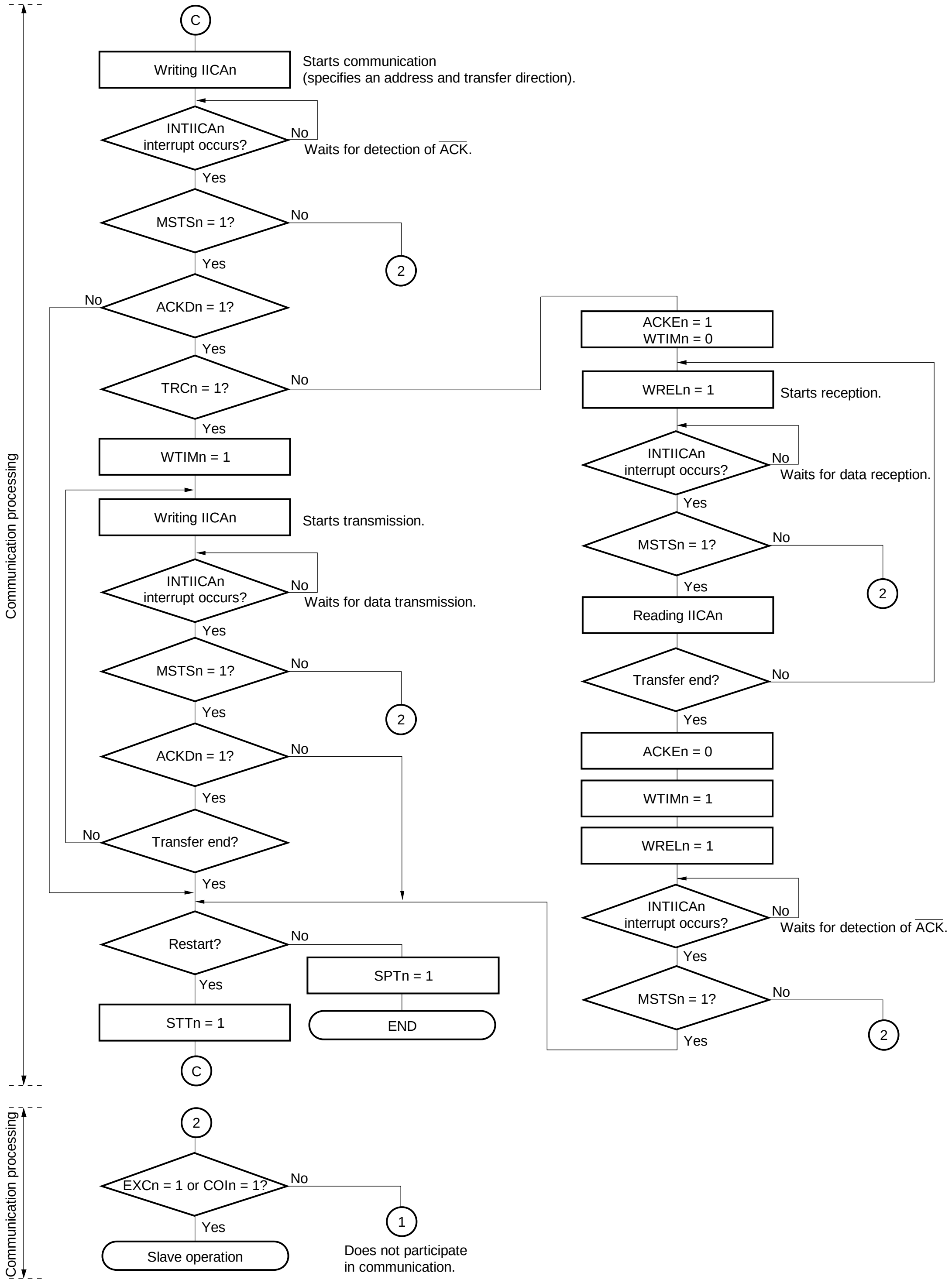


Communication processing
B
Disables reserving communication.
IICBSYn = 0?
No
Yes
D
STTn = 1
Prepares for starting communication (generates a start condition).
Wait
Waits for 5 clocks of fMCK.
STCFn = 0?
No
Yes
C
INTIICAn interrupt occurs?
No
Waits for bus release
Yes
EXCn = 1 or COIn = 1?
No
Detects a stop condition.
Yes
Slave operation
D

Remarks 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

2. n = 0

Figure 21-30. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.

3. To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

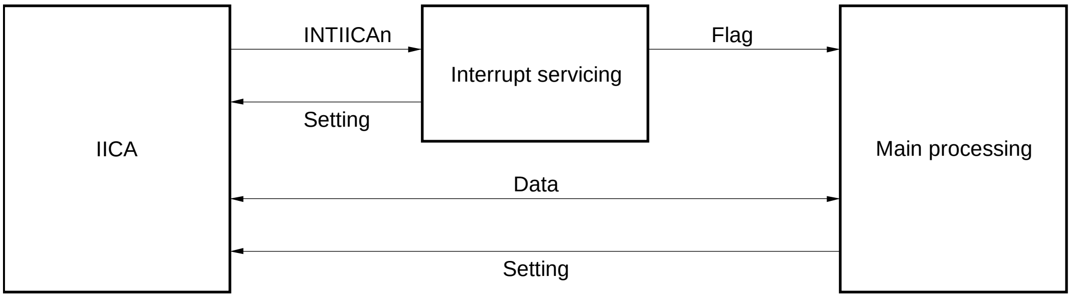
4. n = 0

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

Remark n = 0

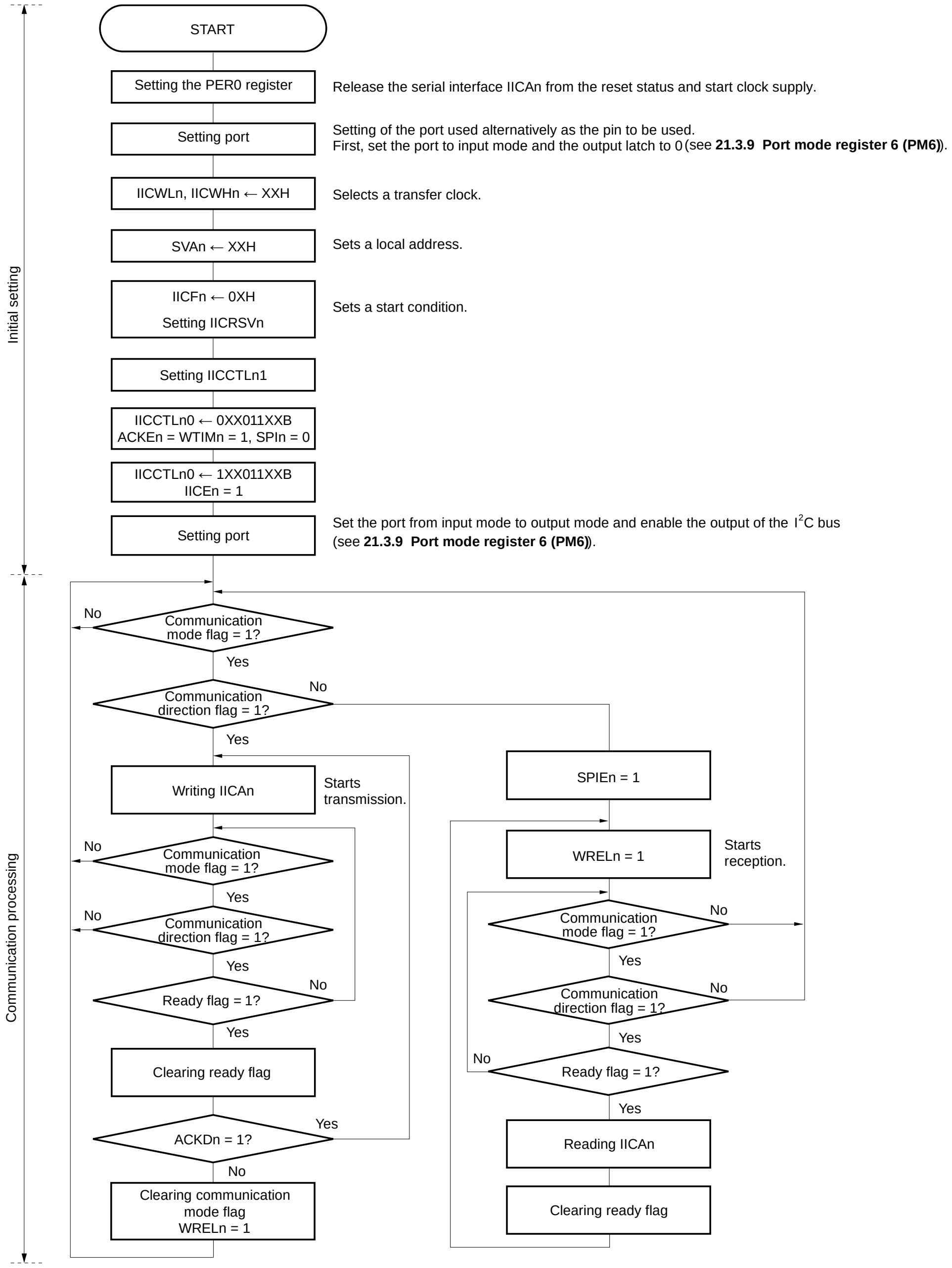
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 21-31. Slave Operation Flowchart (1)


Initial setting
START
Setting the PER0 register
Release the serial interface IICAn from the reset status and start clock supply.
Setting port
Setting of the port used alternatively as the pin to be used.
First, set the port to input mode and the output latch to 0 (see **21.3.9 Port mode register 6 (PM6)**).
IICWLn, IICWHn ← XXH
Selects a transfer clock.
SVAn ← XXH
Sets a local address.
IICFn ← 0XH
Setting IICRSVn
Sets a start condition.
Setting IICCTLn1
IICCTLn0 ← 0XX011XXB
ACKEn = WTIMn = 1, SPIn = 0
IICCTLn0 ← 1XX011XXB
IICEn = 1
Setting port
Set the port from input mode to output mode and enable the output of the I^{2}C bus (see **21.3.9 Port mode register 6 (PM6)**).
Communication processing
Communication mode flag = 1?
No
Yes
Communication direction flag = 1?
No
Yes
Writing IICAn
Starts transmission.
Communication mode flag = 1?
No
Yes
Communication direction flag = 1?
No
Yes
Ready flag = 1?
No
Yes
Clearing ready flag
ACKDn = 1?
Yes
No
Clearing communication mode flag
WRELn = 1
SPIEn = 1
WRELn = 1
Starts reception.
Communication mode flag = 1?
No
Yes
Communication direction flag = 1?
No
Yes
Ready flag = 1?
No
Yes
Reading IICAn
Clearing ready flag

Remarks 1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

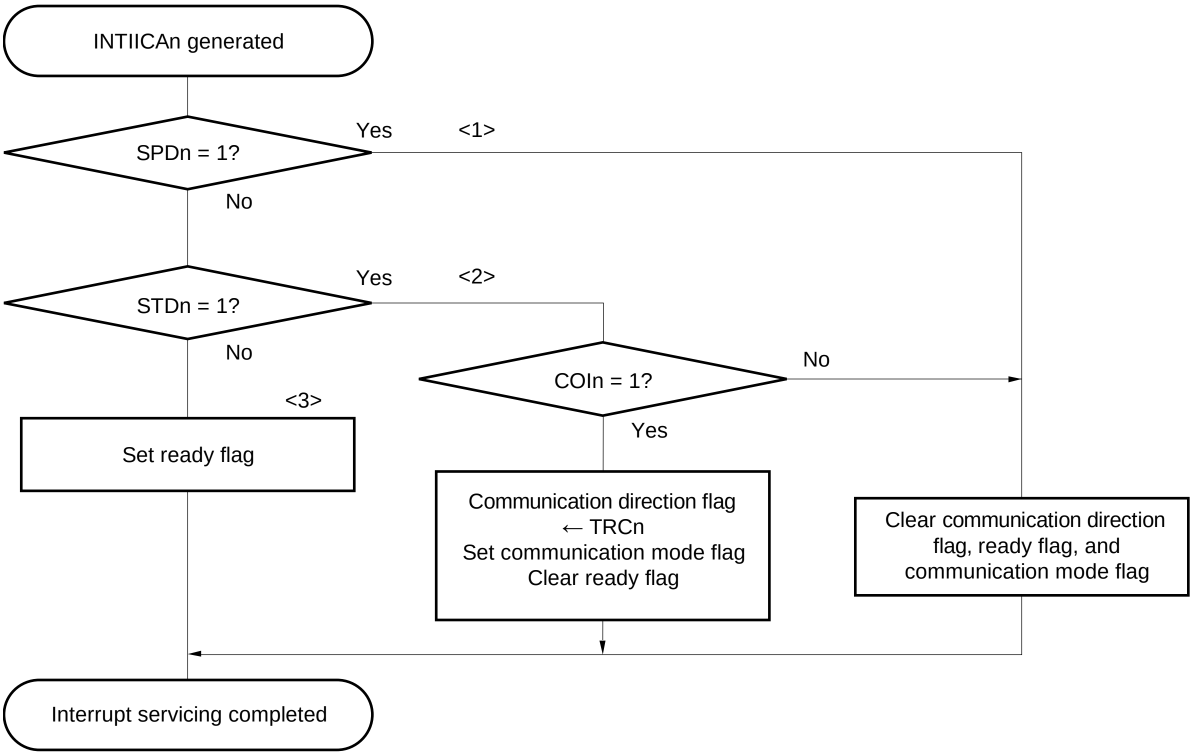
2. n = 0

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn interrupt checks the status, 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 cancelled, 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 I^{2}C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 21-32 Slave Operation Flowchart (2)**.

Figure 21-32. Slave Operation Flowchart (2)


INTIICAn generated
SPDn = 1?
Yes
<1>
No
STDn = 1?
Yes
<2>
No
<3>
COIn = 1?
No
Yes
Set ready flag
Communication direction flag ← TRCn
Set communication mode flag
Clear ready flag
Clear communication direction flag, ready flag, and communication mode flag
Interrupt servicing completed

Remark n = 0

21.5.17 Timing of I^{2}C interrupt request (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.

Remarks 1.

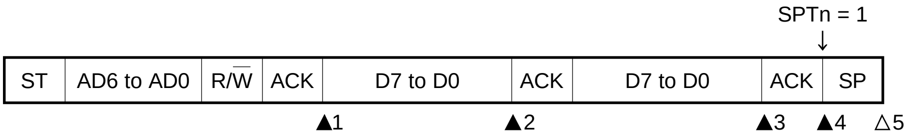
ST:	Start condition
AD6 to AD0:	Address
$R/\overline{W}$:	Transfer direction specification
ACK:	Acknowledge
D7 to D0:	Data
SP:	Stop condition

2. n = 0

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B

▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1)[Note]

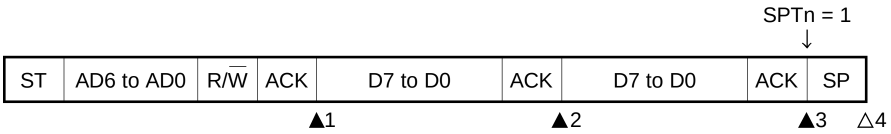
▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△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
△: 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)

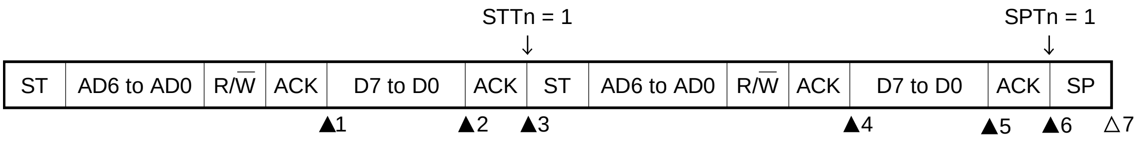
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)[Note 1]

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0[Note 2], sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1)[Note 3]

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

Notes 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

2. Clear the WTIMn bit to 0 to restore the original setting.

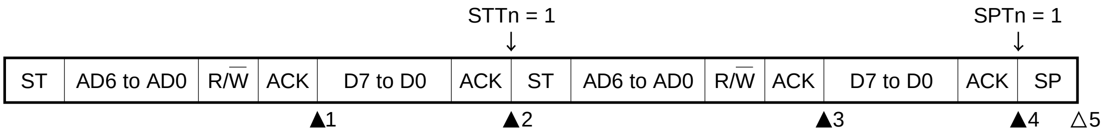
3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: 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)

△5: IICSn = 00000001B

Remark ▲: Always generated

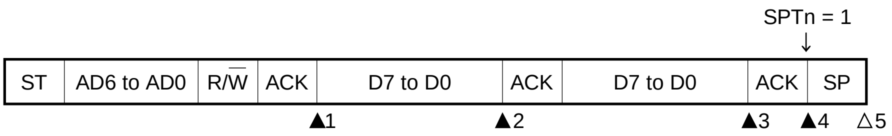
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0

(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

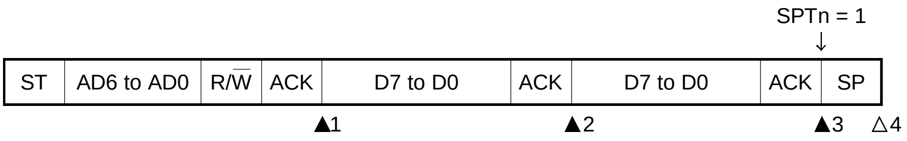
▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△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
△: 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)

△4: IICSn = 00001001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIMn = 0

ST	AD6 to AD0	R/$\overline{\text{W}}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
				▲1	▲2		▲3		△4

▲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

ST	AD6 to AD0	R/$\overline{\text{W}}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
				▲1		▲2		▲3	△4

▲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

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1	▲2					▲3	▲4		△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

(ii) When WTIMn = 1 (after restart, matches with SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001××00B

△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1	▲2				▲3		▲4		△5

▲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))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1		▲2			▲3	▲4		▲5	△6

▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×110B
▲5: IICSn = 0010××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1	▲2					▲3			△4

▲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))

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1		▲2				▲3			△4

▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIMn = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
			▲1		▲2		▲3		△4

▲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

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
			▲1	▲2		▲3		▲4	△5

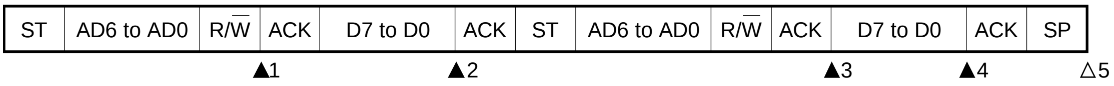
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

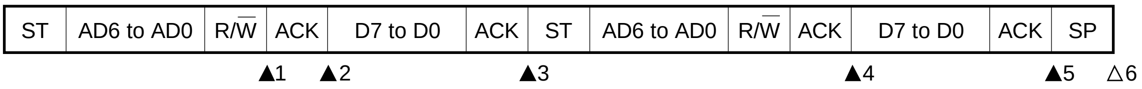
(i) When WTIMn = 0 (after restart, matches SVAn)



▲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 SVAn)



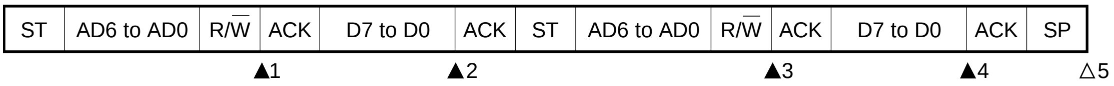
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 0001×110B
▲5: IICSn = 0001××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

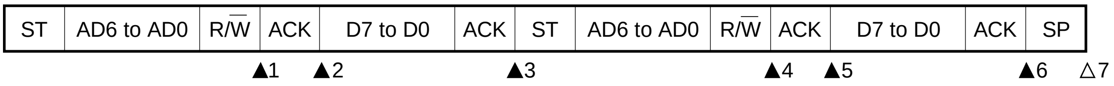
(i) When WTIMn = 0 (after restart, extension code reception)



▲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)



▲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

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	SP
		▲1		▲2					▲3			△4

▲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))

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	SP
		▲1	▲2		▲3				▲4			△5

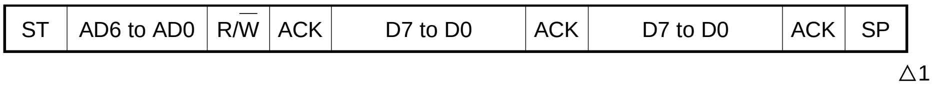
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 00000×10B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

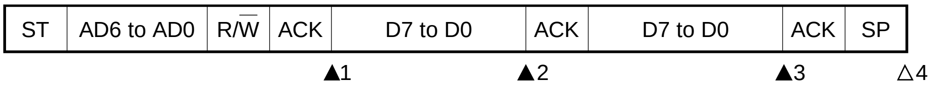
Remark △: Generated only when SPIEn = 1

(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

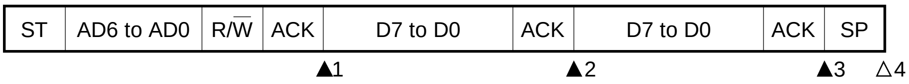


▲1: IICSn = 0101×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(ii) When WTIMn = 1



▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

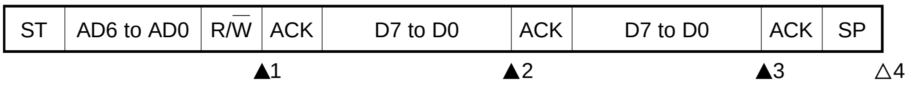
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

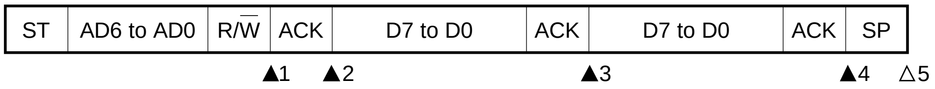
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0

(ii) When WTIMn = 1



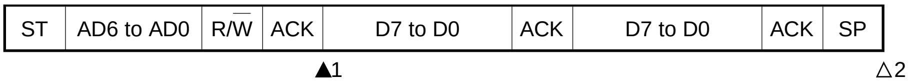
▲1: IICSn = 0110×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)

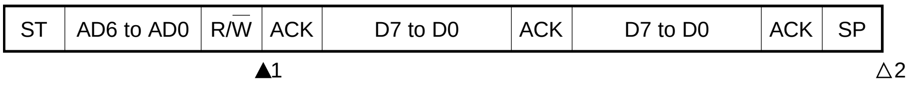


▲1: IICSn = 01000110B
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

Remark n = 0

(b) When arbitration loss occurs during transmission of extension code

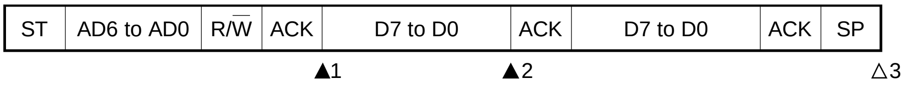


▲1: IICSn = 0110×010B
Sets LRELn = 1 by software
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

(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

Remark n = 0

(ii) When WTIMn = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2			△3

▲1: IICSn = 10001110B
▲2: IICSn = 01000100B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dm	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1					▲2			△3

▲1: IICSn = 1000×110B
▲2: IICSn = 01000110B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care
m = 6 to 0

Remark n = 0

(ii) Extension code

ST	AD6 to AD0	R/$\overline{\text{W}}$	ACK	D7 to Dm	ST	AD6 to AD0	R/$\overline{\text{W}}$	ACK	D7 to D0	ACK	SP	
				▲1				▲2				△3

▲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

ST	AD6 to AD0	R/$\overline{\text{W}}$	ACK	D7 to Dm	SP	
				▲1		△2

▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

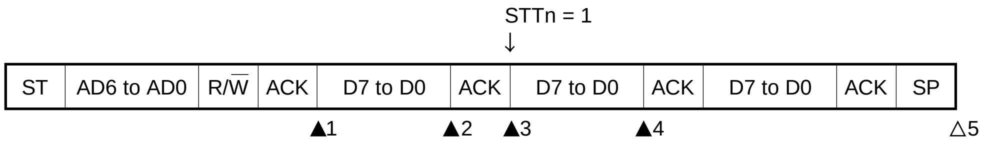
×: Don’t care

m = 6 to 0

Remark n = 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

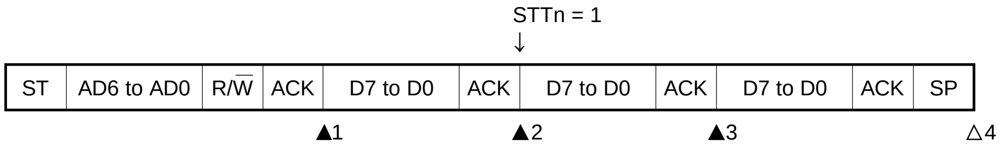
▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000000B

△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the STTn bit to 1)

▲3: IICSn = 01000100B

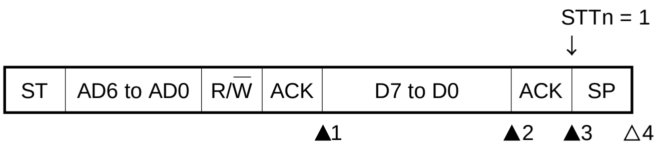
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don’t care

Remark n = 0

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000××00B (Sets the STTn bit to 1)

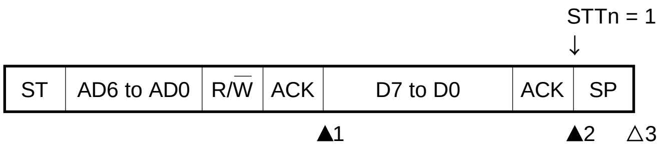
△4: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

△3: IICSn = 01000001B

Remark ▲: Always generated

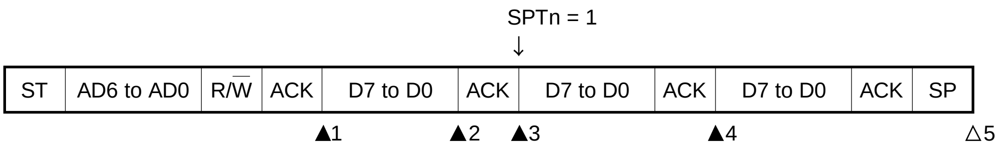
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

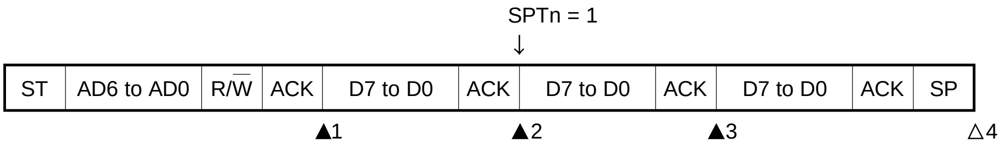
▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲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

21.6 Timing Charts

When using the I^{2}C 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.

Figures 21-33 and **21-34** show timing charts of the data communication.

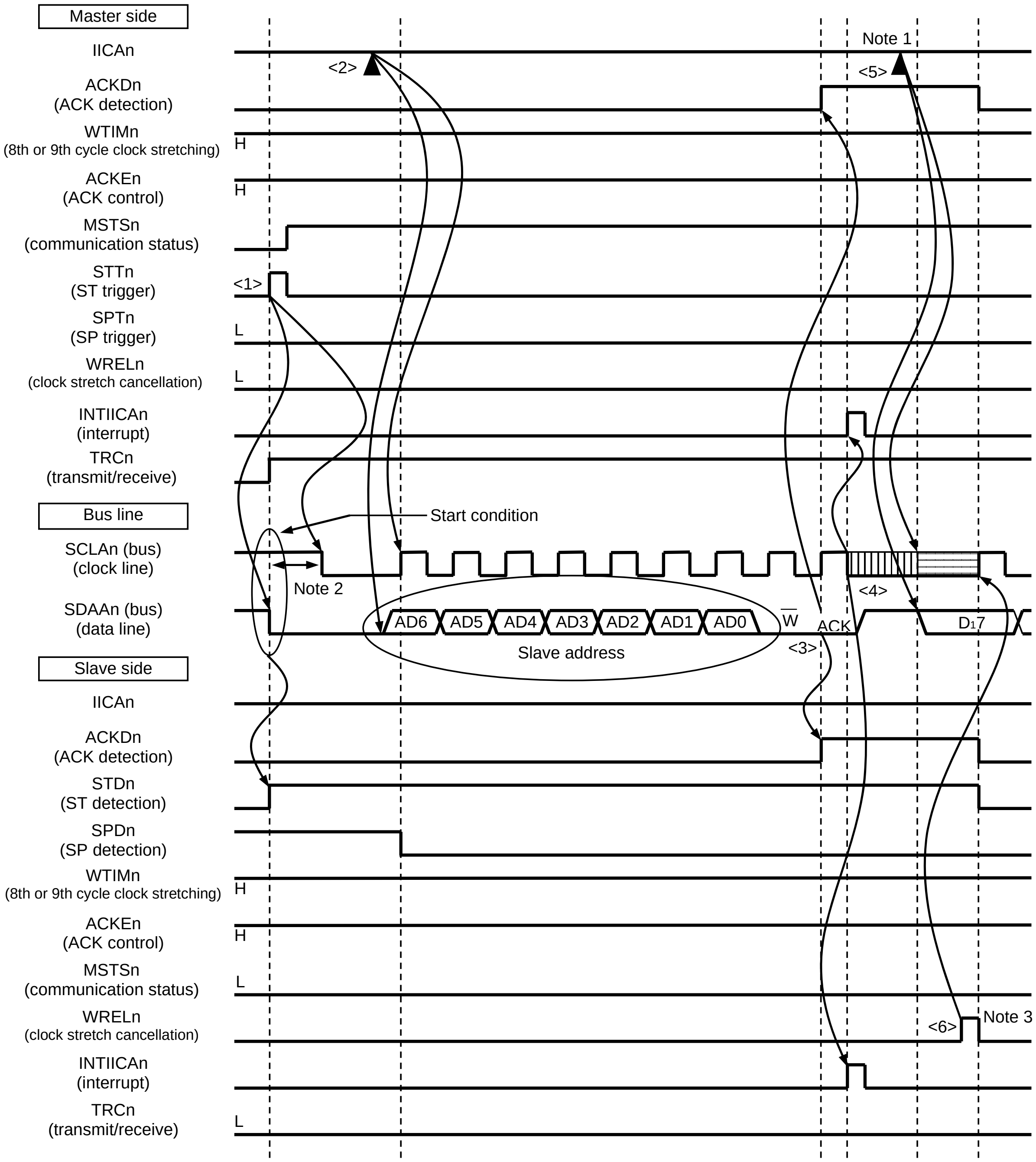
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.

Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark n = 0

Figure 21-33. Example of Master to Slave Communication (9-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

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.

3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The following is a description of the **Figure 21-33 (1) Start condition ~ address ~ data** <1> to <6>.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn 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 communication status (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 whose address matched the transmitted slave address sets a clock stretch status (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 status that it set by the master device.

<6> If the slave device releases the clock stretch status (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 a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

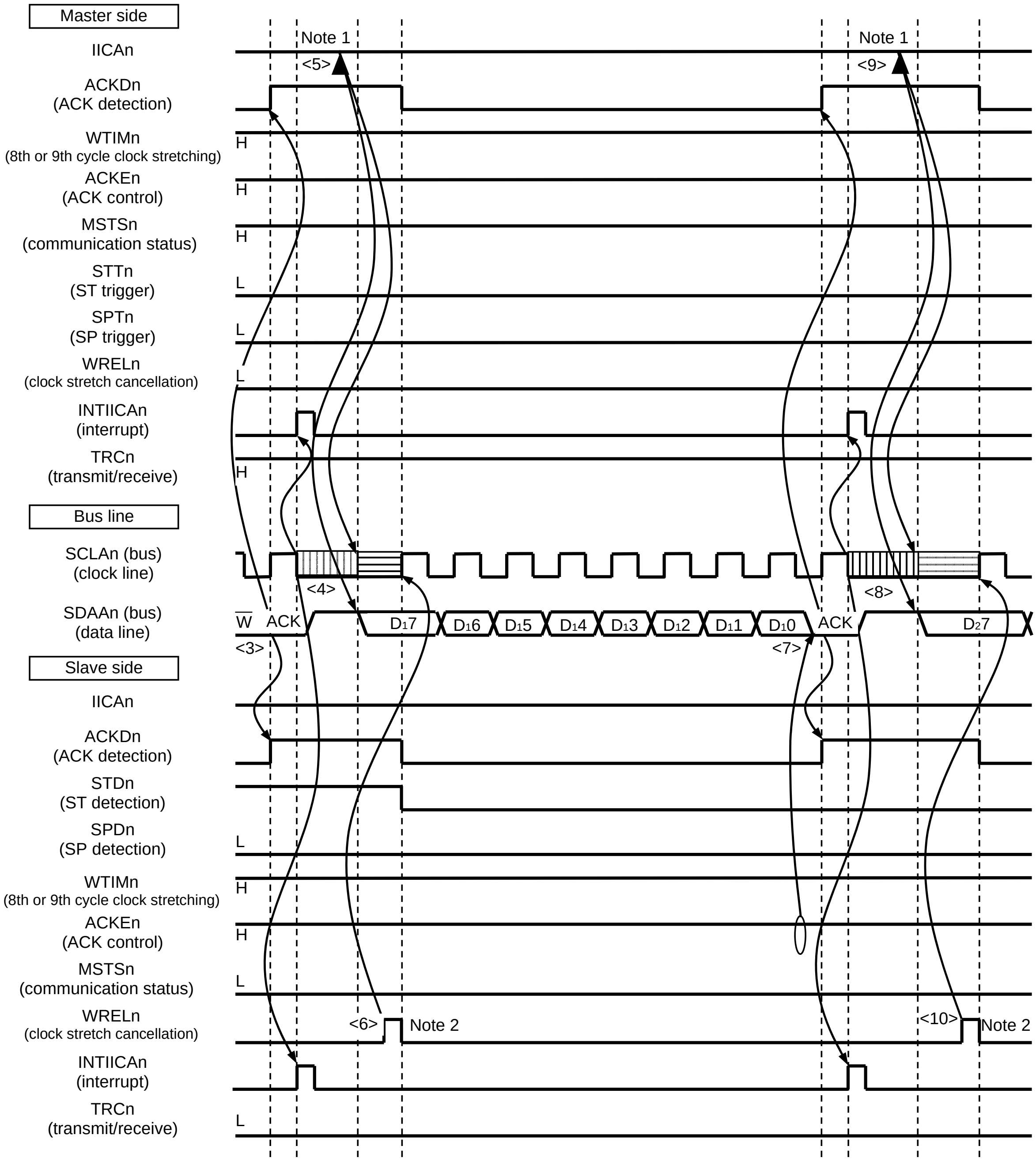
Remarks 1. <1> to <15> in **Figure 21-33** represent the entire procedure for communicating data using the I^{2}C bus.

Figure 21-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, F**igure 21-33 (2) Address ~ data ~ data** shows the processing from <3> to <10>, and **Figure 21-33 (3) Data ~ data ~ stop condition** shows the processing from <7> to <15>.

2. n = 0

Figure 21-33. Example of Master to Slave Communication
(9-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The following is a description of the **Figure 21-33 (2) Address ~ data ~ data** <3> to <10>.

<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 whose address matched the transmitted slave address sets a clock stretch status (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 status that it set by the master device.

<6> If the slave device releases the clock stretch status (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 a clock stretch status (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 status that it set by the master device.

<10> The slave device reads the received data and releases the clock stretch status (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 a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

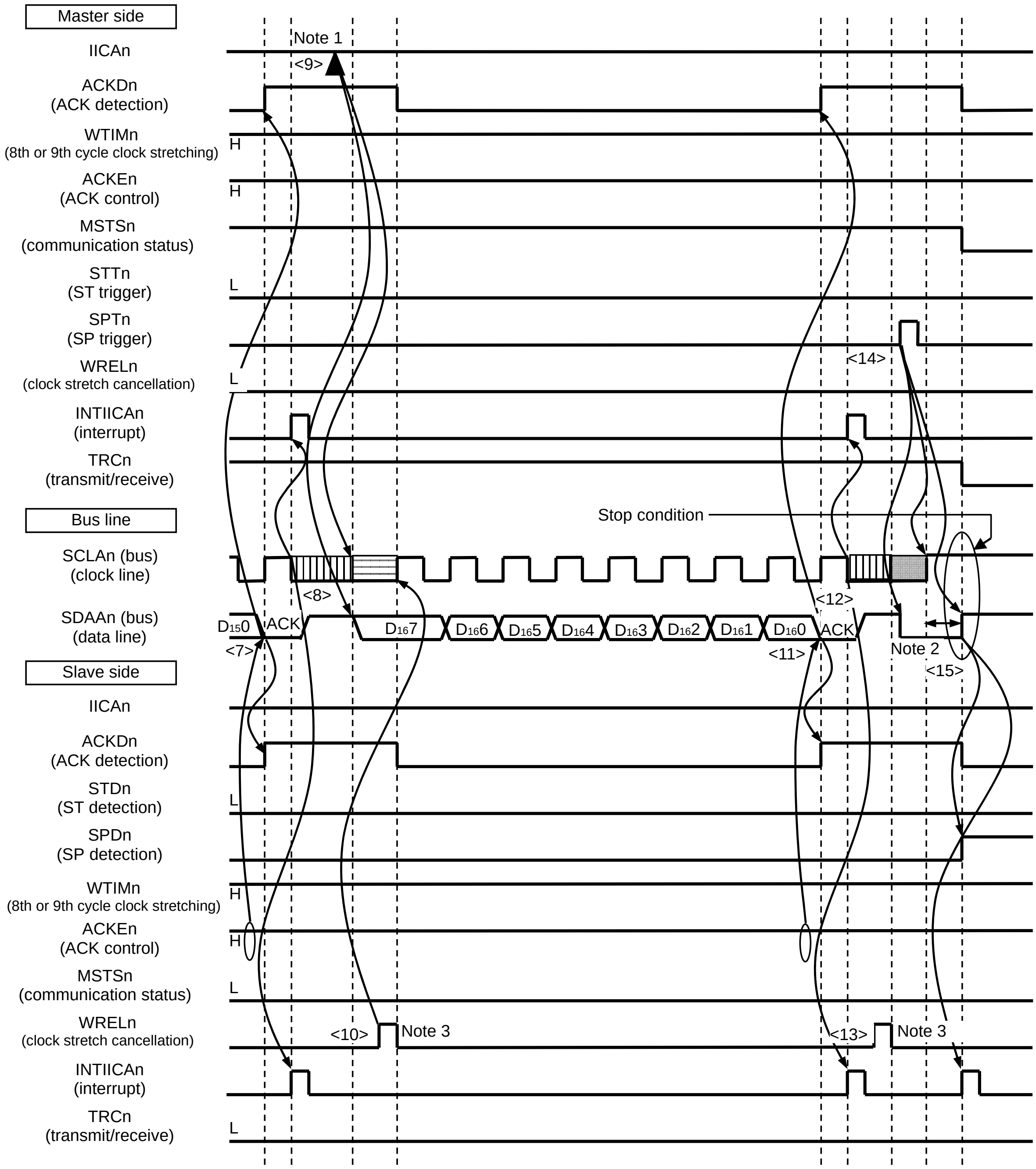
Remarks 1. <1> to <15> in **Figure 21-33** represent the entire procedure for communicating data using the I^{2}C bus.

Figure 21-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, **Figure 21-33 (2) Address ~ data ~ data** shows the processing from <3> to <10>, and **Figure 21-33 (3) Data ~ data ~ stop condition** shows the processing from <7> to <15>.

2. n = 0

Figure 21-33. Example of Master to Slave Communication (9-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



▓ : Clock stretch state by master device

▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The following is a description of the **Figure 21-33 (3) Data ~ data ~ stop condition** <7> to <15>.

<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 a clock stretch status (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 status that it set by the master device.

<10> The slave device reads the received data and releases the clock stretch status (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 a clock stretch status (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 status (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 changes SDAAn 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).

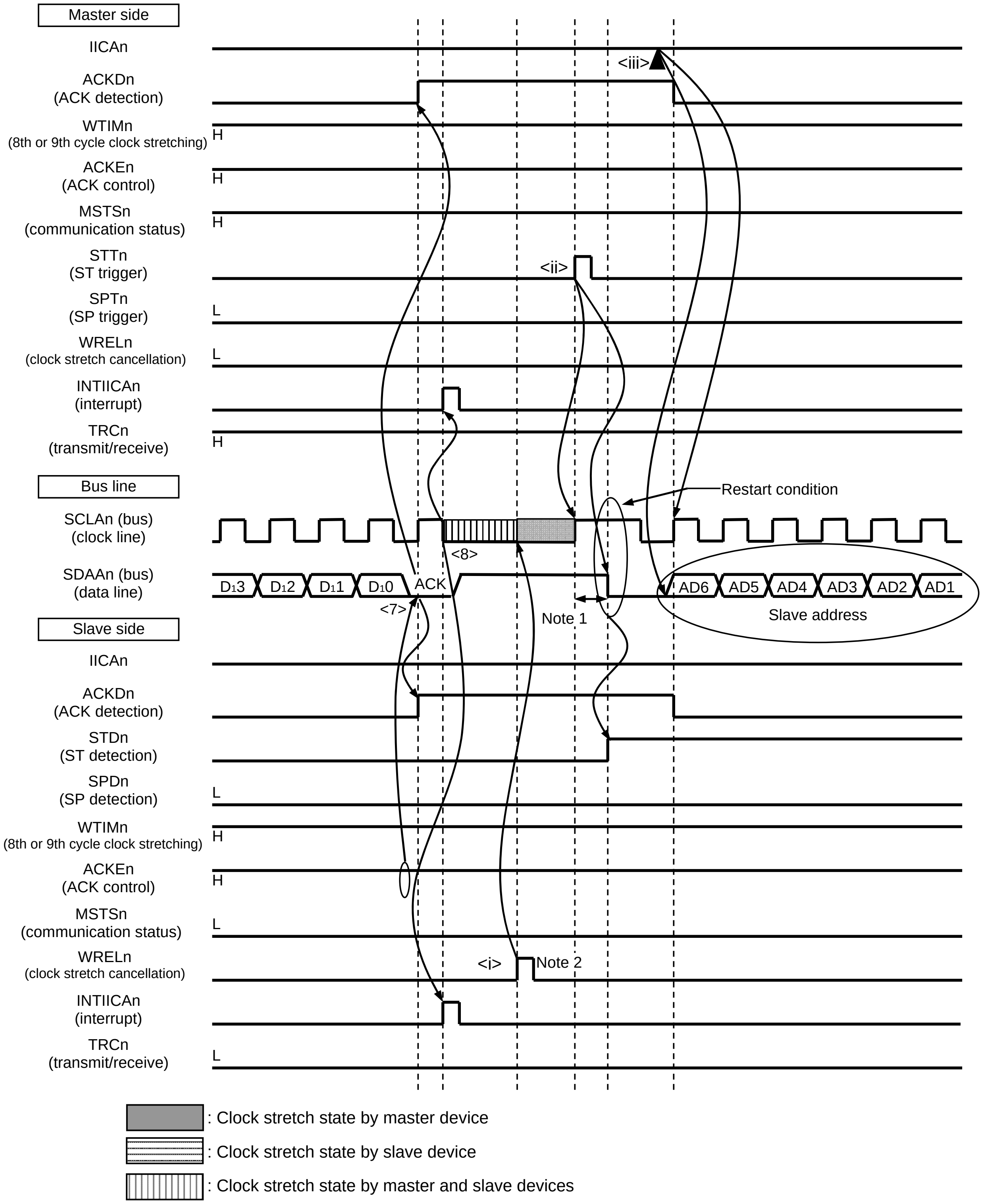
Remarks 1. <1> to <15> in F**igure 21-33** represent the entire procedure for communicating data using the I^{2}C bus.

Figure 21-33 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, **Figure 21-33 (2) Address ~ data ~ data** shows the processing from <3> to <10>, and **Figure 21-33 (3) Data ~ data ~ stop condition** shows the processing from <7> to <15>.

2. n = 0

Figure 21-33. Example of Master to Slave Communication (9-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0

The following describes the operations in **Figure 21-33 (4) Data ~ restart condition ~ address**. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, 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 a clock stretch status (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 status (WRELn = 1).

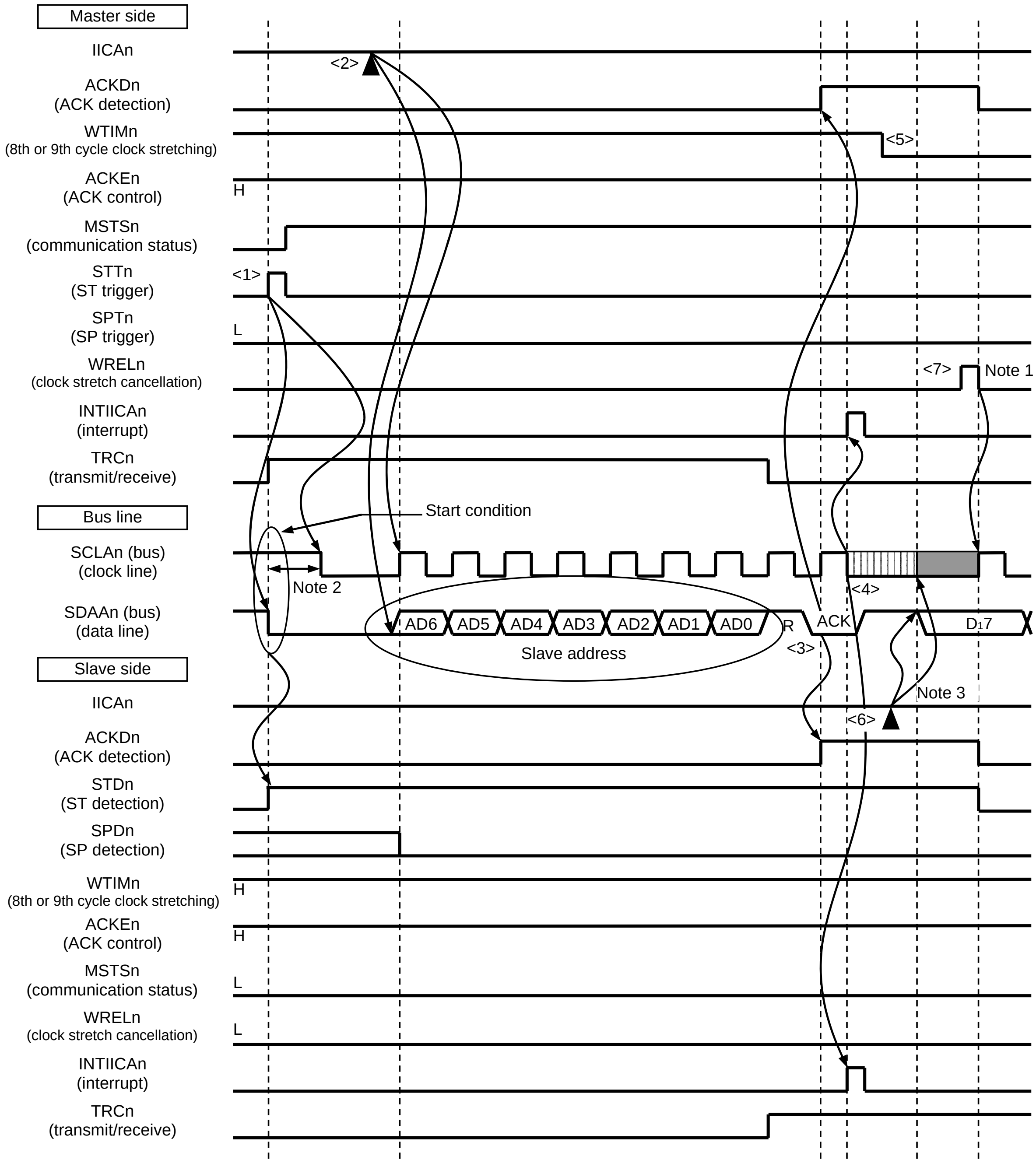
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0

Figure 21-34. Example of Slave to Master Communication (8-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



■ : Clock stretch state by master device

☰ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.

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.

3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Remark n = 0

The following is a description of the **Figure 21-34 (1) Start condition ~ address ~ data** <1> to <7>.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn 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 communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The timing at which the master device sets the clock stretch status 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 status that it set by the slave device.

<7> The master device releases the clock stretch status (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 a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

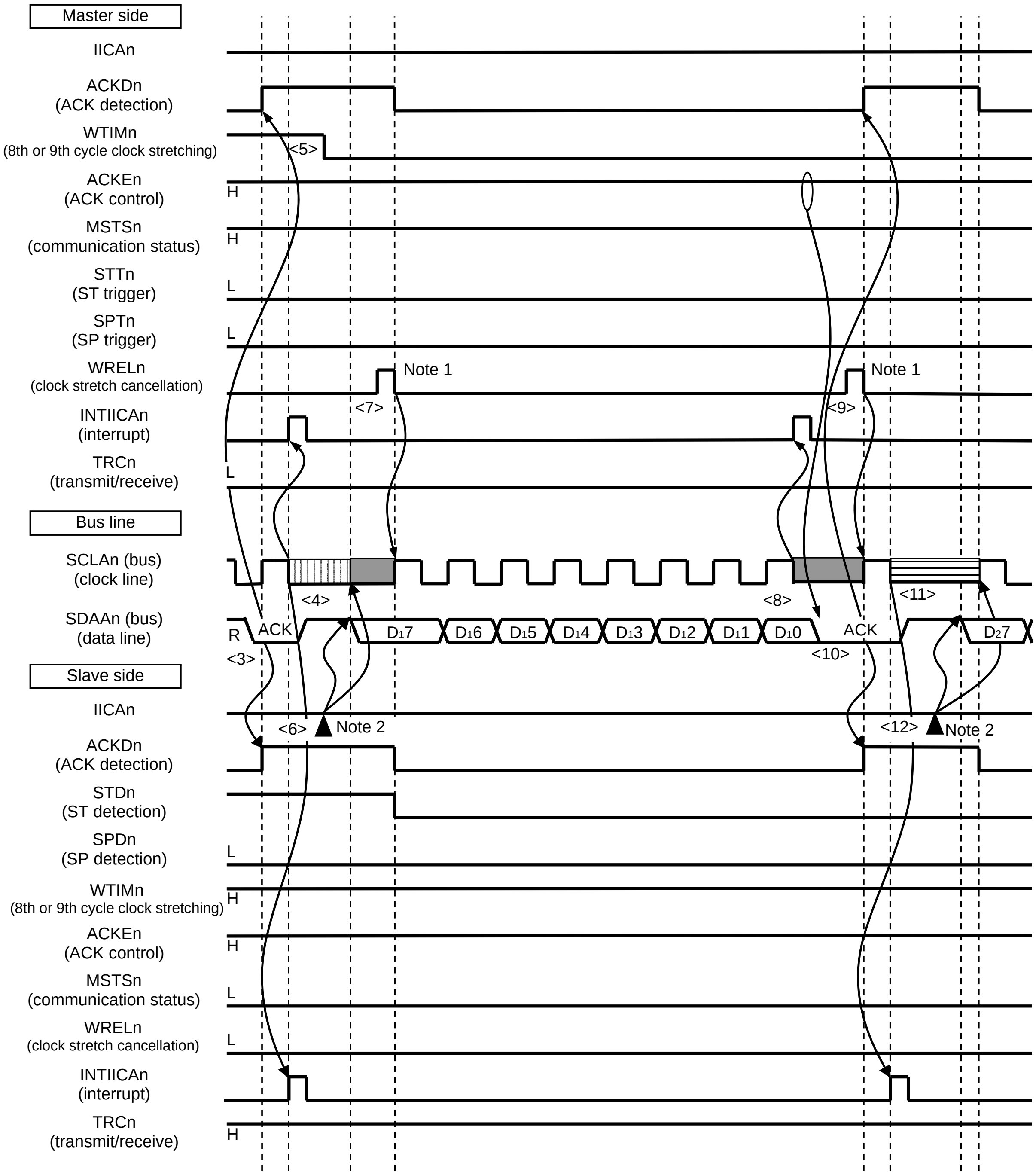
Remarks 1. <1> to <19> in **Figure 21-34** represent the entire procedure for communicating data using the I^{2}C bus.

Figure 21-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 21-34 (2) Address ~ data ~ data** shows the processing from <3> to <12>, and **Figure 21-34 (3) Data ~ data ~ stop condition** shows the processing from <8> to <19>.

2. n = 0

Figure 21-34. Example of Slave to Master Communication (8-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



▨ : Clock stretch state by master device

▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.

2. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Remark n = 0

The following is a description of the **Figure 21-34 (2) Address ~ data ~ data** <3> to <12>.

<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 whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match)[Note].

<5> The master device changes the timing of the clock stretch status 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 status that it set by the slave device.

<7> The master device releases the clock stretch status (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a clock stretch status (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 status (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 set a clock stretch status (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 status 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 a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

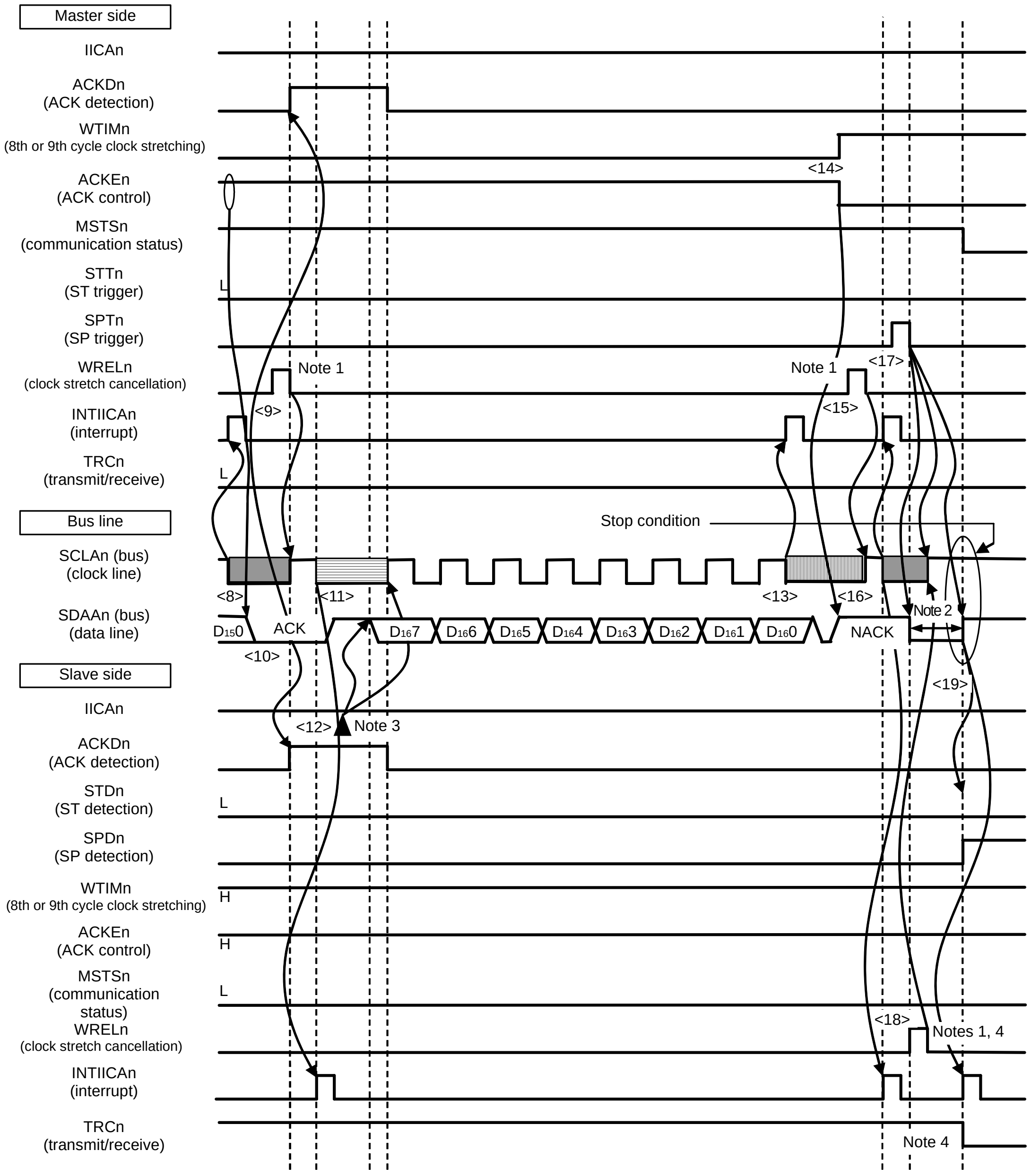
Remarks 1. <1> to <19> in **Figure 21-34** represent the entire procedure for communicating data using the I^{2}C bus.

Figure 21-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 21-34 (2) Address ~ data ~ data** shows the processing from <3> to <12>, and **Figure 21-34 (3) Data ~ data ~ stop condition** shows the processing from <8> to <19>.

2. n = 0

Figure 21-34. Example of Slave to Master Communication
(8-Clock and 9-Clock Clock Stretching Is Selected for Master, 9-Clock Clock Stretching Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



▩ : Clock stretch state by master device

▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Notes 1. To cancel a clock stretch state, write “FFH” to IICAn or set the WRELn bit.

2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

4. If a clock stretch state during transmission by a slave device is canceled by setting the WRELn bit, the TRCn bit will be cleared.

Remark n = 0

The following is a description of the **Figure 21-34 (3) Data ~ data ~ stop condition** <8> to <19>.

<8> The master device sets a clock stretch status (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 status (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 set a clock stretch status (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 status 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 a clock stretch status (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 status to the 9th clock (WTIMn = 1).

<15> If the master device releases the clock stretch status (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 a clock stretch status (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 status. The master device then clock stretchs until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch status (WRELn = 1) to end communication. Once the slave device releases the clock stretch status, 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 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remarks 1. <1> to <19> in **Figure 21-34** represent the entire procedure for communicating data using the I^2C bus.

Figure 21-34 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 21-34 (2) Address ~ data ~ data** shows the processing from <3> to <12>, and **Figure 21-34 (3) Data ~ data ~ stop condition** shows the processing from <8> to <19>.

2. n = 0

CHAPTER 22 SERIAL INTERFACE UARTMG

22.1 Overview

The serial interface (UARTMGn, n = 0, 1) 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.

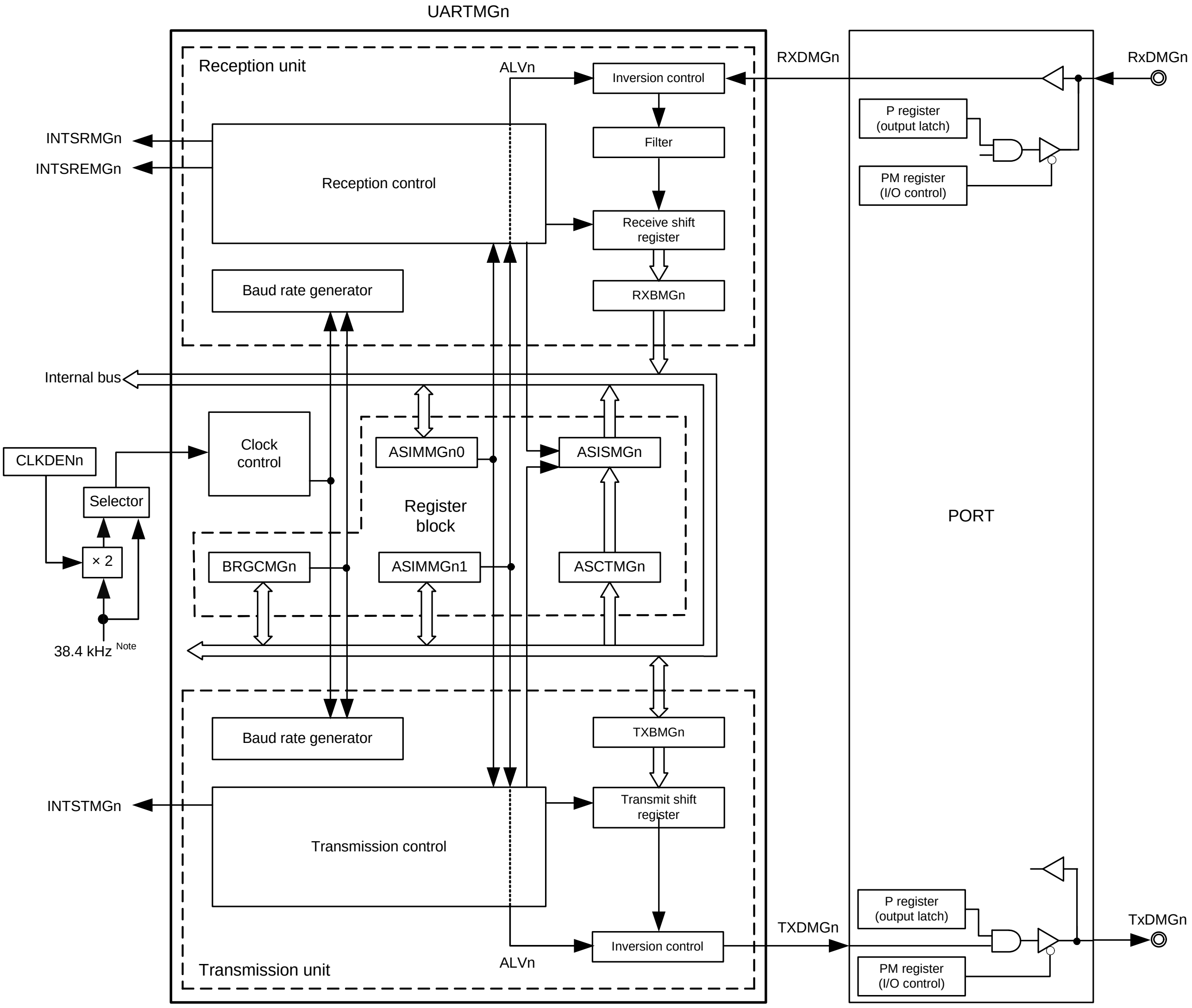
UARTMGnperforms an asynchronous communication. It has the following functions.

- Maximum transfer rate: 9600 bps (19200 bps with a clock doubler used)
- Transmission and reception using two pins
  TxDMGn: Transmit data output pin
  RxDMGn: 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

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

Figure 22-1 shows a block diagram of UARTMGn and **Table 22-1** shows the pin configuration of UARTMGn.

Figure 22-1. Block Diagram of UARTMGn



Note Be sure to select the subsystem clock (WUTMMCK0 bit = 0).

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

Table 22-1. UARTMGn Pin Configuration (n = 0, 1)

Name	I/O	Function
RxDMGn	Input	Serial data input signal
TxDMGn	Output	Serial data output signal

22.2 Register Descriptions

Table 22-2 shows the registers used in UARTMGn.

Table 22-2. UARTMGn Registers

Register Name	Symbol	Value after Reset	Address	Access Size
Peripheral enable register 2	PER2	00H	F00FCH	8
Peripheral reset control register 2	PRR2	00H	F00FDH	8
Clock doubler control register	CLKDCTL	00H	F0076H	8
Transmit buffer register n	TXBMGn	FFH	F0210H, F0218H	8
Receive buffer register n	RXBMGn	FFH	F0211H, F0219H	8
Operation mode setting register n0	ASIMMGn0	01H	F0212H, F021AH	8
Operation mode setting register n1	ASIMMGn1	1AH	F0213H, F021BH	8
Baud rate generator control register n	BRGCMGn	FFH	F0214H, F021CH	8
Status register n	ASISMGn	00H	F0215H, F021DH	8
Status clear trigger register n	ASCTMGn	00H	F0216H, F021EH	8

Remarks 1. UARTMGn uses the following registers, in addition to those listed above.

- Port mode registers 0 and 5 (PM0 and PM5) (See **CHAPTER 4 PORT FUNCTIONS**.)
- Port registers 0 and 5 (P0 and P5) (See **CHAPTER 4 PORT FUNCTIONS**.)

2. n: Channel number (n = 0, 1)

22.2.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the serial interface UARTMG0 is to be used, be sure to set bit 4 (UARTMG0EN) to 1.

When the serial interface UARTMG1 is to be used, be sure to set bit 5 (UARTMG1EN) to 1.

The PER2 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 22-2. Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

UARTMGnEN	Control of serial interface UARTMGn input clock supply
0	Stops input clock supply. • SFR used by the serial interface UARTMGn cannot be written. The read value is 00H. However, the SFR is not initialized. Notes 1, 2
1	Enables input clock supply. • SFR used by the serial interface UARTMGn can be read and written.

Notes 1. To initialize the serial interface UARTMG0 and the SFR used by the serial interface UARTMG0, use bit 4 (UARTMG0RES) of PRR2.

2. To initialize the serial interface UARTMG1 and the SFR used by the serial interface UARTMG1, use bit 5 (UARTMG1RES) of PRR2.

Cautions 1. When using the serial interface UARTMGn, be sure to set the following registers first while the UARTMGnEN bit is set to 1. If UARTMGnEN = 0, writing to the registers controlling the serial interface UARTMGn is ignored, and, even if the register is read, only the default value is read (except for the port mode registers 0, 5 (PM0, PM5) and port registers 0, 5 (P0, P5)).

- **Clock doubler control register (CLKDCTL)**
- **Transmit buffer register n (TXBMGn)**
- **Receive buffer register n (RXBMGn)**
- **Operation mode setting register n0 (ASIMMGn0)**
- **Operation mode setting register n1 (ASIMMGn1)**
- **Baud rate generator control register n (BRGCMGn)**
- **Status register n (ASISMGn)**
- **Status clear trigger register n (ASCTMGn)**

2. Be sure to clear bits 1 and 3 to 0.

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

22.2.2 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.

To place the serial interface UARTMG0 in the reset state, be sure to set bit 4 (UARTMG0RES) to 1.

To place the serial interface UARTMG1 in the reset state, be sure to set bit 5 (UARTMG1RES) to 1.

The PRR2 register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 22-3. Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	0
PRR2	TMKARES	OSDCRES	UARTMG1RES	UARTMG0RES	0	MACRES	0	0

UARTMGnRES	Control resetting of the serial interface UARTMGn
0	The serial interface UARTMGn is released from the reset state.
1	The serial interface UARTMGn is in the reset state.

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

22.2.3 Clock Doubler Control Register (CLKDCTL)

The CLKDCTL register sets the the operating clock of the serial interface UARTMGn.

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

Reset signal generation clears this register to 00H.

Figure 22-4. Format of Clock Doubler Control Register (CLKDCTL)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
CLKDCTL	0	0	0	0	0	0	CLKDEN1	CLKDEN0

CLKDENn	Selection of operating clock for serial interface UARTMGn
0	f_{SX}
1	f_{SX} × 2

Cautions 1. The clock doubler starts operation when CLKDENn is set to 1.

2. Setting the CLKDENn bit during the UARTMGn operation is prohibited. Be sure to set the bit to 1 before UARTMGn starts communication.

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

22.2.4 Transmit Buffer Register (TXBMGn) (n = 0, 1)

The TXBMGn register can be read and written by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 22-5. Format of Transmit Buffer Register (TXBMGn)

Address: F0210H (TXBMG0), F0218H (TXBMG1) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
TXBMGn	—	—	—	—	—	—	—	—

Bits 7 to 0	Function
—	A buffer register for setting transmit data. Transmission starts by writing data for transmission to the TXBMGn register. When a character length of 8 bits is specified: • Data in bits 7 to 0 of TXBMGn are transferred. When a character length of 7 bits is specified: • Data in bits 6 to 0 of TXBMGn 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 TXBMGn are transferred in either MSB- or LSB-first mode; bits 7 to 5 are invalid.

Cautions 1. When the TXBFMGn bit of the ASISMGn register is 1, writing to the TXBMGn register is prohibited.

2. After setting TXEMGn = 1, wait for at least one cycle of the base clock (f_{SX}) before setting the first data for transmission in the TXBMGn register. Although the data for transmission can be set normally even when the first data for transmission is set in the TXBMGn register before one cycle of the base clock has elapsed, this delays the start of transmission.

Remark Transmit shift register:

Data is transferred from the TXBMGn register to this register, and is then transmitted as serial data through the TXDMGn pin. In the first transmission, data is transferred from the TXBMGn register to this register immediately after data is written to the TXBMGn register. In continuous transmission, data is transferred after transmission of one frame and just before generation of the transmission completion interrupt.

The transmit shift register cannot be manipulated directly by a program.

22.2.5 Receive Buffer Register (RXBMGn) (n = 0, 1)

The RXBMGn register can be read by an 8-bit memory manipulation instruction. Writing to this register is disabled.
Reset signal generation sets this register to FFH.

Figure 22-6. Format of Receive Buffer Register (RXBMGn)

Address: F0211H (RXBMG0), F0219H (RXBMG1) After reset: FFH R

Symbol	7	6	5	4	3	2	1	0
RXBMGn	—	—	—	—	—	—	—	—

Bits 7 to 0	Function
—	This register stores the parallel data converted by the receive shift register. Every time one byte of data is received, the next receive data is transferred from the receive shift register [Note] to this register. When a character length of 8 bits is specified: • Receive data is transferred to bits 7 to 0 of this register. When a character length of 7 bits is specified: • Receive data is transferred to bits 6 to 0 of this register in either MSB- or LSB-first mode; bit 7 is always 0. When a character length of 5 bits is specified: • Receive data is transferred to bits 4 to 0 of this register in either MSB- or LSB-first mode; bits 7 to 5 are always 0.

Note The receive shift register converts the serial data that is input through the RXDMGn pin to parallel data. The receive shift register cannot be manipulated directly by a program.

Caution If an overrun error (OVEMGn) occurs, the data received at that time are not stored in the RXBMGn register.

22.2.6 Operation Mode Setting Register 0 (ASIMMGn0) (n = 0, 1)

The ASIMMGn0 register is an 8-bit register that controls serial communication of the serial interface UARTMGn.
The ASIMMGn0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 01H.

Figure 22-7. Format of Operation Mode Setting Register 0 (ASIMMGn0)

Address: F0212H (ASIMMG00), F021AH (ASIMMG10) After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
ASIMMGn0	POWERMGn	TXEMGn	RXEMGn	—	—	—	ISSMMGn	ISRMMGn

POWERMGn Note 1	UART operation enable
0	Disables the UART operation clock (fixed to low level). Asynchronously resets the internal circuits. Note 2
1	Enables the UART operation clock.

TXEMGn	Transmission enable
0	Disables transmission. (Synchronously resets the transmission circuit.)
1	Enables transmission.

RXEMGn	Reception enable
0	Disables reception. (Synchronously resets the reception circuit.)
1	Enables reception.

ISSMMGn	Transmit interrupt mode select
0	The INTSTMGn interrupt is generated on completion of transmission.
1	The INTSTMGn interrupt is generated when the transmit buffer becomes empty. (for continuous transmission)

ISRMMGn	Receive interrupt mode select
0	The INTSREMGn interrupt is generated when a reception error occurs. (INTSRMGn is not generated.)
1	The INTSRMGn interrupt is generated when a reception error occurs. (INTSREMGn is not generated.)

Notes 1. When POWERMGn = 0, the level being output from the TXDMGn pin and the level being input from the RXDMGn pin are determined according to the setting of the ALVn bit as described below.
When ALVn = 0, output from the TXDMGn pin is high, and input from the RXDMGn pin is fixed to the high level.
When ALVn = 1, output from the TXDMGn pin is low, and input from the RXDMGn pin is fixed to the low level.

2. The ASISMGn and RXBMGn registers are reset by clearing the POWERMGn bit to 0.

(**Cautions** are listed on the next page.)

Cautions 1. To start transmission, set the POWERMGn bit to 1 and then set the TXEMGn bit to 1. To stop transmission, clear the TXEMGn bit to 0 and then clear the POWERMGn bit to 0.

2. To start reception, set the POWERMGn bit to 1 and then set the RXEMGn bit to 1. To stop reception, clear the RXEMGn bit to 0 and then clear the POWERMGn bit to 0.

3. Follow the procedure below when setting the POWERMGn bit to 1 and then setting the RXEMGn bit to 1.

When ALVn = 0, the setting must be made while the level being input to the RXDMGn pin is high. Otherwise, reception is started at that point.

When ALVn = 1, the setting must be made while the level being input to the RXDMGn pin is low. Otherwise, reception is started at that point.

4. TXEMGn and RXEMGn are synchronized with the sub clock (f_{SX}). To enable transmission or reception again, set the TXEMGn or RXEMGn bit to 1 at least two cycles of the sub clock after clearing the TXEMGn or RXEMGn bit to 0. If the bit is set to 1 within two cycles of the sub clock after the clearing, the transmission or reception circuit may not be able to be initialized.

5. After setting the TXEMGn bit to 1, wait at least one cycle of the sub clock before setting the transmit data in the TXBMGn register.

6. Clear the RXEMGn bit to 0 before modifying the ISRMMGn bit.

7. Be sure to clear bits 2, 3, and 4 to 0.

22.2.7 Operation Mode Setting Register 1 (ASIMMGn1) (n = 0, 1)

The ASIMMGn1 register is an 8-bit register that controls serial communication of the serial interface UARTMGn.
The ASIMMGn1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
The ASIMMGn1 register must be modified while TXEMGn = 0 and RXEMGn = 0.
Reset signal generation sets this register to 1AH.

Figure 22-8. Format of Operation Mode Setting Register 1 (ASIMMGn1)

Address: F0213H (ASIMMG01), F021BH (ASIMMG11) After reset: 1AH R/W

Symbol	7	6	5	4	3	2	1	0
ASIMMGn1	—	PSn1	PSn0	CLn1	CLn0	SLn	DIRn	ALVn

PSn1 [Note 1]	PSn0 [Note 1]	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 4]
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 [Note 3]	CLn0 [Note 3]	Transmission/reception character length setting 1, 0
0	0	Character length of data = 5 bits
0	1	Character length of data = 5 bits
1	0	Character length of data = 7 bits
1	1	Character length of data = 8 bits

SLn [Note 2]	Transmission stop bit length setting
0	Stop bit length = 1 bit
1	Stop bit length = 2 bits

DIRn [Note 1]	Transmission/reception order setting
0	MSB first
1	LSB first

ALVn [Note 1]	Transmission/reception level setting [Note 5]
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)

(**Notes** and **Cautions** are listed on the next page.)

Notes 1. Modify the ALVn, DIRn, PSn0, and PSn1 bits while the TXEMGn and RXEMGn bits are 0 (in the transmission/reception stopped state).

2. Modify the SLn bit while the TXEMGn bit is 0 (in the transmission stopped state).

3. Modify the CLn0 and CLn1 bits while the TXEMGn and RXEMGn bits are 0 (in the transmission/reception stopped state). Otherwise, the character length of communication data is not guaranteed.

4. When “Data is received with 0 parity” is set, parity check is not performed. Accordingly the PEMGn bit of the ASISMGn register is not set: no error interrupts are generated.

5. Data level inversion is controlled in the “Inversion control” part in **Figure 22-1 Block Diagram of UARTMGn**.

Cautions 1. Clear both the TXEMGn and RXEMGn bits to 0 before modifying the ASIMMGn1 register.

2. Reception is always handled as including 1 stop bit; the setting of the SLn bit does not affect reception.

3. Be sure to clear bit 7 to 0.

22.2.8 Baud Rate Generator Control Register (BRGCMGn) (n = 0, 1)

The BRGCMGn register sets the frequency divisor for the 8-bit counter in the serial interface UARTMGn.

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

Reset signal generation sets this register to FFH.

Figure 22-9. Format of Baud Rate Generator Control Register (BRGCMGn)

Address: F0214H (BRGCMG0), F021CH (BRGCMG1) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGCMGn	—	—	—	—	—	—	—	—

Bits 7 to 0	Function
—	Controls the UART baud rate (serial transfer speed). For details on the settings, see **Table 22-3**.

Caution Modify the BRGCMGn bits while the TXEMGn and RXEMGn bits are 0 (in the transmission/reception stopped state).

Table 22-3. BRGCMGn Settings

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	k	Selection of 8-bit counter output clock
0	0	0	0	0	0	0	X	X	Setting prohibited
0	0	0	0	0	0	1	0	2	fsx/2
0	0	0	0	0	0	1	1	3	fsx/3
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	fsx/252
1	1	1	1	1	1	0	1	253	fsx/253
1	1	1	1	1	1	1	0	254	fsx/254
1	1	1	1	1	1	1	1	255	fsx/255

Caution The baud rate is one half the frequency of the output clock signal from the 8-bit counter.

Remarks 1. k: The value set with the BRGCMGn bits (k = 2, 3, 4, 5, 6, …, 255)

2. X: Don’t care.

For an example of the baud rate setting, see **22.3.4 (3) (c) Baud rate setting example**.

22.2.9 Status Register (ASISMGn) (n = 0, 1)

The ASISMGn register indicates the error status and the transmission status on completion of reception by the serial interface UARTMGn. It consists of three error flag bits (PEMGn, FEMGn, and OVEMGn) and two transmission status flag bits (TXBFMGn and TXSFMGn).

The ASISMGn register is read-only and can be read by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H. The PEMGn, FEMGn, and OVEMGn bits are initialized by clearing the POWERMGn or RXEMGn bit to 0. These bits are also cleared by writing to the corresponding bit of the ASCTMGn register. The TXBFMGn and TXSFMGn bits are initialized by clearing the POWERMGn or TXEMGn bit to 0.

Figure 22-10. Format of Status Register (ASISMGn)

Address: F0215H (ASISMG0), F021DH (ASISMG1) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASISMGn	—	—	TXBFMGn	TXSFMGn	—	PEMGn	FEMGn	OVEMGn

TXBFMGn	Transmit buffer data flag
—	[Setting condition] • Data is written to the TXBMGn register. (Data exists in the TXBMGn register.) [Clearing conditions] • The POWERMGn or TXEMGn bit is cleared to 0. • Data is transferred to the transmit shift register.

TXSFMGn	Transmit shift register data flag
—	[Setting condition] • Data is transferred from the TXBMGn register. (Data is being transmitted.) [Clearing conditions] • The POWERMGn or TXEMGn bit is cleared to 0. • Data is transferred from the transmit shift register and then no subsequent data is transferred from the TXBMGn register.

PEMGn	Parity error flag
—	[Setting condition] • The parity of the received data does not match the parity bit. [Clearing conditions] • The POWERMGn or RXEMGn bit is cleared to 0. • 1 is written to the PECTMGn bit.

FEMGn	Framing error flag
—	[Setting condition] • A stop bit is not detected when receiving data. [Clearing conditions] • The POWERMGn or RXEMGn bit is cleared to 0. • 1 is written to the FECTMGn bit.

OVEMGn	Overrun error flag
—	[Setting condition] • The next reception is completed before the receive data in the RXBMGn register is read. [Clearing conditions] • The POWERMGn or RXEMGn bit is cleared to 0. • 1 is written to the OVECTMGn bit.

(**Cautions** are listed on the next page.)

Cautions

1. **For continuous transmission, be sure to check that the TXBFMGn flag is 0 after writing the the first transmit data (the first byte) to the TXBMGn register and then write the next transmit data (the second byte) to the TXBMGn register. Otherwise, the reliability of the transmit data is not guaranteed. However, the TXBFMGn flag need not be checked when continuous transmission is performed by using the transmit buffer empty interrupt (ISSMMGn bit = 1).**
2. **When initializing the transmission unit (TXEMGn = 0) after completion of continuous transmission, be sure to check that the TXSFMGn flag is 0 after the transmission completion interrupt is generated, and then initialize the unit. Otherwise, the reliability of the transmit data is not guaranteed.**
3. **The setting condition for the PEMGn bit depends on the setting of the PSn1 and PSn0 bits of the ASIMMGn1 register.**
4. **For the receive data, only the first 1 bit of the stop bits is checked regardless of the stop bit length.**
5. **When an overrun error occurs, the next receive data is not written to the RXBMGn register and discarded.**
6. **Be sure to clear bits 7, 6, and 3 to 0.**

22.2.10 Status Clear Trigger Register (ASCTMGn) (n = 0, 1)

The ASCTMGn register sets the trigger to clear the error status on completion of reception of the serial interface UARTMGn. It contains 3 bits of the error clear trigger flags (PECTMGn, FECTMGn, and OVECTMGn).

The ASCTMGn register can be written by an 8-bit or 1-bit memory manipulation instruction.

When the ASISMGn register is read, all bits are always read as 0.

Reset signal generation clears this register to 00H. Writing 1 to the PECTMGn, FECTMGn, and OVECTMGn bits clears the PEMGn, FEMGn, and OVEMGn bits of the ASISMGn register, respectively. When writing 0, the corresponding error flags are not cleared.

Figure 22-11. Format of Status Clear Trigger Register (ASCTMGn)

Address: F0216H (ASCTMG0), F021EH (ASCTMG1) After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
ASCTMGn	—	—	—	—	—	PECTMGn	FECTMGn	OVECTMGn

PECTMGn	Parity error flag clear trigger
0	Does not clear the PEMGn flag. (The flag is retained.)
1	Clears the PEMGn flag.

FECTMGn	Framing error flag clear trigger
0	Does not clear the FEMGn flag. (The flag is retained.)
1	Clears the FEMGn flag.

OVECTMGn	Overrun error flag clear trigger
0	Does not clear the OVEMGn flag. (The flag is retained.)
1	Clears the OVEMGn flag.

Cautions 1. When reading the ASCTMGn register, 0 is returned.

2. After writing 1 to the trigger bit, the corresponding error flag is cleared on the next rising edge of the operating clock (f_{SX}). Accordingly, if reading the ASISMGn register immediately after writing 1 to the trigger bit, the corresponding error flag may not have been cleared yet.

22.3 Operation

UARTMGn operates in the following two modes

- Operation stop mode
- UART mode

22.3.1 Operation Stop Mode

In the operation stop mode, serial communication is not performed, and thus the power consumption can be reduced. In addition, in this mode, the pins can be used as ordinary port pins. To set the operation stop mode, clear bits 7, 6, and 5 (POWERMGn, TXEMGn, RXEMGn) of the ASIMMGn0 register to 0.

The bus clock is not stopped by the above setting. To completely stop operation, clear the UARTMGnEN bit of the PER2 register to 0 after the above setting.

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

22.3.2 UART Mode

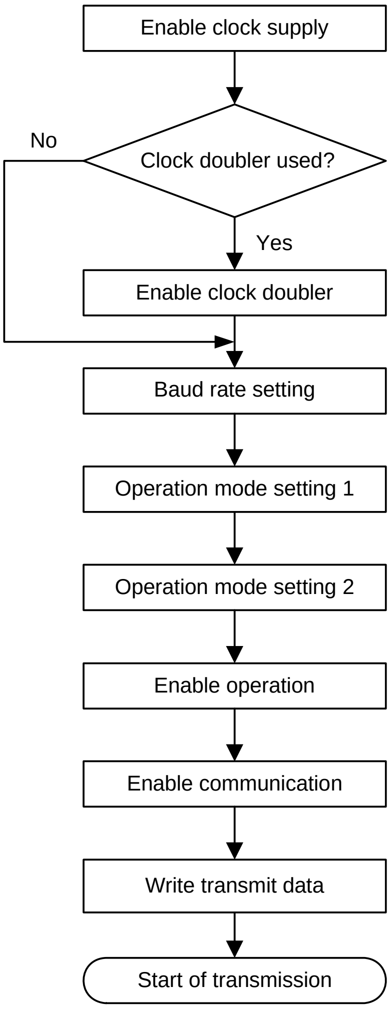
In this mode, one byte of data is transmitted and one byte is received following the start bit. That is, operation is full duplex.

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 22-12 shows the flowchart of communication procedure.

Figure 22-12. Flowchart of Communication Procedure



Set the UARTnEN bit of the PER2 register to 1.

Set the CLKDENn bit of the CLKDCTL register to 1.

Set the BRGCMGn register.

Set bits 0 to 6 (ALVn, DIRn, SLn, CLn0, CLn1, PSn0, and PSn1) of the ASIMMGn1 register.

Set bits 0 and 1 (ISRMMGn and ISSMMGn) of the ASIMMGn0 register.

Set bit 7 (POWERMGn) of the ASIMMGn0 register to 1.

Set bit 6 (TXEMGn) of the ASIMMGn0 register to 1 to enable transmission.
Set bit 5 (RXEMGn) of the ASIMMGn0 register to 1 to enable reception.

Write transmit data to the TXBMGn register.

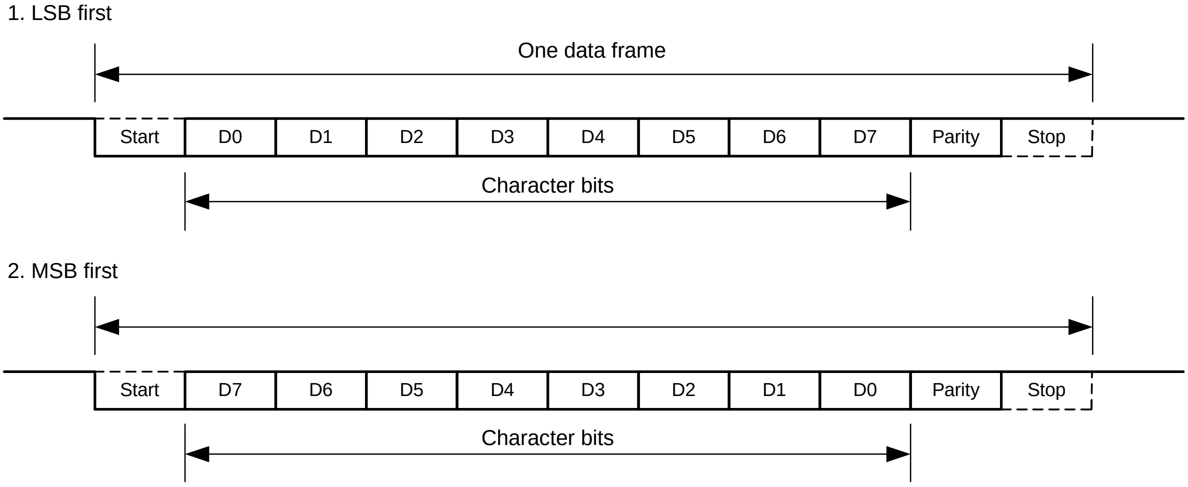
Caution When using the receiving function, set the port mode registers to input mode.
When using the transmitting function, set the port mode registers to output mode, and set the port register to the high level.
Frequency-multiplied clock starts to be supplied with a delay of one cycle of the sub clock (f_{SX}) after setting the CLKDCTL register.

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

(2) Format and waveform example of transmit/receive data

The following describes the communication data format of UARTMGn.
Figure 22-13 shows the data format.

Figure 22-13. Transmit/Receive Data Format



One data frame consists of the following bits.

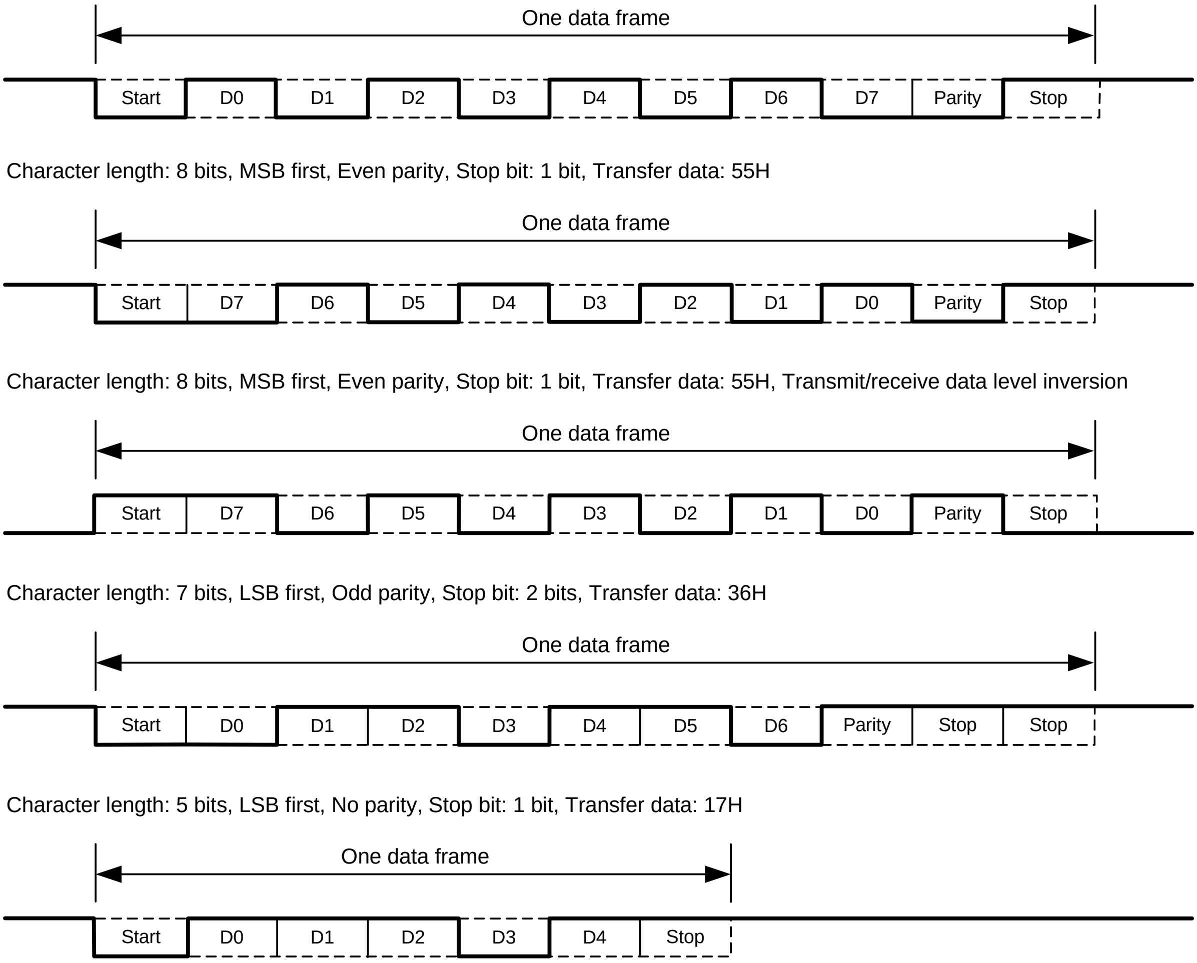
• Start bit:	1 bit
• Character bits:	5, 7 or 8 bits
• Parity bit:	Even parity, odd parity, 0 parity, or no parity
• Stop bit:	1 or 2 bits

The character bit length, the parity, the stop bit length, the transfer direction (LSB/MSB first), and the TXDMGn pin output (direct/inverted) in one data frame are specified by the ASIMMGn1 register.

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

Figure 22-14 shows the examples of transmit/receive data waveforms.

Figure 22-14. Example of Transmit/Receive Data Waveform



Character length: 8 bits, LSB first, Even parity, Stop bit: 1 bit, Transfer data: 55H

Character length: 8 bits, MSB first, Even parity, Stop bit: 1 bit, Transfer data: 55H

Character length: 8 bits, MSB first, Even parity, Stop bit: 1 bit, Transfer data: 55H, Transmit/receive data level inversion

Character length: 7 bits, LSB first, Odd parity, Stop bit: 2 bits, Transfer data: 36H

Character length: 5 bits, LSB first, No parity, Stop bit: 1 bit, Transfer data: 17H

(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) Parity types and operation

- In transmission

  Data for transmission, including the parity bit, are controlled so that an even number of bits have the value “1”.

  The value of the parity bit is set as follows.

  If the data for transmission have an odd number of bits with the value “1”: 1

  If the data for transmission have an even number of bits with the value “1”: 0

- In reception

  In the data for reception, including the parity bit, the number of bits with the value "1" , is counted. If it is odd, a parity error occurs.

(b) Odd parity

- In transmission

  Unlike even parity, data for transmission, including the parity bit, are controlled so that an odd number of bits have the value “1”.

  If the data for transmission have an odd number of bits with the value “1”: 0

  If the data for transmission have an even number of bits with the value “1”: 1

- In reception

  In the data for reception, including the parity bit, the number of bits with the value "1" , is counted. If it is even, a parity error occurs.

(c) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.

The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(d) No parity

No parity bit is appended to the transmit data.

Reception is performed assuming that there is no parity bit. A parity error does not occur, because there is no parity bit.

(4) Normal transmission

Transmission is enabled by setting bit 7 (POWERMGn) of the operation mode register 0 (ASIMMGn0) to 1 and then setting bit 6 (TXEMGn) of ASIMMGn0 to 1. Transmission can be started by writing the data for transmission to the transmission buffer register (TXBMGn). The start bit, parity bit, and stop bit are automatically appended to the data.
When transmission is started, the data in the TXBMGn register is transferred to the transmit shift register. After that, the transmit data bits are sequentially output from the transmit shift register to the TXDMGn pin in the specified transfer direction. When transmission is completed, the parity and stop bits which are set by the ASIMMGn0 register are appended and a transmission completion interrupt request (INTSTMGn) is generated.
Transmission is suspended until the next transmit data is written to the TXBMGn register.

Figure 22-15 shows the timing of the transmission completion interrupt request (INTSTMGn). INTSTMGn is issued at the following timing.

(a) When ISSMMGn = 0 (INTSTMGn functions as a transmission completion interrupt.)

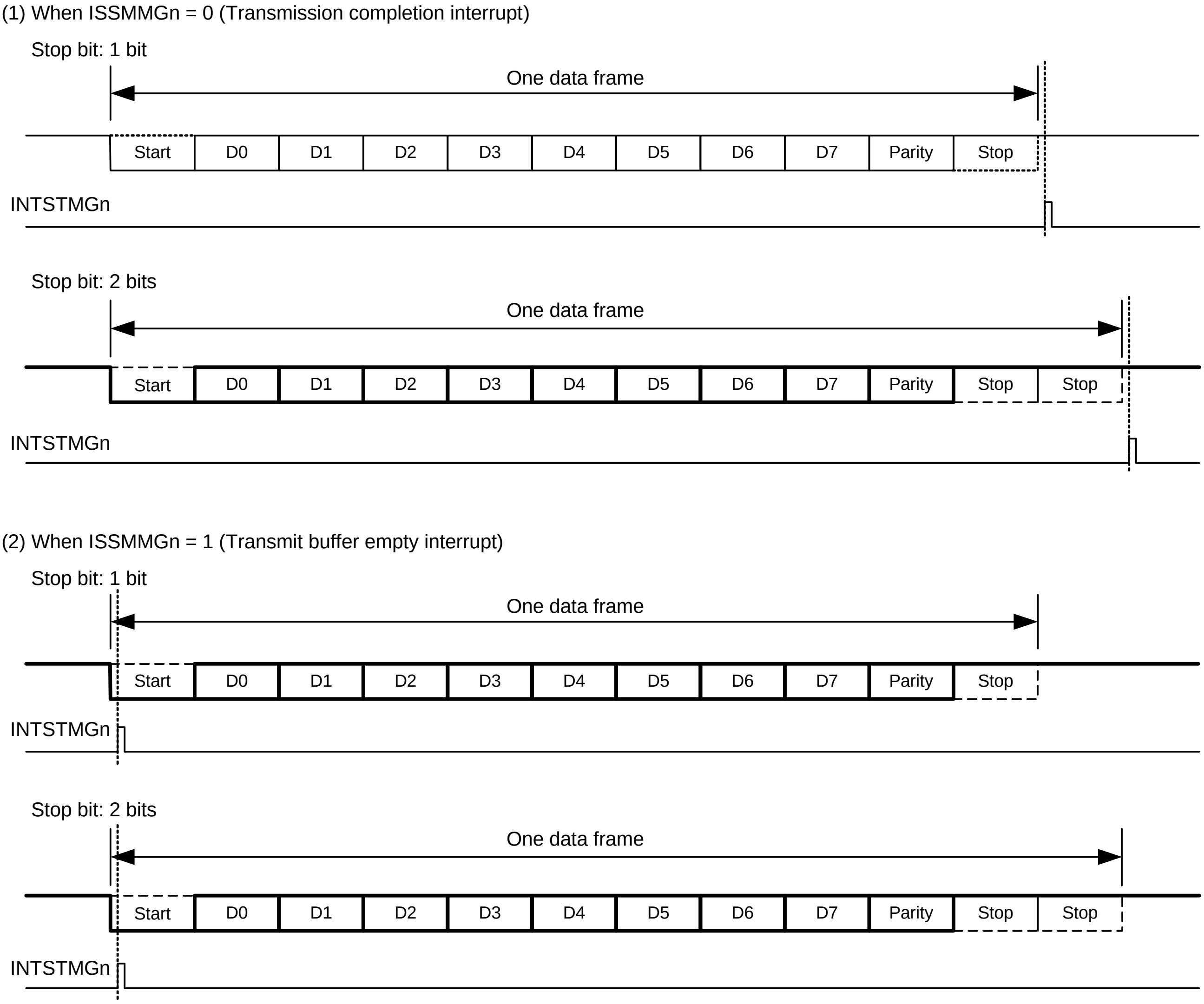
INTSTMGn is issued after the output of the last stop bit.

(b) ISSMMGn = 1 (INTSTMGn functions as a transfer buffer empty interrupt.)

INTSTMGn is issued when the start bit is output.

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

Figure 22-15. Interrupt Output Timing



(1) When ISSMMGn = 0 (Transmission completion interrupt)

Stop bit: 1 bit

Stop bit: 2 bits

(2) When ISSMMGn = 1 (Transmit buffer empty interrupt)

Stop bit: 1 bit

Stop bit: 2 bits

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

(5) Continuous transmission

UARTMGn has two separate registers for continuous transmission: the transmit buffer register (TXBMGn) and the transmit shift register.
At the moment the transmit shift register starts a shift operation, the next transmit data can be written to the transmit buffer register (TXBMGn). This operation enables continuous transmission, thereby improving communication rate.
Continuous transmission is achieved by the following two methods.

(a) Parity types and operation

Continuous transmission is achieved by polling the transmit buffer data flag (bit 5: TXBFMGn) and the transmit shift register data flag (bit 4: TXSFMGn) of the status register (ASISMGn).

When using this method, clear bit 1 (ISSMMGn) of the operation mode setting register 0 (ASIMMGn0) to 0.

- At the start of continuous transmission

At the start of continuous transmission, write the first byte of data to the TXBMGn register, check that the transmit buffer data flag (TXBFMGn) is 0, and then write the second byte of data.

TXBFMGn	Determination flag indicating that writing to TXBMGn is enabled or disabled at the start of continuous transmission
0	Writing is enabled.
1	Writing is disabled.

Caution At the start of continuous transmission, only check this flag. The TXSFMGn flag must not be used for judgment in combination with this flag, as its value changes from 0 to 1.

- During continuous transmission

When continuous transmission is in progress, check the transmit shift register data flag after the transmission completion interrupt is generated to determine if the subsequent data can be written to the TXBMGn register.

When the flag is 1: Continuous transmission is in progress. One byte of data can be written.

When the flag is 0: Continuous transmission has been completed. Two bytes of data can be written according to the method used at the start of continuous transmission.

TXSFMGn	Determination flag for write processing during continuous transmission
0	Writing 2 bytes of data or transmission end processing is enabled.
1	Writing 1 byte of data is enabled.

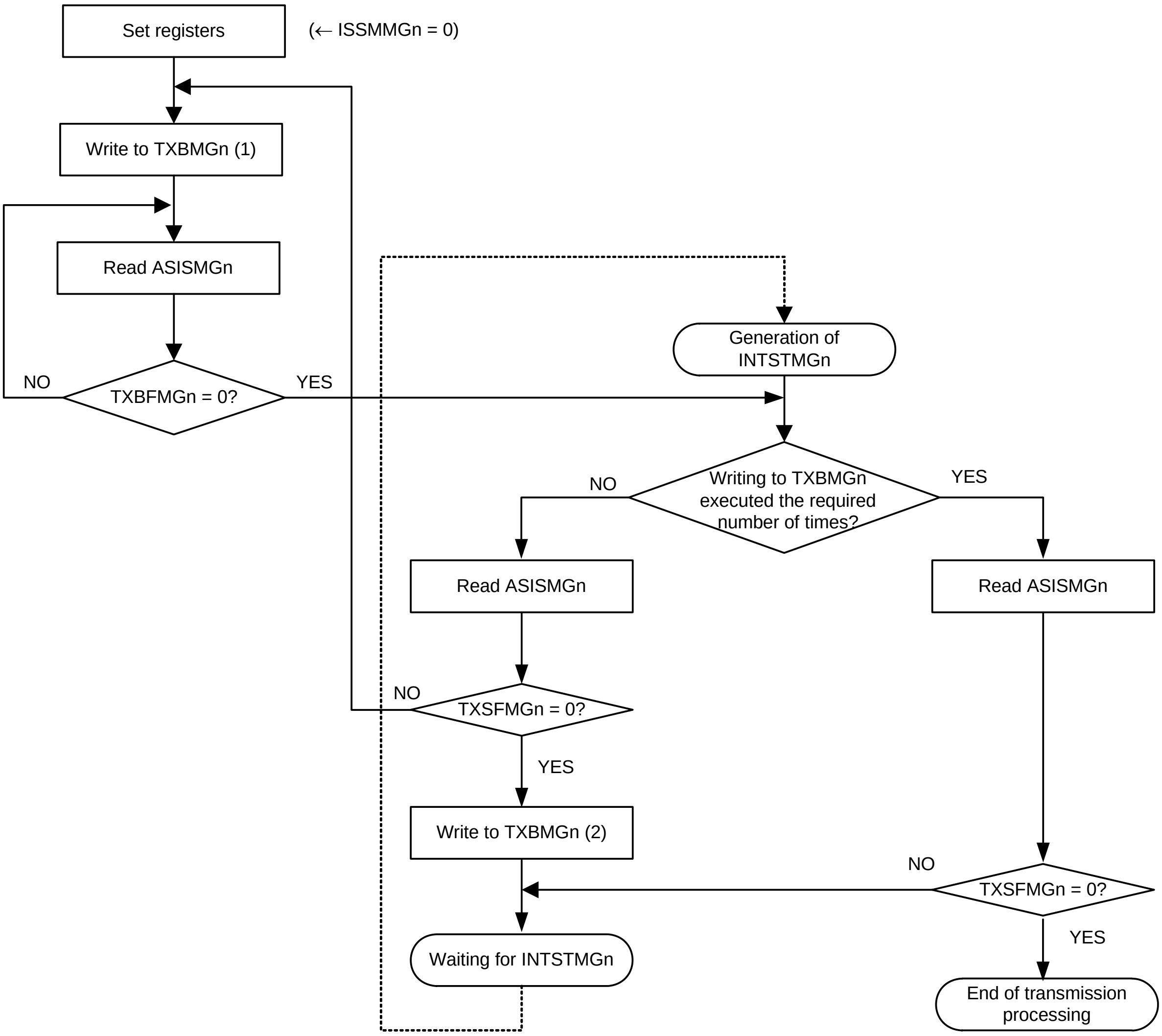
Cautions 1. To execute initialization during continuous transmission, check before initialization that this flag has been cleared 0 after generation of a transmission completion interrupt. If initialization is executed while this flag is 1, the reliability of the transmit data is not guaranteed.

2. During continuous transmission, after transmission of one data frame, the subsequent transmission may be completed before execution of the INTSTMGn interrupt processing. This is called an overrun error. An overrun error can be detected by incorporating the program that counts the number of transmit data and by referencing the TXSFMGn flag.

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

Figure 22-16 shows a flow example of continuous transmission processing by polling.

Figure 22-16. Flow Example of Continuous Transmission Processing by Polling



TXBFMGn
 0: Writing to TXBMGn is enabled.
 1: Writing to TXBMGn is disabled.

TXSFMGn
 0: Writing 2 bytes of data or transmission end processing is enabled.
 1: Writing 1 byte of data is enabled.

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

(b) Continuous transfer by using an interrupt

Continuous transmission is achieved by using the interrupt (INTSTMGn).

The transfer completion interrupt can be switched to the transmit buffer register (TXBMGn) empty interrupt by setting bit 1 (ISSMMGn) to 1 in the operation mode setting register 0 (ASIMMGn0).

With this setting, data can be written to the TXBMGn register on occurrence of the INTSTMGn interrupt; thereby continuous transmission is enabled.

In addition, the transfer completion interrupt can be generated on completion of continuous transmission by clearing the ISSMMGn bit to 0 after writing the last transmit data to the TXBMGn register.

Note that continuous transmission is not achieved when writing to the TXBMGn register is not completed within the maximum number of clock cycles defined below from generation of the transmit buffer register (TXBMGn) empty interrupt.

Maximum number of clock cycles = Data transfer length × 2k - (2k + 3)

k: the value set with the BRGCMGn bits (k = 2, 3, 4, 5, 6, …, 255)

An example of calculating the maximum number of clock cycles is described below.

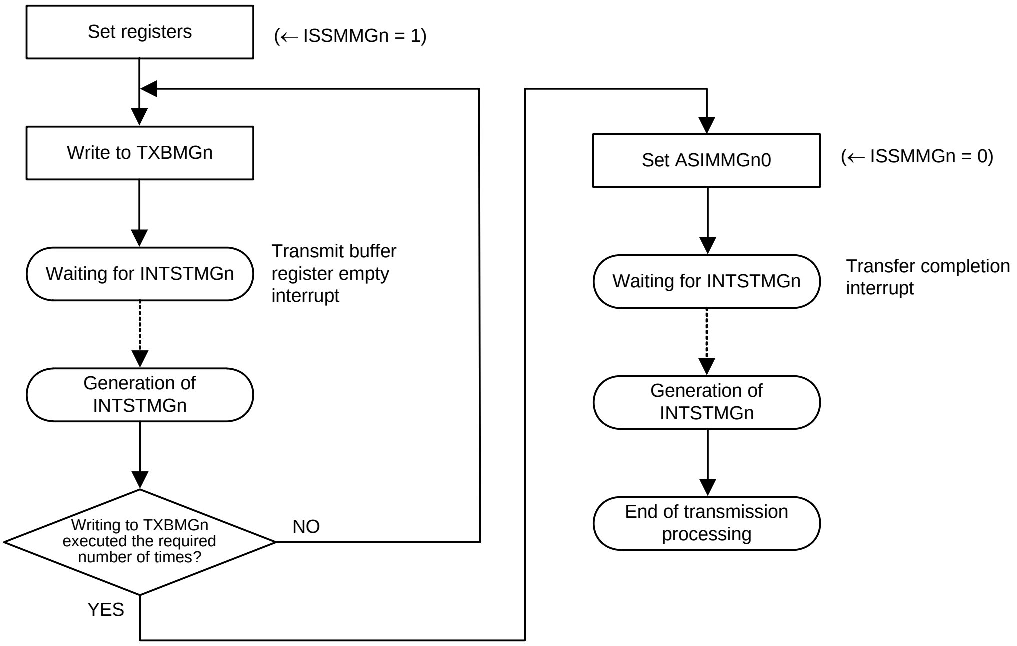
When the BRGCMGn register = 02H (k = 2), start bit = 1 bit, character length = 8 bits, parity used, and stop bit = 1 bit:

The maximum number of clock cycles = Transfer length × 2k - (2k + 3) = 11 × 2 × 2 - (2 × 2 + 3) = 37

(Writing must be completed within 37 clock cycles.)

Figure 22-17 shows a flow example of continuous transmission using interrupt.

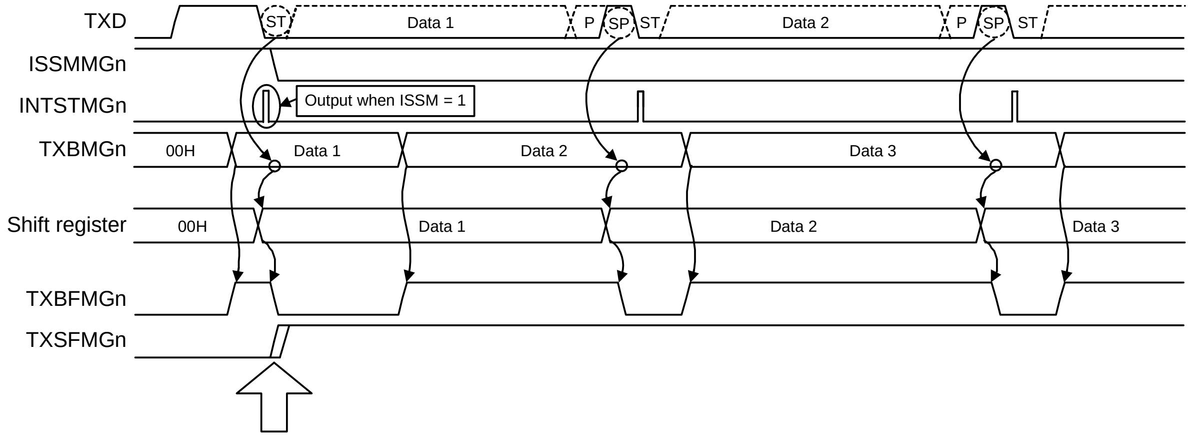
Figure 22-17. Flow Example of Continuous Transmission Using Interrupt



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

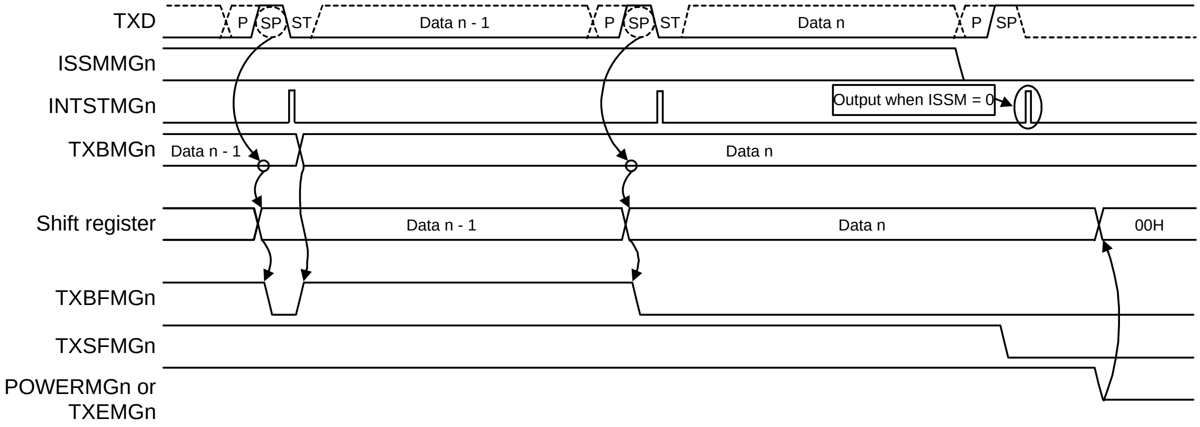
Figure 22-18 and **Figure 22-19** show the timing charts when the continuous transmission is started and completed, respectively.

Figure 22-18. Timing Chart When Continuous Transmission Is Started



Caution **When the ASISMGn register is read, both the TXBFMGn and TXSFMGn bits are read as 1 within this period. Accordingly, use only the TXBFMGn flag to determine if writing is enabled or disabled.**

Figure 22-19. Timing Chart When Continuous Transmission is Completed



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

(6) Normal reception

When setting bit 7 (POWERMGn) of the operation mode register 0 (ASIMMGn0) to 1 and then setting bit 5 (RXEMGn) of the ASIMMGn0 register to 1, reception is enabled, and sampling of the input to the RXDMGn pin is performed.

When the ALVn bit is 0, the 8-bit counter of the baud rate generator starts counting on detection of the falling edge on the RXDMGn pin. When the counter reaches the set value of the baud rate generator control register (BRGCMGn), the input to the RXDMGn pin is sampled again (at the point indicated with ▽ in **Figure 22-20**). If the RXDMGn 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 RXDMGn pin. When the counter reaches the set value of the baud rate generator control register (BRGCMGn), the input to the RXDMGn pin is sampled again (at the point indicated with ▽ in **Figure 22-20**). If the RXDMGn pin is high, it is regarded as a start bit.

Figure 22-20 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 reception completion interrupt (INTSRMGn) is generated, and at the same time, the data in the receive shift register is written to the receive buffer register (RXBMGn).

Note that when an overrun error (OVEMGn) occurs, the data received on occurrence of the error is not written to the RXBMGn register.

When a parity error (PEMGn) or a framing error (FEMGn) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (INTSRMGn/INTSREMGn) set in ISRMMGn is generated.

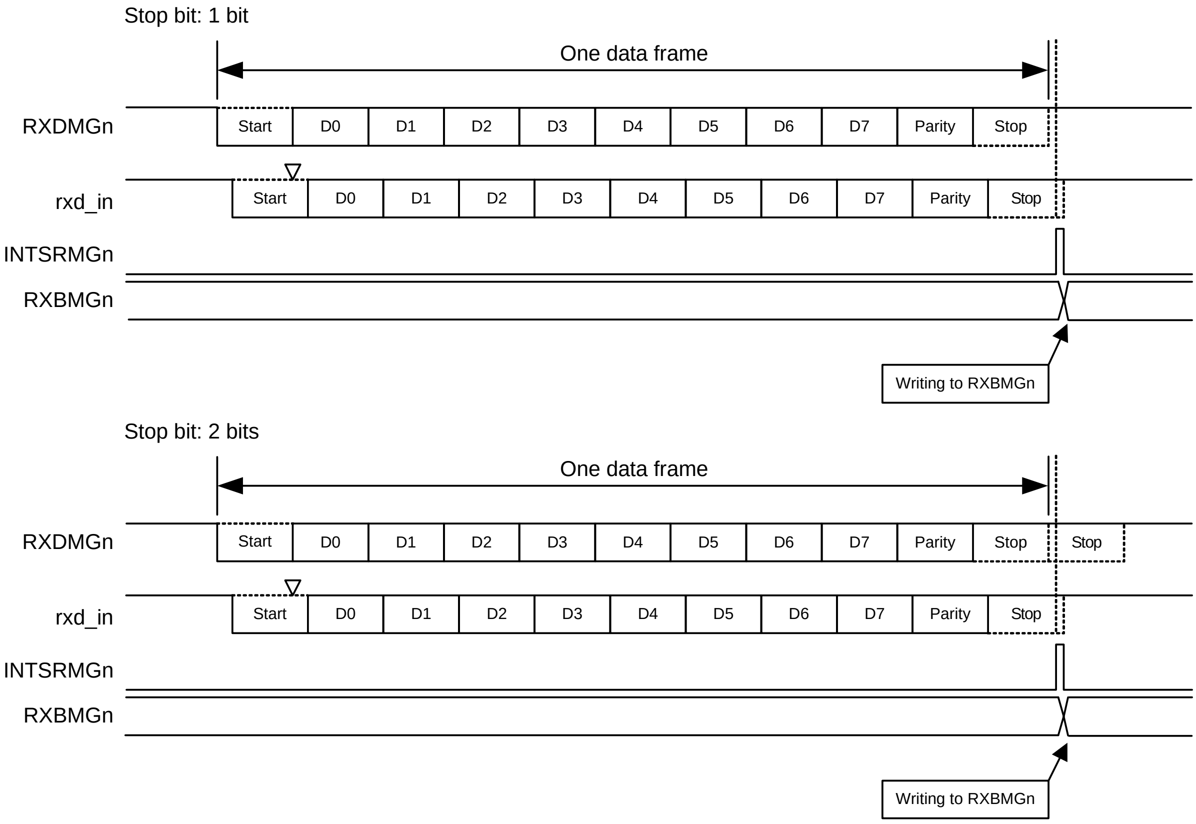
When a reception error occurs, read the status register (ASISMGn) and then read the receive buffer register (RXBMGn) to clear the error flag.

If the receive buffer register (RXBMGn) 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 1 stop bit. Accordingly, the second stop bit is ignored.

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

Figure 22-20. Timing of UART Receive Operation



Remarks 1. rxd_in: the internal signal generated by latching RXDMGn with a noise filter (rxd_in is delayed relative to RXDMGn by maximum of 3 cycles of the UART operation clock.)

2. The INTSRMGn output timing in the figure is just an example. The timing relative to RXDMGn varies according to the setting of the BRGCMGn register.

3. n: Channel number (n = 0, 1)

(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 (ASISMGn) is set, and the reception error interrupt (INTSRMGn or INTSREMGn) 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 (ASISMGn).

The contents of the ASISMGn register is cleared to 0 by setting the corresponding bit of the status clear trigger register (ASCTMGn) to 1.

Table 22-4 shows the causes of the reception errors.

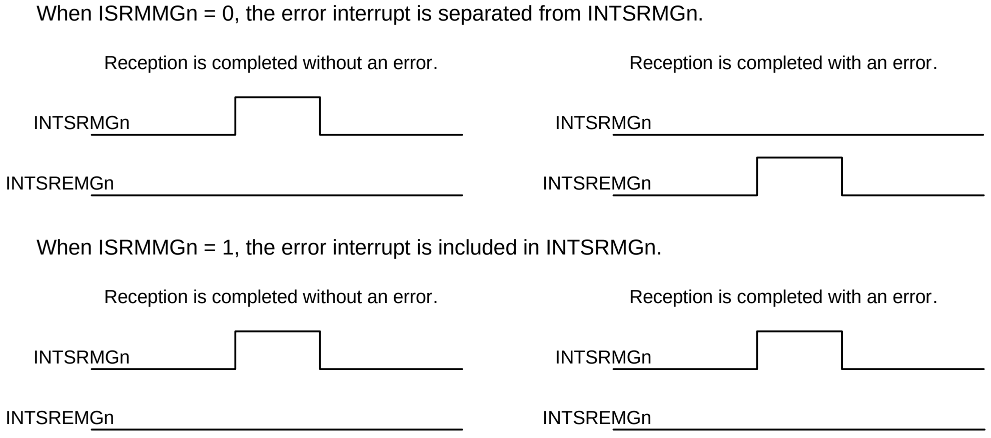
Table 22-4. Causes of Reception Errors

Error flag	Reception error	Cause
PEMGn	Parity error	The parity specified for reception does not match the parity of receive data.
FEMGn	Framing error	No stop bit is detected.
OVEMGn	Overrun error	OVEMGn Before the receive data is read from the receive buffer, the next data reception is completed.

Setting bit 0 (ISRMMGn) of the operation mode register 0 (ASIMMGn0) to 0 allows the reception error interrupt to be separated from the reception completion interrupt (INTSRMGn) and allows it to be generated as the error interrupt (INTSREMGn).

Figure 22-21 shows the interrupt output waveform which varies depending on the setting of ISRMMGn.

Figure 22-21. Various Interrupt Output Waveforms Depending on ISRMMGn Setting

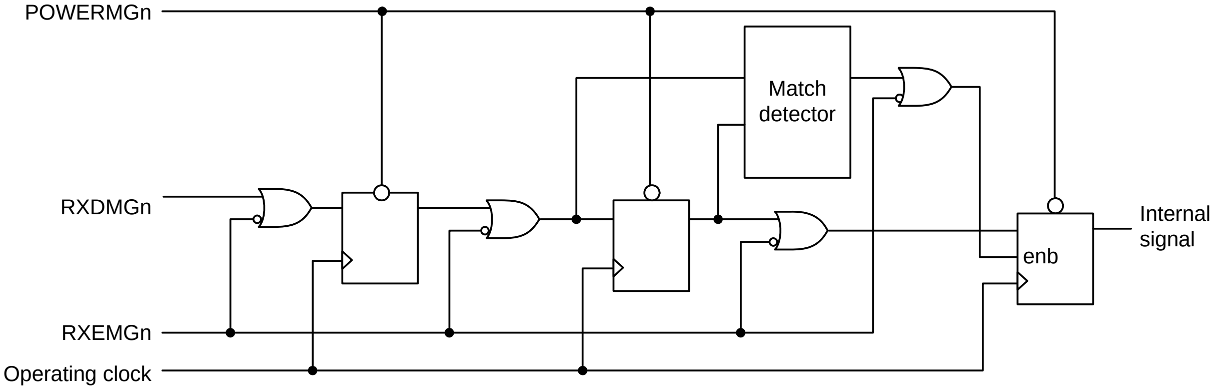

When ISRMMGn = 0, the error interrupt is separated from INTSRMGn.
Reception is completed without an error.
Reception is completed with an error.
INTSRMGn
INTSRMGn
INTSREMGn
INTSREMGn
When ISRMMGn = 1, the error interrupt is included in INTSRMGn.
Reception is completed without an error.
Reception is completed with an error.
INTSRMGn
INTSRMGn
INTSREMGn
INTSREMGn

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

22.3.3 Receive Data Noise Filter

This filter samples the receive data (RXDMGn), 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 22-22 shows the noise filter circuit.

Figure 22-22. Noise Filter



Cautions 1. When ALVn = 0 (wait state = high level; start bit = low level), the initial value of the receive data (RXDMGn) needs to be "high".

2. When ALVn = 1 (wait state = low level; start bit = high level), the initial value of the receive data (RXDMGn) needs to be "low".

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

22.3.4 Baud Rate Generator

The baud rate generator consists of 8-bit programmable counters, and generates a serial clock for transmission/reception of UARTMGn.

An 8-bit counter is provided each for transmission and reception.

(1) Configuration of baud rate generator

(a) Base clock

When bit 7 (POWERMGn) = 1 in the operation mode register 0 (ASIMMGn0), the operating clock (f_{SX} or f_{SX} × 2) of UARTMGn is supplied to each module. This clock is called the base clock. When POWERMGn = 0, the base clock is fixed to low level.

(b) Transmission counter

This counter is cleared to 0 and stops when bit 7 (POWERMGn) = 0 or bit 6 (TXEMGn) = 0 in the operation mode register 0 (ASIMMGn0). It starts counting when POWERMGn = 1 and TXEMGn = 1.

The counter is cleared to 0 when the first transmit data is written to the transmit buffer register (TXBMGn).

When continuous transmission is performed, the counter is cleared to 0 again when transmission of one frame of data has been completed. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until the POWERMGn or TXEMGn bit is cleared to 0. When POWERMGn = 0 or TXEMGn =0 in the ASIMMGn0 register, the counter stops at 00H.

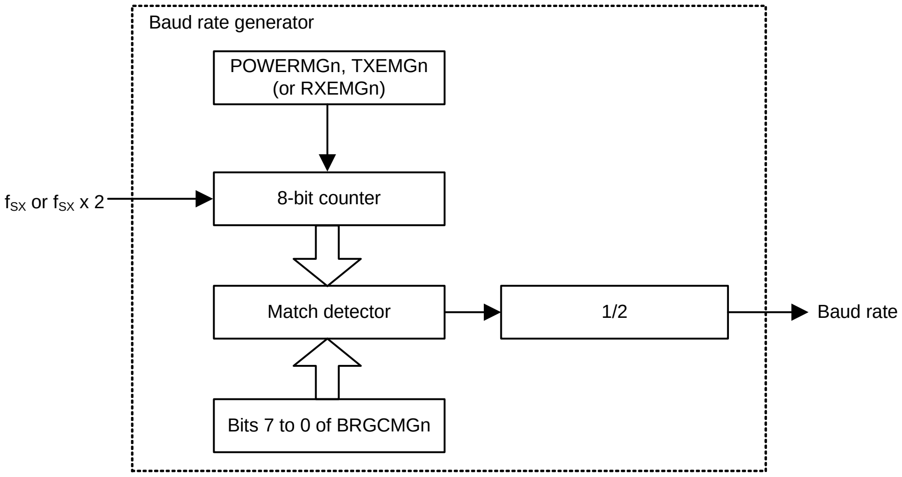
(c) Reception counter

This counter is cleared to 0 and stops when bit 7 (POWERMGn) = 0 or bit 5 (RXEMGn) = 0 in the operation mode register 0 (ASIMMGn0). 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 POWERMGn = 0 or RXEMGn = 0 in the ASIMMGn0 register, the counter stops at 00H.

Figure 22-23 shows the configuration of the baud rate generator.

Figure 22-23. Configuration of Baud Rate Generator



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

(2) Generation of serial clock

A serial clock to be generated can be specified by using the baud rate generator control register (BRGCMGn). The frequency divisor for the 8-bit counter can be selected as a value in the range from $f_{SX}/2$ to $f_{SX}/255$ by the setting of bits 7 to 0 in BRGCMGn.

(3) Baud rate calculation

(a) Baud rate calculation expression

The baud rate can be calculated by the following expression.

Baud rate = $f_{SX} \div (2 \times k)$ [bps]

f_{SX}: Frequency of operating clock

k: Value set by bits 7 to 0 of the BRGCMGn register (k = 2, 3, 4, …, 255)

(b) Baud rate error

The baud rate error can be calculated by the following expression.

$$\text{Error} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100\ [\%]$$

Cautions 1. Keep the baud rate error during transmission to within the permissible error range on the reception side.

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 22.3.4 (3) (d) Permissible baud rate range during reception.

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

(c) Baud rate setting example

Table 22-5. Set Data of Baud Rate Generator

Desired baud rate	In operation with f_{SX} = 38.4 kHz				In operation with f_{SX} = 32.768 kHz	
	CLKDENn = 0		CLKDENn = 1		CLKDENn = 0	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	96	±0.00%	192	±0.00%	82	-0.10%
300 bps	64	±0.00%	128	±0.00%	55	-0.70%
1200 bps	16	±0.00%	32	±0.00%	14	-2.48%
2400 bps	8	±0.00%	16	±0.00%	7	-2.48%
4800 bps	4	±0.00%	8	±0.00%	Disabled	
9600 bps	2	±0.00%	4	±0.00%	Disabled	
19200 bps	Disabled		2	±0.00%	Disabled	

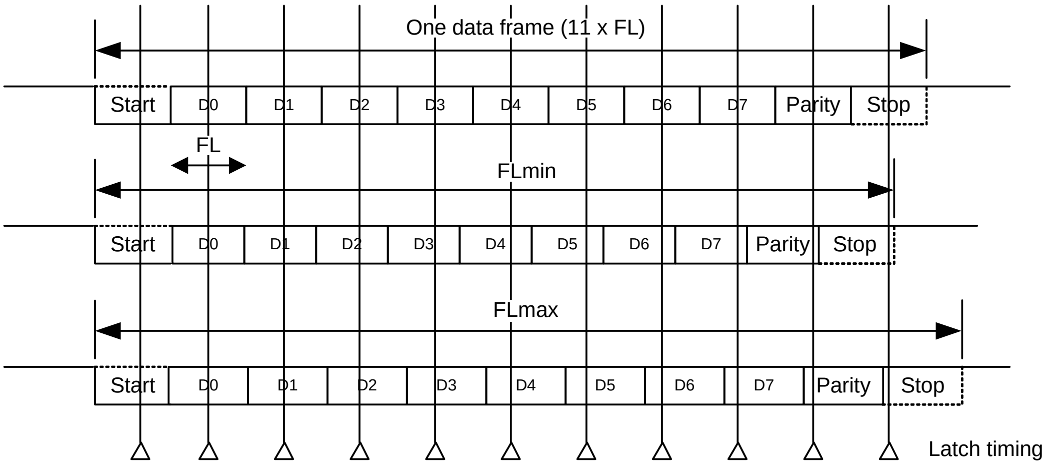
Remark k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCMGn) (k = 2, 3, 4, …, 255)

(d) Permissible baud rate range during reception

Figure 22-24 shows the permissible error from the baud rate on the transmitting side during reception.

Figure 22-24. Permissible Baud Rate Range during Reception

Data length: 8 bits, with parity, stop bit: 1 bit



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 (BRGCMGn). 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.

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

- The relation between 1-bit data length and baud rate

  FL = (Brate) - 1

  Brate: Baud rate of UART

  K: Set value of BRGCMGn

  FL: 1-bit data length

  Margin of latch timing: 1 clock

- Minimum permissible data frame length (FLmin)

$$\text{FLmin} = 11 \times \text{FL} - \frac{k - 1}{2k} \times \text{FL} = \frac{21k + 1}{2k}\ \text{FL}$$

- Maximum permissible baud rate for reception on the transmitting side (BRmax)

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k + 1}\ \text{Brate}$$

- Maximum permissible data frame length (FLmax)

$$\text{FLmax} = \frac{21k + 1}{20k}\ \text{FL} \times 11$$

- Minimum permissible baud rate for reception on the transmitting side (BRmin)

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k - 1}\ \text{Brate}$$

The permissible baud rate error between UART and the transmitting side can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 22-6. Maximum/Minimum Permissible Baud Rate Error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.32%	-2.43%
4	+3.52%	-3.61%
8	+4.14%	-4.19%
20	+4.51%	-4.53%
50	+4.66%	-4.67%
100	+4.71%	-4.71%
255	+4.74%	-4.74%

Remarks 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.

2. k: Set value of BRGCMGn

3. n: Channel number (n = 0, 1)

22.4 Usage Notes

22.4.1 Port Setting for RXDMGn Pin

When ALVn = 0 (wait state = high level, start bit = low level), the initial value of receive data (RXDMGn) must be high.
When ALVn = 1 (wait state = low level, start bit = high level), the initial value of receive data (RXDMGn) must be low.
Accordingly, port setting is required for the RXDMGn pin before setting POWERMGn = 1.

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

CHAPTER 23 IrDA

The IrDA sends and receives IrDA data communication waveforms in cooperation with the Serial Array Unit (SAU) based on the IrDA (Infrared Data Association) standard 1.0.

23.1 Functions of IrDA

Enabling the IrDA function by using the IRE bit in the IRCR register allows encoding and decoding the TxD2 and RxD2 signals of the SAU to the waveforms conforming to the IrDA standard 1.0 (IrTxD and IrRxD pins). Connecting these waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Since the IrDA cannot change the transfer rate automatically, the transfer rate should be changed through software.

When the high-speed on-chip oscillator (f_{IH} = 32/24/12/6/3 MHz) is selected, the following baud rates can be selected:

- 115.2 kbps/57.6 kbps/38.4 kbps/19.2 kbps/9600 bps/2400 bps

Figure 23-1 is a block diagram showing cooperation between IrDA and SAU.

Figure 23-1. Block Diagram Showing Cooperation Between IrDA and SAU

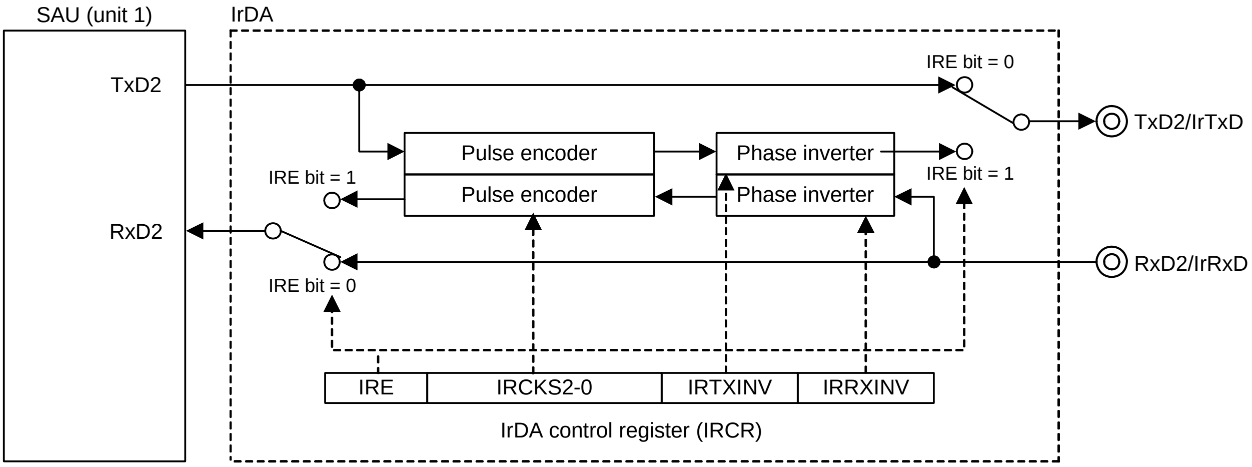


Table 23-1. IrDA Pin Configuration

Pin Name	I/O	Function
IrTxD	Output	Outputs data to be transmitted.
IrRxD	Input	Inputs received data.

23.2 Registers

Table 23-2 lists the IrDA register configuration.

Table 23-2. IrDA Register Configuration

Item	Configuration
Control registers	Peripheral enable register 0 (PER0)
	Peripheral reset control register 0 (PRR0)
	IrDA control register (IRCR)

23.2.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the IrDA is to be used, be sure to set bit 6 (IRDAEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 23-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	0	IRDAEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	SAU2EN	TAU0EN

IRDAEN	Control of IrDA input clock supply
0	Stops input clock supply. ● SFR used by IrDA cannot be written. The read value is 00H. However, the SFR is not initialized. Note
1	Enables input clock supply. ● SFR used by IrDA can be read and written.

Note To initialize the IrDA and the SFR used by the IrDA, use bit 6 (IRDARES) of PRR0.

Cautions 1. When setting the IrDA, be sure to set the IRDAEN bit to 1 first.
If IRDAEN = 0, writing to a control register of the IrDA is ignored, and read value of the register is all the initial value.

2. Be sure to clear bit 7 to "0".

23.2.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules.

Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module.

To place the IrDA in the reset state, be sure to set bit 6 (IRDARES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears the PRR0 register to 00H.

Figure 23-3. Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

IRDARES	Control resetting of the IrDA
0	The IrDA is released from the reset state.
1	The IrDA is in the reset state.

23.2.3 IrDA control register (IRCR)

The IRCR register is used to control the IrDA function. This register is used to switch the polarity of receive data and transmit data, select the IrDA clock, and select the serial I/O pin function (normal serial function or IrDA function).

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

Reset signal generation clears this register to 00H.

Figure 23-4. Format of IrDA Control Register (IRCR)

Address: F03A0H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
IRCR	IRE	IRCKS2	IRCKS2	IRCKS0	IRTXINV	IRRXINV	0	0

IRE	IrDA enable
0	Serial I/O pins are used for normal serial communication.
1	Serial I/O pins are used for IrDA data communication.

IRCKS2	IRCKS1	IRCKS0	IrDA clock selection
0	0	0	B × 3/16 (B = bit rate)
0	0	1	f_{CLK}/2
0	1	0	f_{CLK}/4
0	1	1	f_{CLK}/8
1	0	0	f_{CLK}/16
1	0	1	f_{CLK}/32
1	1	0	f_{CLK}/64
1	1	1	Setting prohibited

IRTXINV	IrTxD data polarity switching
0	Data to be transmitted is output to IrTxD as is.
1	Data to be transmitted is output to IrTxD after the polarity is inverted.

IRRXINV	IrRxD data polarity switching
0	IrRxD input is used as received data as is.
1	IrRxD input is used as received data after the polarity is inverted.

Cautions 1. Be sure to clear bits 1 and 0 to “0”.

2. IRCKS[2:0], IRTXINV, and IRRXINV can be set only when IRE bit is 0.

23.3 Operation

23.3.1 IrDA communication operation procedure

(1) IrDA Communication Initial configuration flow
Perform IrDA initial configuration as follows:

<1> Set PER0 register bit IRDAEN to 1.
<2> Set the IRCR register.
<3> Set the SAU related registers (refer to the UART mode configuration procedure).

(2) IrDA communication termination flow

<1> Configure the port register and port mode register to set the status of the IrTxD pin after stopping IrDA communication.

Remark The output status may change because the IrTxD pin changes to normal serial interface UART data output when IrDA is reset in step 3.

- To output low level from IrTxD pin
  Set port register to 0. Immediately after this, the IrTxD pin is fixed at low level.
- To output high level from IrTxD pin
  Set port register to 1. This will fix IrTxD pin at high level immediately after IrDA reset in step 3.
- To set IrTxD pin to Hi-Z status
  Set port mode register to 1. Immediately after this, IrTxD pin is set to Hi-Z.

<2> Set STm register (SAU related register) bits STm0 and STm1 to 1 (stop SAU channels 0 and 1).
<3> Set PER0 register bit IRDAEN to 0 and reset IrDA.

Do not set STm register bits STm0 and STm1 to 1 or IrDA bit IRE to 0 with any procedure other than the above.

(3) Procedure when IrDA framing error occurs
If a framing error occurs during IrDA communication, the following procedure is necessary to enable receiving of subsequent data.

<1> Set SAU STm register bit STm1 to 1 (stop SAU CH1 operation)
<2> Set SAU SSm register bit SSm1 to 1 (start SAU CH1 operation)

Remark m: Unit number (m = 0, 1)

Also refer to the chapter on SAU for information on SAU framing error processing.

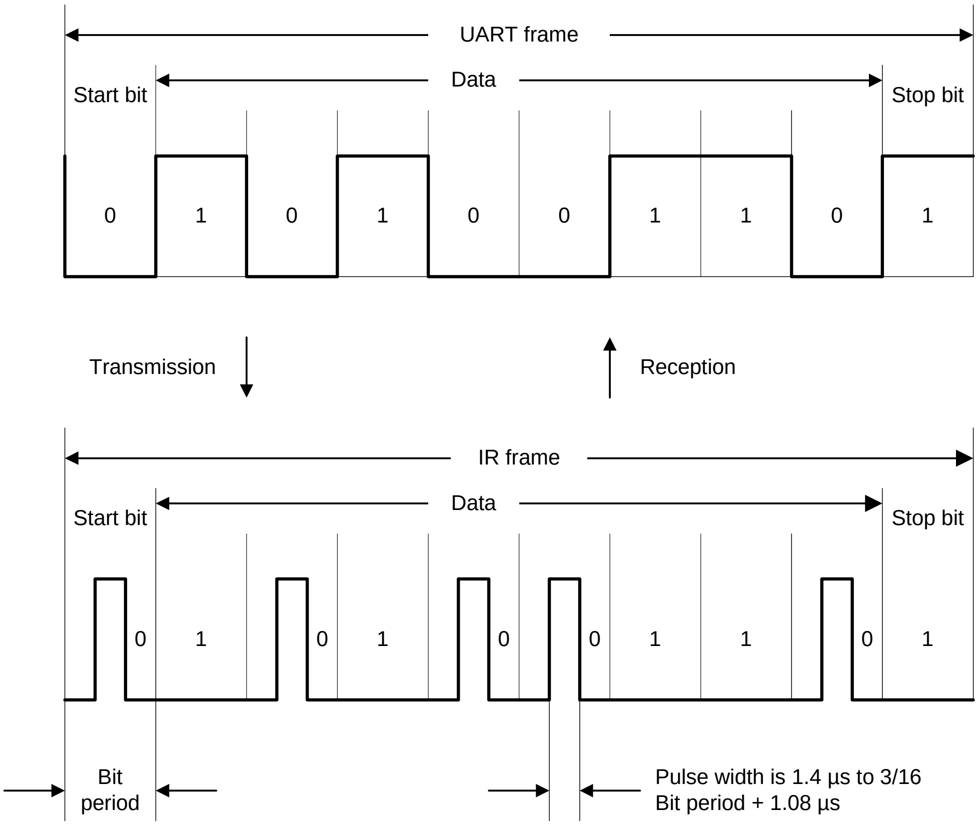
23.3.2 Transmission

In transmission, the signals output from the SAU (UART frames) are converted to the IR frame data through the IrDA (see **Figure 23-5**). When IRTXINV bit is 0 and serial data is 0, high-level pulses with the width of 3/16 the bit rate (1-bit width period) are output (initial setting). The high-level pulse width can be changed by using the IRCKS2 to IRCKS0 bits. The standard prescribes that the minimum high-level pulse width should be 1.41 µs and the maximum high-level pulse width be (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs.

When the CPU/peripheral hardware clock (f_{CLK}) is 20 MHz, the high-level pulse width can be 1.41 µs to 1.6 µs.

When serial data is 1, no pulses are output.

Figure 23-5. IrDA Transmission/Reception



23.3.3 Reception

In reception, the IR frame data is converted to the UART frame data through the IrDA and is input to the SAU.

Low-level data is output when the IRRXINV bit is 0 and a high-level pulse is detected, and high-level data is output when no pulse is detected for 1-bit period. Note that a pulse shorter than 1.41 µs, which is the minimum pulse width, is identified as a low signal.

23.3.4 Selecting high-level pulse width

When the pulse width should be shorter than the bit rate × 3/16 for transmission, applicable IRCKS2 to IRCKS0 bit settings (minimum pulse width) and the corresponding high-level pulse widths shown in **Table 23-4** can be used.

Table 23-4. IRCKS2 to IRCKS0 Bit Settings

f_{CLK} [MHz]	Item	<Upper Row> Bit Rate [kbps] <Lower Row> Bit Rate × 3/16 [µs]					
		2.4	9.6	19.2	38.4	57.6	115.2
		78.13	19.53	9.77	4.87	3.26	1.63
1	IRCKS2 to IRCKS0	001	001	001	–[Note 1]	–[Note 1]	–[Note 1]
	High-level pulse width [µs]	2.00	2.00	2.00	–[Note 1]	–[Note 1]	–[Note 1]
2	IRCKS2 to IRCKS0	010	010	010	010	010	–[Note 1]
	High-level pulse width [µs]	2.00	2.00	2.00	2.00	2.00	–[Note 1]
3	IRCKS2 to IRCKS0	011	011	011	011	011	–[Note 1]
	High-level pulse width [µs]	2.67	2.67	2.67	2.67	2.67	–[Note 1]
4	IRCKS2 to IRCKS0	011	011	011	011	011	000[Note 2]
	High-level pulse width [µs]	2.00	2.00	2.00	2.00	2.00	1.50
6	IRCKS2 to IRCKS0	100	100	100	100	100	000[Note 2]
	High-level pulse width [µs]	2.67	2.67	2.67	2.67	2.67	1.50
8	IRCKS2 to IRCKS0	100	100	100	100	100	000[Note 2]
	High-level pulse width [µs]	2.00	2.00	2.00	2.00	2.00	1.50
12	IRCKS2 to IRCKS0	101	101	101	101	101	000[Note 2]
	High-level pulse width [µs]	2.67	2.67	2.67	2.67	2.67	1.50
16	IRCKS2 to IRCKS0	101	101	101	101	101	000[Note 2]
	High-level pulse width [µs]	2.00	2.00	2.00	2.00	2.00	1.50
24	IRCKS2 to IRCKS0	110	110	110	110	110	000[Note 2]
	High-level pulse width [µs]	2.67	2.67	2.67	2.67	2.67	1.50
32	IRCKS2 to IRCKS0	110	110	110	110	110	000[Note 2]
	High-level pulse width [µs]	2.00	2.00	2.00	2.00	2.00	1.50

Notes 1. "–" indicates that the communication specification cannot be satisfied.

2. The pulse width cannot be shorter than the bit rate × 3/16.

23.4 Usage Notes on IrDA

(1) The IrDA function cannot be used to transition to SNOOZE via IrRxD reception.

(2) The input of IrDA operating clock can be disabled/enabled with the peripheral enable register. Initially, register access is disabled because clock input is disabled. Enable IrDA operating clock input with the peripheral enable register before setting the register.

(3) During HALT mode, the IrDA function continues to run.

(4) The use of SAU initialization function (SS bit= 1) is prohibited during IrDA communication.

(5) The IRCR register bits IRRXINV, IRTXINV, and IRCKS[2:0] can be set only when IRE bit is 0.

CHAPTER 24 LCD CONTROLLER/DRIVER

The number of LCD display function pins of the RL78/I1C (512 KB) differs depending on the product. The following table shows the number of pins of each product.

Table 24-1. Number of LCD Display Function Pins of Each Product

Item		RL78/I1C (512 KB)															
		80 pins (R5F10NML)								100 pins (R5F10NPL)							
LCD controller/driver		Segment signal outputs: 34 (30) Note 1 Common signal outputs: 8								Segment signal outputs: 42 (38) Note 1 Common signal outputs: 8							
Multiplexed I/O port Note 2		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
Segment	P0	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32	–	–	–	–	–	–	–	–	–	–
	P1	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4	SEG11	SEG10	SEG9	SEG8	SEG7	SEG6	SEG5	SEG4
	P3	–	–	–	–	SEG27	SEG26	SEG25	SEG24	SEG31	SEG30	SEG29	SEG28	SEG27	SEG26	SEG25	SEG24
	P5	–	–	–	–	–	–	–	–	SEG39	SEG38	SEG37	SEG36	SEG35	SEG34	SEG33	SEG32
	P7	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16	SEG23	SEG22	SEG21	SEG20	SEG19	SEG18	SEG17	SEG16
	P8	–	–	–	–	SEG15	SEG14	SEG13	SEG12	–	–	SEG41	SEG40	SEG15	SEG14	SEG13	SEG12
Alternate relationship between COM signal output pins and I/O pots	P9	COM7/SEG3	COM6/SEG2	COM5/SEG1	COM4/SEG0	COM3	COM2	COM1	COM0	COM7/SEG3	COM6/SEG2	COM5/SEG1	COM4/SEG0	COM3	COM2	COM1	COM0

Notes **1.** ( ) indicates the number of signal output pins when 8 com is used.

2. For the LCD display function pins, the number and placement depend on the product and the setting of peripheral redirection register 0 (PIOR0).
For details, see **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

24.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the RL78/I1C (512 KB) 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 can be selected from 16 steps (contrast adjustment).
(5) LCD blinking is available

Table 24-2 lists the maximum number of pixels that can be displayed in each display mode.

Table 24-2. Maximum Number of Pixels (1/2)

(a) 80-pin product

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	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)
	Capacitor split	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)
Waveform B	External resistance division, internal voltage boosting	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	
	Capacitor split	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 24-2. Maximum Number of Pixels (2/2)

(b) 100-pin product

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	42 (42 segment signals, 1 common signal)
		1/2	2	84 (42 segment signals, 2 common signals)
			3	126 (42 segment signals, 3 common signals)
		1/3	3	
			4	168 (42 segment signals, 4 common signals)
			6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)
		1/4	8	
	Internal voltage boosting	1/3	3	126 (42 segment signals, 3 common signals)
			4	168 (42 segment signals, 4 common signals)
			6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)
		1/4	6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)
	Capacitor split	1/3	3	126 (42 segment signals, 3 common signals)
			4	168 (42 segment signals, 4 common signals)
			6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	3	126 (42 segment signals, 3 common signals)
			4	168 (42 segment signals, 4 common signals)
			6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)
		1/4	8	
	Capacitor split	1/3	3	126 (42 segment signals, 3 common signals)
			4	168 (42 segment signals, 4 common signals)
			6	240 (40 segment signals, 6 common signals)
			8	304 (38 segment signals, 8 common signals)

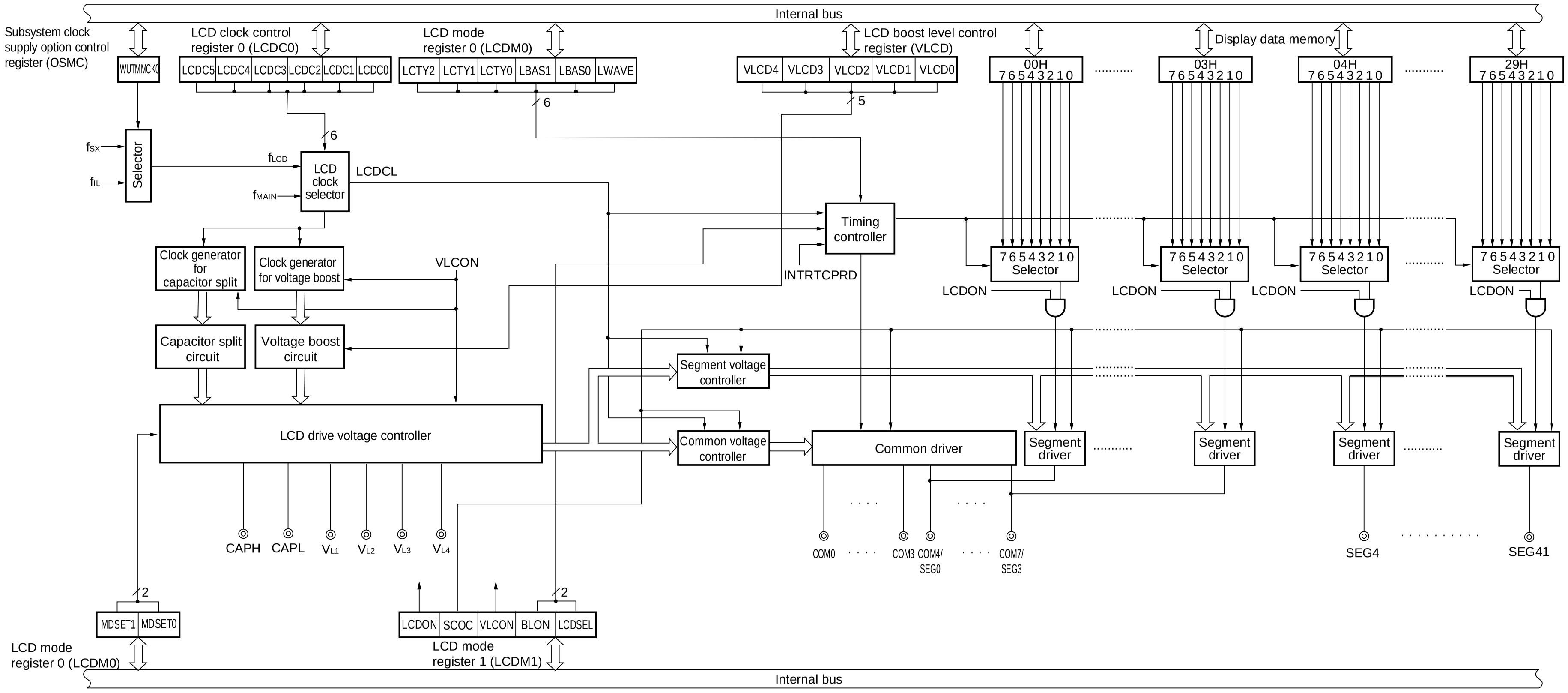
24.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 24-3. Configuration of LCD Controller/Driver

Item	Configuration
Control registers	LCD mode register 0 (LCDM0) LCD mode register 1 (LCDM1) Subsystem clock supply option control register (OSMC) LCD clock control register 0 (LCDC0) LCD boost level control register (VLCD) LCD input switch control register (ISCLCD) LCD port function registers 0 to 6 (PFSEG0 to PFSEG6) Port mode registers 0, 1, 3, 5, 7, 8, 9 (PM0, PM1, PM3, PM5, PM7, PM8, PM9)

Figure 24-1. Block Diagram of LCD Controller/Driver



24.3 Registers Controlling LCD Controller/Driver

The following ten registers are used to control the LCD controller/driver.

- LCD mode register 0 (LCDM0)
- LCD mode register 1 (LCDM1)
- Subsystem clock supply option control register (OSMC)
- LCD clock control register 0 (LCDC0)
- LCD boost level control register (VLCD)
- LCD input switch control register (ISCLCD)
- LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)
- Port mode registers 0, 1, 3, 5, 7, 8, 9 (PM0, PM1, PM3, PM5, PM7, PM8, PM9)

24.3.1 LCD mode register 0 (LCDM0)

LCDM0 specifies the LCD operation.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDM0 to 00H.

Figure 24-2. Format of LCD Mode Register 0 (LCDM0) (1/2)

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

MDSET1	MDSET0	LCD drive voltage generator selection
0	0	External resistance division method
0	1	Internal voltage boosting method
1	0	Capacitor split method
1	1	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

Figure 24-2. Format of LCD Mode Register 0 (LCDM0) (2/2)

Address: FFF40H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LCDM0	MDSET1	MDSET0	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0

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

Cautions 1. Do not rewrite the LCDM0 value while the SCOC bit of the LCDM1 register = 1.

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.

3. Only the combinations of display waveform, number of time slices, and bias method shown in Table 24-4 are supported.

Combinations of settings not shown in Table 24-4 are prohibited.

Table 24-4. Combinations of Display Waveform, Time Slices, Bias Method, and Frame Frequency

Display Mode			Set Value						Driving Voltage Generation Method		
Display Waveform	Number of Time Slices	Bias Mode	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0	External Resistance Division	Internal Voltage Boosting	Capacitor Split
Waveform A	8	1/4	0	1	0	1	1	0	O (24 to 128 Hz)	O (24 to 64 Hz)	×
Waveform A	6	1/4	0	1	0	0	1	0	×	O (32 to 86 Hz)	×
Waveform A	8	1/3	0	1	0	1	0	1	O (32 to 128 Hz)	O (32 to 64 Hz)	O (32 to 128 Hz)
Waveform A	6	1/3	0	1	0	0	0	1	O (32 to 128 Hz)	O (32 to 86 Hz)	O (32 to 128 Hz)
Waveform A	4	1/3	0	0	1	1	0	1	O (24 to 128 Hz)	O (24 to 128 Hz)	O (24 to 128 Hz)
Waveform A	3	1/3	0	0	1	0	0	1	O (32 to 128 Hz)	O (32 to 128 Hz)	O (32 to 128 Hz)
Waveform A	3	1/2	0	0	1	0	0	0	O (32 to 128 Hz)	×	×
Waveform A	2	1/2	0	0	0	1	0	0	O (24 to 128 Hz)	×	×
Waveform A	Static		0	0	0	0	0	0	O (24 to 128 Hz)	×	×
Waveform B	8	1/4	1	1	0	1	1	0	O (24 to 128 Hz)	O (24 to 64 Hz)	×
Waveform B	8	1/3	1	1	0	1	0	1	O (32 to 128 Hz)	O (32 to 64 Hz)	O (32 to 128 Hz)
Waveform B	6	1/3	1	1	0	0	0	1	O (32 to 128 Hz)	O (32 to 86 Hz)	O (32 to 128 Hz)
Waveform B	4	1/3	1	0	1	1	0	1	O (24 to 128 Hz)	O (24 to 128 Hz)	O (24 to 128 Hz)
Waveform B	3	1/3	1	0	1	0	0	1	O (32 to 128 Hz)	O (32 to 128 Hz)	O (32 to 128 Hz)

Remark O: Supported
×: Not supported

24.3.2 LCD mode register 1 (LCDM1)

LCDM1 enables or disables display operation, voltage boost circuit operation, and capacitor split circuit operation, and specifies the display data area and the low voltage mode.

LCDM1 is set using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets LCDM1 to 00H.

Figure 24-3. Format of LCD Mode Register 1 (LCDM1) (1/2)

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

SCOC	LCDON	LCD display enable/disable
		When normal liquid crystal waveform (waveform A or B) is output
0	0	Output ground level to segment/common pin
0	1	
1	0	Display off (all segment outputs are deselected.)
1	1	Display on

VLCON[Note 1]	Voltage boost circuit or capacitor split circuit operation enable/disable
0	Stops voltage boost circuit or capacitor split circuit operation
1	Enables voltage boost circuit or capacitor split circuit operation

BLON[Note 2]	LCDSEL	Display data area control
0	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	0	Alternately displaying A-pattern and B-pattern area data (blinking display corresponding to the fixed-cycle interrupt (INTRTCPRD) timing of the independent power supply RTC)
1	1	

Notes 1. Cannot be set during external resistance division mode.

2. When f_{IL} is selected as the LCD source clock (f_{LCD}), be sure to set the BLON bit to “0”.

(**Cautions** are listed on the next page.)

Figure 24-3. Format of LCD Mode Register 1 (LCDM1) (2/2)

Address: FFF41H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
LCDM1	LCDON	SCOC	VLCON	BLON	LCDSEL	0	0	LCDVLM

LCDVLM[Note]	Control of default value of voltage boosting pin
0	Set when $V_{DD} \geq 2.7$ V
1	Set when $V_{DD} \leq 4.2$ V

Note A function to set the initial state of the V_{LX} pin and efficiently boost voltage when using a voltage boosting circuit. Set LCDVLM bit = 0 when V_{DD} at the start of voltage boosting is 2.7 V or more. Set LCDVLM bit = 1 when V_{DD} is 4.2 V or less.
However, when $2.7 \text{ V} \leq V_{DD} \leq 4.2$ V, operation is possible with LCDVLM = 0 or LCDVLM = 1.

Cautions 1. When the voltage boost circuit is used, set SCOC = 0 and VLCON = 0, and MDSET1, MDSET0 = 00 in order to reduce power consumption when the LCD is not used. When MDSET1, MDSET0 = 01, power is consumed by the internal reference voltage generator.

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.

3. Do not rewrite the VLCON and LCDVLM bits while SCOC = 1.

4. Set the BLON and LCDSEL bits to 0 when 8 has been selected as the number of time slices for the display mode.

5. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set the VLCON bit to 1.

24.3.3 Subsystem clock supply option control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 is stopped in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock.

In addition, the OSMC register can be used to select the operating clock of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

The low-speed on-chip oscillator clock cannot be selected as the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1. When the serial interface UARTMG0 or UARTMG1, sampling output timer detector 0, or sampling output timer detector 1 is to be used, select the sub clock (f_{SX}) as the operating clock by setting the WUTMMCK0 bit to 0.

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

Reset signal generation clears this register to 00H.

Figure 24-4. Format of Subsystem clock supply option control register (OSMC)

Address: F00F3H After reset: 00H R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC Note 4	Setting in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock
0	Enables supply of sub clock (f_{SX}) to peripheral functions (See **Tables 29-1** to **29-3** for peripheral functions whose operations are enabled.)
1	Stops supply of sub clock (f_{SX}) to peripheral functions other than the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL}) Notes 2, 3, 6, 7	Low-speed on-chip oscillator clock (f_{IL}) selected Note 6	Clock output is prohibited. Note 5	Setting prohibited

(**Notes** are listed on the next page.)

Notes **1**. Be sure to set bits 0 to 3, 5, and 6 to “0”.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (f_{IL}) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the sub clock (f_{SX}) 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.

5. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

6. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

7. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

24.3.4 LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD source clock and LCD clock.

The frame frequency is determined according to the LCD clock and the number of time slices.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets LCDC0 to 00H.

Figure 24-5. Format of LCD Clock Control Register 0 (LCDC0)

Address: FFF42H After reset: 00H 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	$f_{SX}/2^2$	$f_{IL}/2^2$
0	0	0	0	1	0	$f_{SX}/2^3$	$f_{IL}/2^3$
0	0	0	0	1	1	$f_{SX}/2^4$	$f_{IL}/2^4$
0	0	0	1	0	0	$f_{SX}/2^5$	$f_{IL}/2^5$
0	0	0	1	0	1	$f_{SX}/2^6$	$f_{IL}/2^6$
0	0	0	1	1	0	$f_{SX}/2^7$	$f_{IL}/2^7$
0	0	0	1	1	1	$f_{SX}/2^8$	$f_{IL}/2^8$
0	0	1	0	0	0	$f_{SX}/2^9$	$f_{IL}/2^9$
0	0	1	0	0	1	$f_{SX}/2^{10}$	
0	1	0	0	0	1	$f_{MAIN}/2^8$	
0	1	0	0	1	0	$f_{MAIN}/2^9$	
0	1	0	0	1	1	$f_{MAIN}/2^{10}$	
0	1	0	1	0	0	$f_{MAIN}/2^{11}$	
0	1	0	1	0	1	$f_{MAIN}/2^{12}$	
0	1	0	1	1	0	$f_{MAIN}/2^{13}$	
0	1	0	1	1	1	$f_{MAIN}/2^{14}$	
0	1	1	0	0	0	$f_{MAIN}/2^{15}$	
0	1	1	0	0	1	$f_{MAIN}/2^{16}$	
0	1	1	0	1	0	$f_{MAIN}/2^{17}$	
0	1	1	0	1	1	$f_{MAIN}/2^{18}$	
1	0	1	0	1	1	$f_{MAIN}/2^{19}$	
Other than above						Setting prohibited	

Cautions 1. Be sure to set bits 6 and 7 to “0”.

2. Set the frame frequency between 32 and 128 Hz (24 to 128 Hz when f_{IL} is selected). Also, when set to internal voltage boosting method, capacitor spit method, set the LCD clock (LCDCL) to 512 Hz or less (235 Hz or less when f_{IL} is selected).

3. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.

Remark f_{MAIN}: Main system clock frequency

f_{SX}: Sub clock

f_{IL}: Low-speed on-chip oscillator clock frequency

24.3.5 LCD boost level control register (VLCD)

VLCD selects the reference voltage that is to be generated when operating the voltage boost circuit (contrast adjustment). The reference voltage can be selected from 16 steps.

This register is set by using an 8-bit memory manipulation instruction.

Reset signal generation sets VLCD to 04H.

Figure 24-6. Format of LCD Boost Level Control Register (VLCD)

Address: FFF43H After reset: 04H R/W

Symbol	7	6	5	4	3	2	1	0
VLCD	0	0	0	VLCD4	VLCD3	VLCD2	VLCD1	VLCD0

VLCD4	VLCD3	VLCD2	VLCD1	VLCD0	Reference voltage selection (contrast adjustment)	V_{L4} voltage	
						1/3 bias method	1/4 bias method
0	0	1	0	0	1.00 V (default)	3.00 V	4.00 V
0	0	1	0	1	1.05 V	3.15 V	4.20 V
0	0	1	1	0	1.10 V	3.30 V	4.40 V
0	0	1	1	1	1.15 V	3.45 V	4.60 V
0	1	0	0	0	1.20 V	3.60 V	4.80 V
0	1	0	0	1	1.25 V	3.75 V	5.00 V
0	1	0	1	0	1.30 V	3.90 V	5.20 V
0	1	0	1	1	1.35 V	4.05 V	Setting prohibited
0	1	1	0	0	1.40 V	4.20 V	Setting prohibited
0	1	1	0	1	1.45 V	4.35 V	Setting prohibited
0	1	1	1	0	1.50 V	4.50 V	Setting prohibited
0	1	1	1	1	1.55 V	4.65 V	Setting prohibited
1	0	0	0	0	1.60 V	4.80 V	Setting prohibited
1	0	0	0	1	1.65 V	4.95 V	Setting prohibited
1	0	0	1	0	1.70 V	5.10 V	Setting prohibited
1	0	0	1	1	1.75 V[Note]	5.25 V	Setting prohibited
Other than above					Setting prohibited		

Note Use the LCD driving voltage V_{L1} if the condition $V_{DD} \geq V_{L1}$ applies.

Cautions 1. The VLCD setting is valid only when the voltage boost circuit is operating.

2. Be sure to set bits 5 to 7 to "0".

3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).

4. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set VLCON to 1.

5. To use the external resistance division method or capacitor split method, use the VLCD register with its initial value (04H).

24.3.6 LCD input switch control register (ISCLCD)

Input to the Schmitt trigger buffer must be disabled until the CAPL/P126, CAPH/P127, and V_{L3}/P125 pins are set to operate as LCD function pins in order to prevent through-current from entering.

This register is set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets ISCLCD to 00H.

Figure 24-7. Format of LCD Input Switch Control Register (ISCLCD)

Address: F0308H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISCLCD	0	0	0	0	0	0	ISCVL3	ISCCAP

ISCVL3	V_{L3}/P125 pin Schmitt trigger buffer control
0	Input invalid
1	Input valid

ISCCAP	CAPL/P126, CAPH/P127 pins Schmitt trigger buffer control
0	Input invalid
1	Input valid

Cautions 1. If ISCVL3 = 0, set the corresponding port registers as follows:
PU125 bit of PU12 register = 0, P125 bit of P12 register = 0

2. If ISCCAP = 0, set the corresponding port registers 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

(1) Operation of ports that alternately function as V_{L3}, CAPL, and CAPH pins

The functions of the V_{L3}/P125, CAPL/P126, and CAPH/P127 pins can be selected by using the LCD input switch control register (ISCLCD), LCD mode register 0 (LCDM0), and port mode register 12 (PM12).

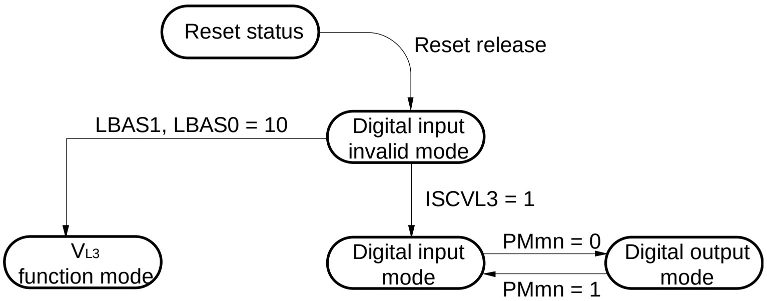
- **V_{L3}/P125**

Table 24-5. Settings of V_{L3}/P125 Pin Function

Bias Setting (LBAS1 and LBAS0 Bits of LCDM0 Register)	ISCVL3 Bit of ISCLCD Register	PM125 Bit of PM12 Register	Pin Function	Initial Status
Other than 1/4 bias method (LBAS1, LBAS0 = 00 or 01)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
1/4 bias method (LBAS1, LBAS0 = 10)	0	1	V_{L3} function mode	–
Other than above			Setting prohibited	

The following shows the V_{L3}/P125 pin function status transitions.

Figure 24-8. V_{L3}/P125 Pin Function Status Transitions



Caution Be sure to set the V_{L3} function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

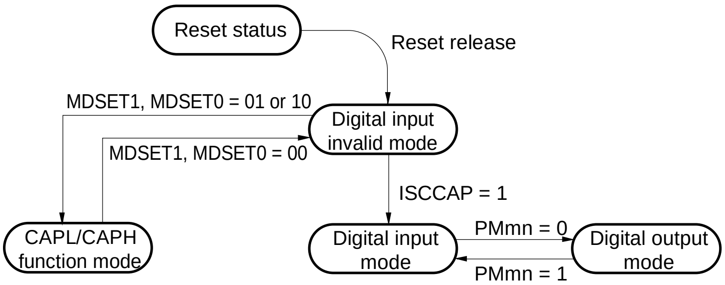
- **CAPL/P126 and CAPH/P127**

Table 24-6. Settings of CAPL/P126 and CAPH/P127 Pin Functions

LCD Drive Voltage Generator (MDSET1 and MDSET0 Bits of LCDM0 Register)	ISCCAP Bit of ISCLCD Register	PM126 and PM127 Bits of PM12 Register	Pin Function	Initial Status
External resistance division (MDSET1, MDSET0 = 00)	0	1	Digital input invalid mode	√
	1	0	Digital output mode	–
	1	1	Digital input mode	–
Internal voltage boosting or capacitor split (MDSET1, MDSET0 = 01 or 10)	0	1	CAPL/CAPH function mode	–
Other than above			Setting prohibited	

The following shows the CAPL/P126 and CAPH/P127 pin function status transitions.

Figure 24-9. CAPL/P126 and CAPH/P127 Pin Function Status Transitions



Caution Be sure to set the CAPL/CAPH function mode before segment output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

24.3.7 LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

These registers specify whether to use pins P02 to P07, P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, and P90 to P97 as port pins (other than as segment and common output pins) or as segment and common output pins.

These registers are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to the value after a reset shown in **Figure 24-10**.

Remark The correspondence between the common output pins (COMx) and the PFSEG register (PFCOMx bits) and the existence of COMx pins in each product are shown in **Table 24-7 Common Output Pins in Each Product and Correspondence with PFSEG Register (PFCOM Bits)**.
The correspondence between the segment output pins (SEGxx) and the PFSEG register (PFSEGxx bits) and the existence of SEGxx pins in each product are shown in **Table 24-8 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)**.

Figure 24-10. Format of LCD port function registers 0 to 6 (PFSEG0 to PFSEG6)

Address: F0300H After reset: 0FH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG0	0	0	0	0	PFCOM3	PFCOM2	PFCOM1	PFCOM0

Address: F0301H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG1	PFSEG07	PFSEG06	PFSEG05	PFSEG04	PFSEG03	PFSEG02	PFSEG01	PFSEG00

Address: F0302H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG2	PFSEG15	PFSEG14	PFSEG13	PFSEG12	PFSEG11	PFSEG10	PFSEG09	PFSEG08

Address: F0303H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG3	PFSEG23	PFSEG22	PFSEG21	PFSEG20	PFSEG19	PFSEG18	PFSEG17	PFSEG16

Address: F0304H After reset: FFH (R5F10NPL), 0FH (R5F10NML) R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG4	PFSEG31 Notes 1, 2	PFSEG30 Notes 1, 2	PFSEG29 Notes 1, 2	PFSEG28 Notes 1, 2	PFSEG27	PFSEG26	PFSEG25	PFSEG24

Address: F0305H After reset: FFH (R5F10NPL), 3FH (R5F10NML) R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG5	PFSEG39 Notes 1, 2	PFSEG38 Notes 1, 2	PFSEG37	PFSEG36	PFSEG35	PFSEG34	PFSEG33	PFSEG32

Address: F0306H After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
PFSEG6 Note 3	0	0	0	0	0	0	PFSEG41	PFSEG40

(**Notes** and **Caution** are listed on the next page.)

PFCOMx (x = 0 to 3)	Selection of Pmn pins to operate as port pins (other than as common outputs) or as common outputs (mn = 90 to 93)
0	Use the Pmn pin as a port pin (other than as a common output)
1	Use the Pmn pin as a common output

PFSEGxx (xx = 00 to 41)	Selection of Pmn pins to operate as port pins (other than as segment and common outputs) or as segment and common outputs (mn = 02 to 07, 10 to 17, 30 to 37, 50 to 57, 70 to 77, 80 to 85, 94 to 97)
0	Use the Pmn pin as a port pin (other than as a segment or common output).
1	Use the Pmn pin as a segment or common output.

Notes 1. For the R5F10NML, the initial value is 0.
Writing 1 to this bit does not affect operation, and the value read is 0.

2. Be sure to set “1” for the 80-pin product.

3. R5F10NPLDFB only

Caution **To use the Pmn pins as segment or common output pins (PFSEGxx = 1 or PFCOMx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to “0”.**

Table 24-7. Common Output Pins in Each Product and Correspondence with PFSEG Register (PFCOM Bits)

Bit Name of PFSEG Register	Corresponding COMx Pins	Alternate Port	100-pin	80-pin
PFCOM0	COM0	P90, (P37)	√	–
		P90	–	√
PFCOM1	COM1	P91, (P36)	√	–
		P91	–	√
PFCOM2	COM2	P92, (P35)	√	–
		P92	–	√
PFCOM3	COM3	P93, (P34)	√	–
		P93	–	√

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

Table 24-8. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits) (1/2)

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	100-pin	80-pin
PFSEG00	SEG0	P94, (P33)	√	–
		P94	–	√
PFSEG01	SEG1	P95, (P32)	√	–
		P95	–	√
PFSEG02	SEG2	P96, (P31)	√	–
		P96	–	√
PFSEG03	SEG3	P97, (P30)	√	–
		P97	–	√
PFSEG04	SEG4	P10	√	√
PFSEG05	SEG5	P11	√	√
PFSEG06	SEG6	P12	√	√
PFSEG07	SEG7	P13	√	√
PFSEG08	SEG8	P14	√	√
PFSEG09	SEG9	P15	√	√
PFSEG10	SEG10	P16	√	√
PFSEG11	SEG11	P17	√	√
PFSEG12	SEG12	P80	√	√
PFSEG13	SEG13	P81	√	√
PFSEG14	SEG14	P82	√	√
PFSEG15	SEG15	P83	√	√
PFSEG16	SEG16	P70	√	√
PFSEG17	SEG17	P71	√	√
PFSEG18	SEG18	P72	√	√
PFSEG19	SEG19	P73	√	√
PFSEG20	SEG20	P74	√	√
PFSEG21	SEG21	P75	√	√
PFSEG22	SEG22	P76	√	√
PFSEG23	SEG23	P77	√	√
PFSEG24	SEG24	P30, (P97)	√	–
		P30	–	√
PFSEG25	SEG25	P31, (P96)	√	–
		P31	–	√
PFSEG26	SEG26	P32, (P95)	√	–
		P32	–	√
PFSEG27	SEG27	P33, (P94)	√	–
		P33	–	√
PFSEG28	SEG28	P34, (P93)	√	–
PFSEG29	SEG29	P35, (P92)	√	–
PFSEG30	SEG30	P36, (P91)	√	–
PFSEG31	SEG31	P37, (P90)	√	–

(**Remark** is listed on the next page.)

Table 24-8. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits) (2/2)

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	100-pin	80-pin
PFSEG32	SEG32	P50	√	–
		P02	–	√
PFSEG33	SEG33	P51	√	–
		P03	–	√
PFSEG34	SEG34	P52	√	–
		P04	–	√
PFSEG35	SEG35	P53	√	–
		P05	–	√
PFSEG36	SEG36	P54	√	–
		P06	–	√
PFSEG37	SEG37	P55	√	–
		P07	–	√
PFSEG38	SEG38	P56	√	–
PFSEG39	SEG39	P57	√	–
PFSEG40	SEG40	P84	√	–
PFSEG41	SEG41	P85	√	–

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR0). See **Figure 4-7 Format of Peripheral I/O Redirection Register (PIOR0)**.

(1) Operation of ports that have multiplexed SEGxx or COMx pin functions

The functions of port pins that also serve as segment output pins (SEGxx) or common output pins (COMx) can be selected by using the port mode register (PMxx) and LCD port function registers 0 to 6 (PFSEG0 to PFSEG6).

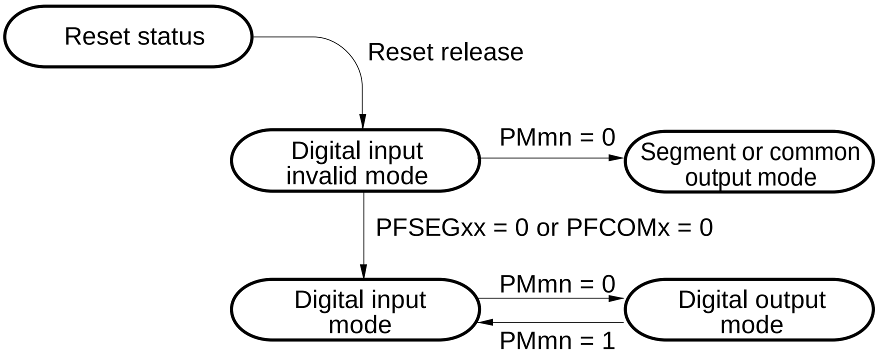
- **P02 to P07, P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P90 to P97 (ports that do not serve as analog input pins (ANIxx))**

Table 24-9. Setting of the SEGxx or COMx and Port Pin Functions

PFSEGxx or PFCOMx Bit of PFSEG0 to PFSEG6 Registers	PMxx Bit of PMxx Register	Pin Function	Initial Status
1	1	Digital input invalid mode	√
0	0	Digital output mode	–
0	1	Digital input mode	–
1	0	Segment or common output mode	–

The following shows the state transitions of the SEGxx or COMx and port pin functions.

Figure 24-11. State Transition Diagram of SEGxx or COMx and Port Pin Functions



Caution Be sure to set the segment or common output mode before segment or common output starts (while SCOC bit of LCD mode register 1 (LCDM1) is 0).

24.3.8 Port mode registers 0, 1, 3, 5, 7, 8, 9 (PM0, PM1, PM3, PM5, PM7, PM8, PM9)

These registers specify input/output of ports 0, 1, 5, 7, 8, and 9 in 1-bit units.

When using the ports (such as P10/SEG4) to be shared with the segment output pin or common output pin for segment or common output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P10/SEG4 for segment output
Set the PM10 bit of port mode register 1 to “0”.
Set the P10 bit of port register 1 to “0”.

These registers are set by using a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 24-12. Format of Port Mode Registers 0, 1, 3, 5, 7, 8, 9 (PM0, PM1, PM3, PM5, PM7, PM8, PM9)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	1	1	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	1	1	PM85	PM84	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 5, 7, 8, 9; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode registers 0, 1, 3, 5, 7, 8, and 9. The format of the port mode register of other products, see **Figure 4-1 Format of Port Mode Register**.

24.4 LCD Display Data Registers

The LCD display data registers are mapped as shown in **Table 24-10**. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 24-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (1/4)

(a) Other than 6-time slice and 8-time slice (static, 2-time slice, 3-time slice, and 4-time slice) (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
		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)				√	√

Table 24-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (2/4)

(a) Other than 6-time slice and 8-time slice (static, 2-time slice, 3-time slice, and 4-time slice) (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
		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)				√	–

Remark √: Supported, –: Not supported

Table 24-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (3/4)

(b) 6-time slice and 8-time slice (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
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG0	F0400H	SEG0$^{\text{Note}}$								√	√
SEG1	F0401H	SEG1$^{\text{Note}}$								√	√
SEG2	F0402H	SEG2$^{\text{Note}}$								√	√
SEG3	F0403H	SEG3$^{\text{Note}}$								√	√
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								√	√
SEG36	F0424H	SEG36								√	√
SEG37	F0425H	SEG37								√	√
SEG38	F0426H	SEG38								√	–
SEG39	F0427H	SEG39								√	–
SEG40	F0428H	SEG40								√	–

Table 24-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (4/4)

(b) 6-time slice and 8-time slice (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
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0		
SEG41	F0429H	SEG41								√	–

Note The COM4 to COM7 pins and SEG0 to SEG3 pins are used alternatively.

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.

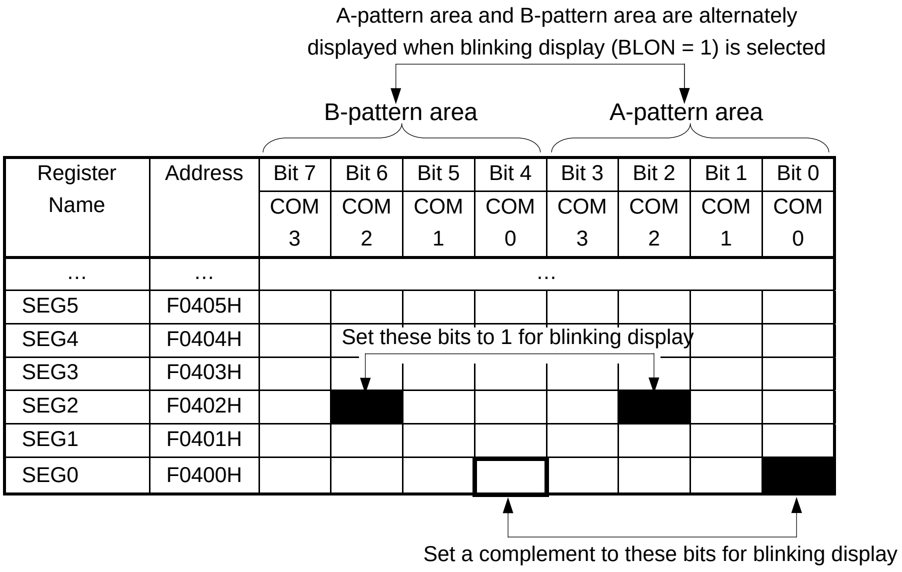
24.5 Selection of LCD Display Register

With the RL78/I1C (512 KB), 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 A-pattern and B-pattern area data (blinking display corresponding to the constant-period interrupt timing of the independent power supply RTC)

Caution When the number of time slices is six or eight, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

Figure 24-13. Example of Setting LCD Display Registers When Pattern Is Changed


A-pattern area and B-pattern area are alternately displayed when blinking display (BLON = 1) is selected
B-pattern area
A-pattern area
Set these bits to 1 for blinking display
Set a complement to these bits for blinking display

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		COM 3	COM 2	COM 1	COM 0	COM 3	COM 2	COM 1	COM 0
…	…	…							
SEG5	F0405H								
SEG4	F0404H								
SEG3	F0403H								
SEG2	F0402H		■				■		
SEG1	F0401H								
SEG0	F0400H				□				■

24.5.1 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 **24.4 LCD Display Data Registers** about the display area.

24.5.2 Blinking display (Alternately displaying A-pattern and B-pattern area data)

When BLON = 1 has been set, A-pattern and B-pattern area data will be alternately displayed, according to the fixed-cycle interrupt (INTRTCPRD) timing of the independent power supply RTC. See **CHAPTER 9 REALTIME CLOCK WITH INDEPENDENT POWER SUPPLY** about the setting of the fixed-cycle interrupt (INTRTCPRD, 0.5 s setting only) timing of the independent power supply RTC.

For blinking display of the LCD, set inverted values to the B-pattern area bits corresponding to the A-pattern area bits. (Example: Set 1 to bit 0 of 00H, and set 0 to bit 4 of F0400H for blinking display.) When not setting blinking display of the LCD, set the same values. (Example: Set 1 to bit 2 of F0402H, and set 1 to bit 6 of F0402H for lighting display.)

See **24.4 LCD Display Data Registers** about the display area.

Next, the timing operation of display switching is shown.

Figure 24-14. Switching Operation from A-Pattern Display to Blinking Display

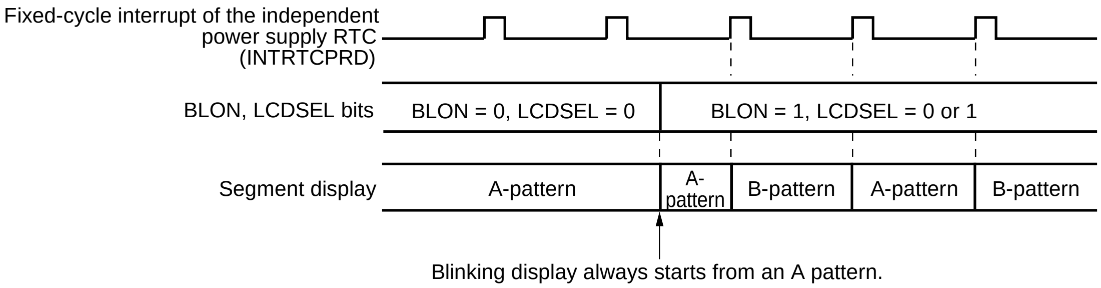
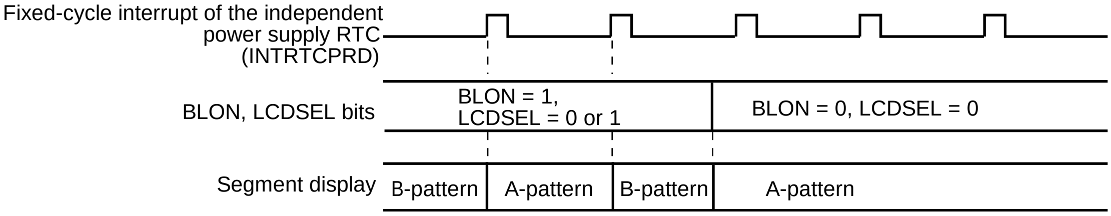

Fixed-cycle interrupt of the independent power supply RTC (INTRTCPRD)
BLON, LCDSEL bits
BLON = 0, LCDSEL = 0
BLON = 1, LCDSEL = 0 or 1
Segment display
A-pattern
A-pattern
B-pattern
A-pattern
B-pattern
Blinking display always starts from an A pattern.

Figure 24-15. Switching Operation from Blinking Display to A-Pattern Display


Fixed-cycle interrupt of the independent power supply RTC (INTRTCPRD)
BLON, LCDSEL bits
BLON = 1, LCDSEL = 0 or 1
BLON = 0, LCDSEL = 0
Segment display
B-pattern
A-pattern
B-pattern
A-pattern

24.6 Setting the LCD Controller/Driver

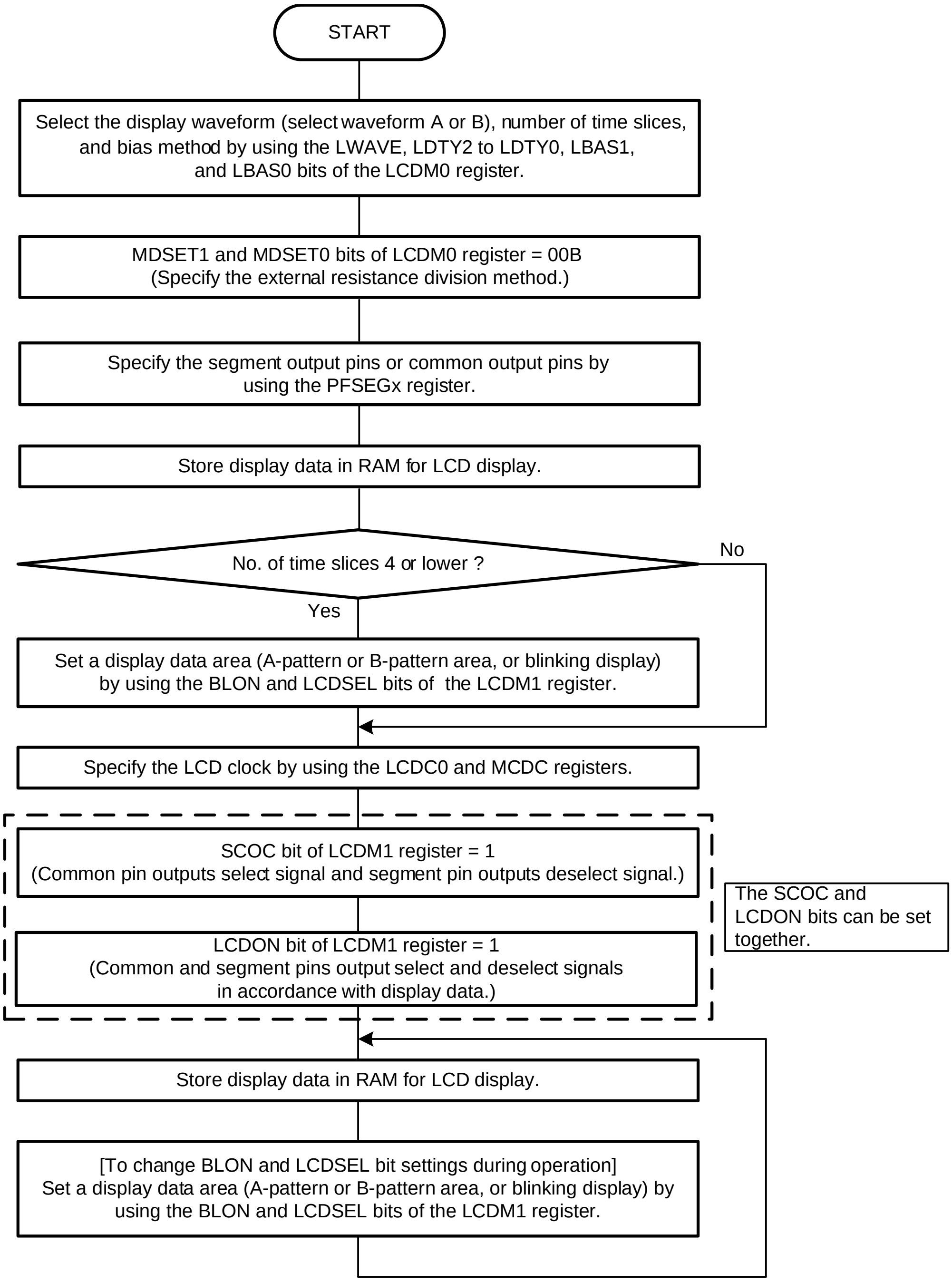
Set the LCD controller/driver using the following procedure.

Cautions 1. To operate the LCD controller/driver, be sure to follow procedures (1) to (3). Unless these procedures are observed, the operation will not be guaranteed.

2. The steps shown in the flowcharts in (1) to (3) are performed by the CPU.

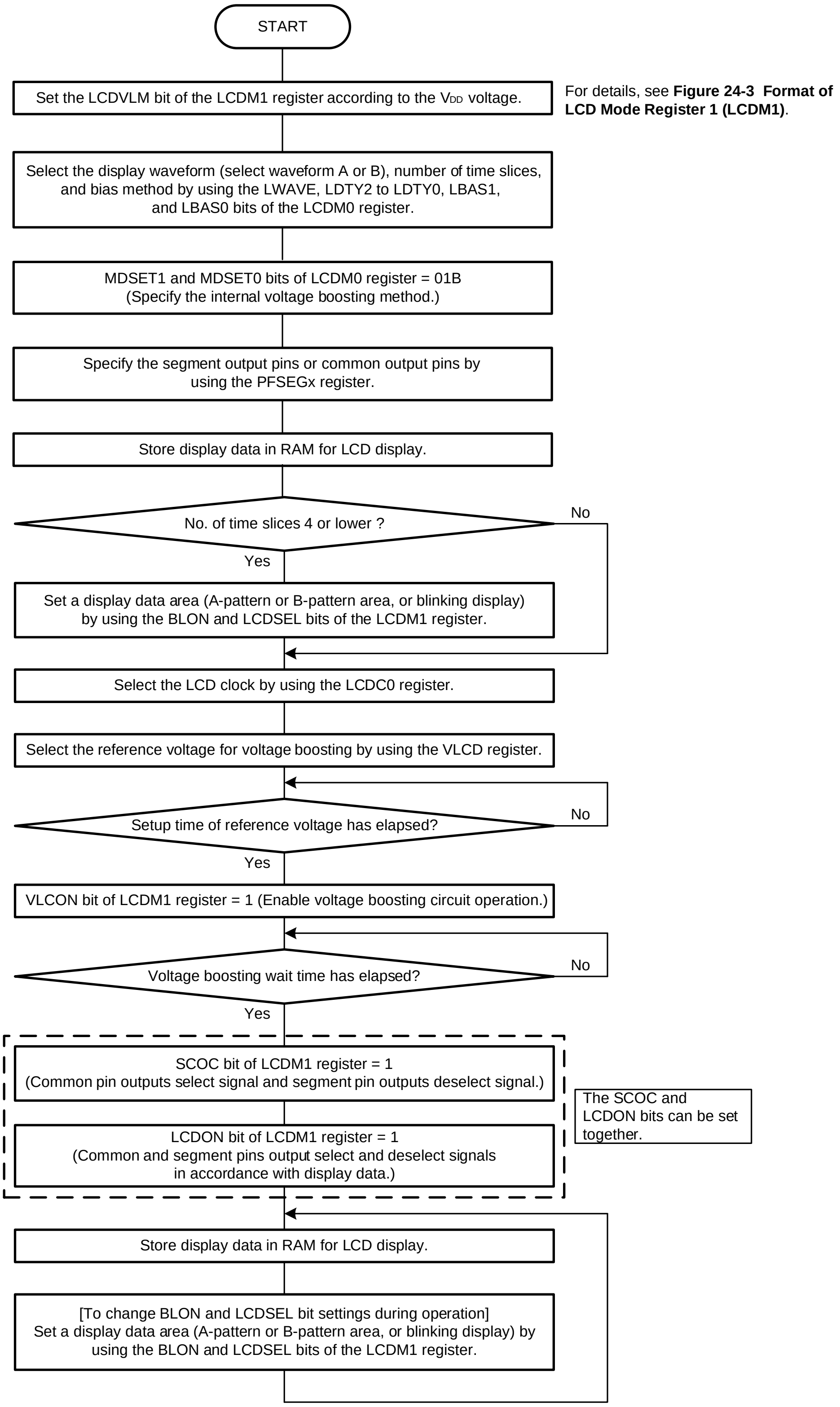
(1) External resistance division method

Figure 24-16. External Resistance Division Method Setting Procedure


START
Select the display waveform (select waveform A or B), number of time slices, and bias method by using the LWAVE, LDTY2 to LDTY0, LBAS1, and LBAS0 bits of the LCDM0 register.
MDSET1 and MDSET0 bits of LCDM0 register = 00B (Specify the external resistance division method.)
Specify the segment output pins or common output pins by using the PFSEGx register.
Store display data in RAM for LCD display.
No. of time slices 4 or lower ?
No
Yes
Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.
Specify the LCD clock by using the LCDC0 and MCDC registers.
SCOC bit of LCDM1 register = 1 (Common pin outputs select signal and segment pin outputs deselect signal.)
LCDON bit of LCDM1 register = 1 (Common and segment pins output select and deselect signals in accordance with display data.)
The SCOC and LCDON bits can be set together.
Store display data in RAM for LCD display.
[To change BLON and LCDSEL bit settings during operation] Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.

(2) Internal voltage boosting method

Figure 24-17. Internal Voltage Boosting Method Setting Procedure

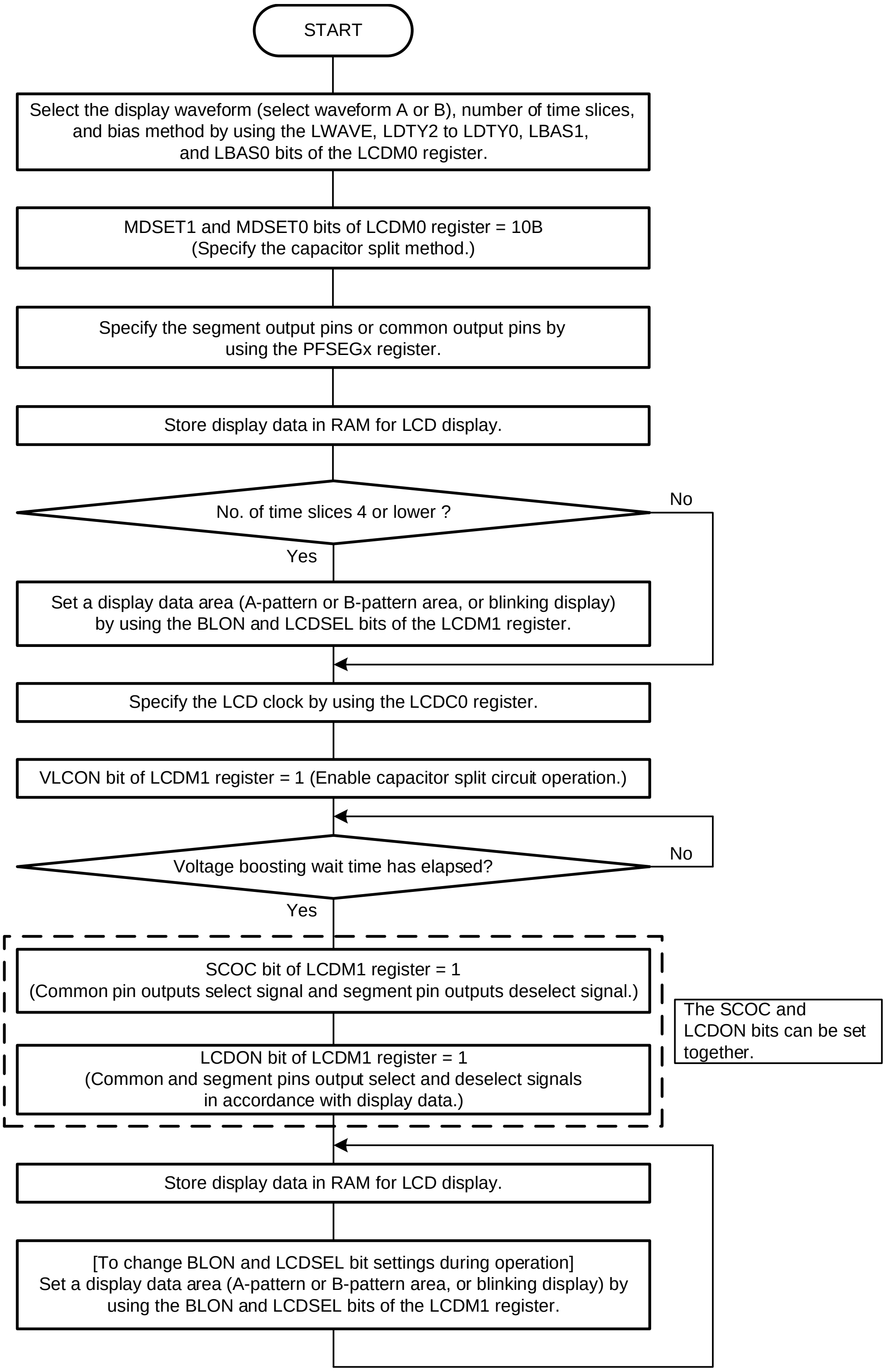

START
Set the LCDVLM bit of the LCDM1 register according to the VDD voltage.
For details, see **Figure 24-3 Format of LCD Mode Register 1 (LCDM1)**.
Select the display waveform (select waveform A or B), number of time slices, and bias method by using the LWAVE, LDTY2 to LDTY0, LBAS1, and LBAS0 bits of the LCDM0 register.
MDSET1 and MDSET0 bits of LCDM0 register = 01B (Specify the internal voltage boosting method.)
Specify the segment output pins or common output pins by using the PFSEGx register.
Store display data in RAM for LCD display.
No. of time slices 4 or lower ?
No
Yes
Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.
Select the LCD clock by using the LCDC0 register.
Select the reference voltage for voltage boosting by using the VLCD register.
Setup time of reference voltage has elapsed?
No
Yes
VLCON bit of LCDM1 register = 1 (Enable voltage boosting circuit operation.)
Voltage boosting wait time has elapsed?
No
Yes
SCOC bit of LCDM1 register = 1 (Common pin outputs select signal and segment pin outputs deselect signal.)
LCDON bit of LCDM1 register = 1 (Common and segment pins output select and deselect signals in accordance with display data.)
The SCOC and LCDON bits can be set together.
Store display data in RAM for LCD display.
[To change BLON and LCDSEL bit settings during operation] Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.

Cautions 1. Wait until the setup time has elapsed even if not changing the setting of the VLCD register.

2. For the specifications of the reference voltage setup time and voltage boosting wait time, see CHAPTER 43 ELECTRICAL SPECIFICATIONS.

(3) Capacitor split method

Figure 24-18. Capacitor Split Method Setting Procedure

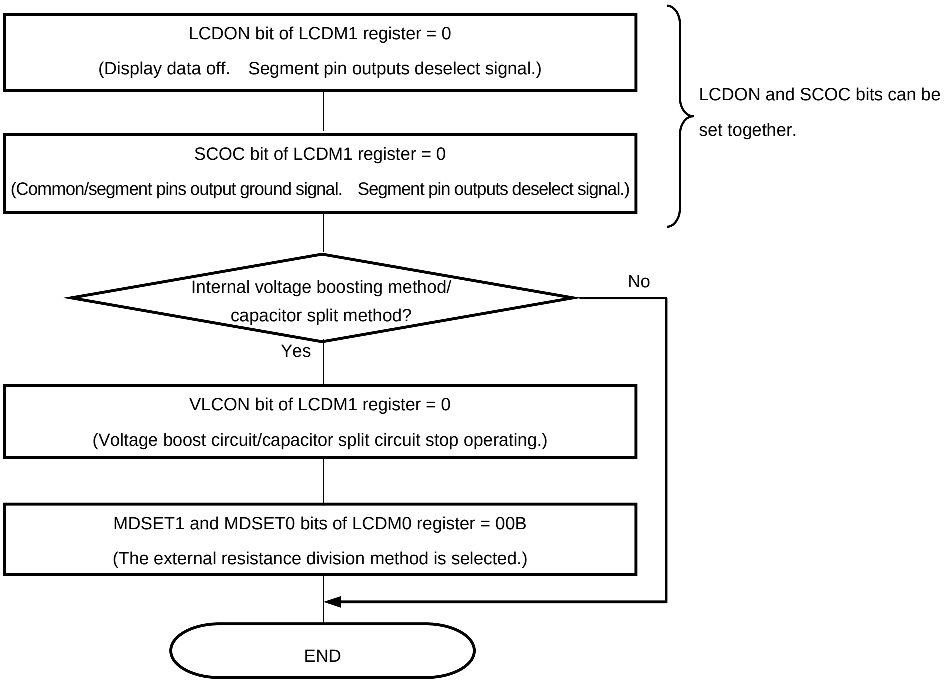

START
Select the display waveform (select waveform A or B), number of time slices, and bias method by using the LWAVE, LDTY2 to LDTY0, LBAS1, and LBAS0 bits of the LCDM0 register.
MDSET1 and MDSET0 bits of LCDM0 register = 10B (Specify the capacitor split method.)
Specify the segment output pins or common output pins by using the PFSEGx register.
Store display data in RAM for LCD display.
No. of time slices 4 or lower ?
No
Yes
Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.
Specify the LCD clock by using the LCDC0 register.
VLCON bit of LCDM1 register = 1 (Enable capacitor split circuit operation.)
Voltage boosting wait time has elapsed?
No
Yes
SCOC bit of LCDM1 register = 1 (Common pin outputs select signal and segment pin outputs deselect signal.)
LCDON bit of LCDM1 register = 1 (Common and segment pins output select and deselect signals in accordance with display data.)
The SCOC and LCDON bits can be set together.
Store display data in RAM for LCD display.
[To change BLON and LCDSEL bit settings during operation] Set a display data area (A-pattern or B-pattern area, or blinking display) by using the BLON and LCDSEL bits of the LCDM1 register.

Caution For the specifications of the voltage boosting wait time, see CHAPTER 43 ELECTRICAL SPECIFICATIONS.

24.7 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 24-19. Operation Stop Procedure


LCDON bit of LCDM1 register = 0
(Display data off. Segment pin outputs deselect signal.)
SCOC bit of LCDM1 register = 0
(Common/segment pins output ground signal. Segment pin outputs deselect signal.)
LCDON and SCOC bits can be set together.
Internal voltage boosting method/ capacitor split method?
No
Yes
VLCON bit of LCDM1 register = 0
(Voltage boost circuit/capacitor split circuit stop operating.)
MDSET1 and MDSET0 bits of LCDM0 register = 00B
(The external resistance division method is selected.)
END

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 = 00B) before stopping the voltage boost/capacitor split circuits (VLCON bit of LCDM1 register = 0).**

24.8 Supplying LCD Drive Voltages V_{L1}, V_{L2}, V_{L3}, and V_{L4}

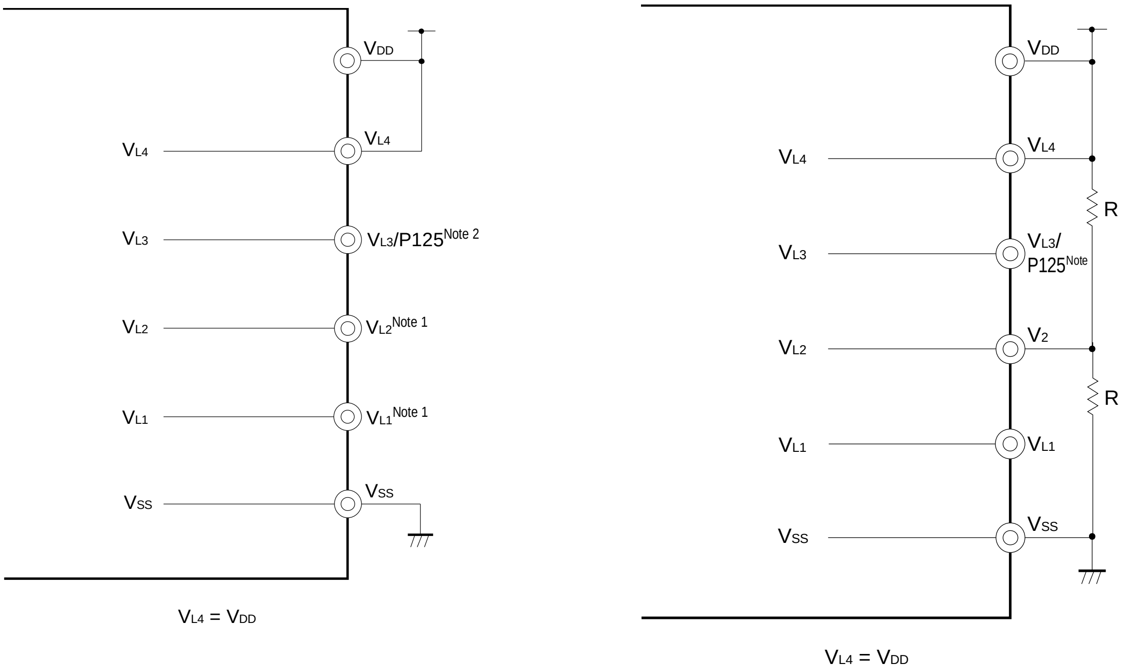
The external resistance division method, internal voltage boosting method, and capacitor split method can be selected as LCD drive power generating method.

24.8.1 External resistance division method

Figure 24-20 shows examples of LCD drive voltage connection, corresponding to each bias method.

Figure 24-20. Examples of LCD Drive Power Connections (External Resistance Division Method) (1/2)

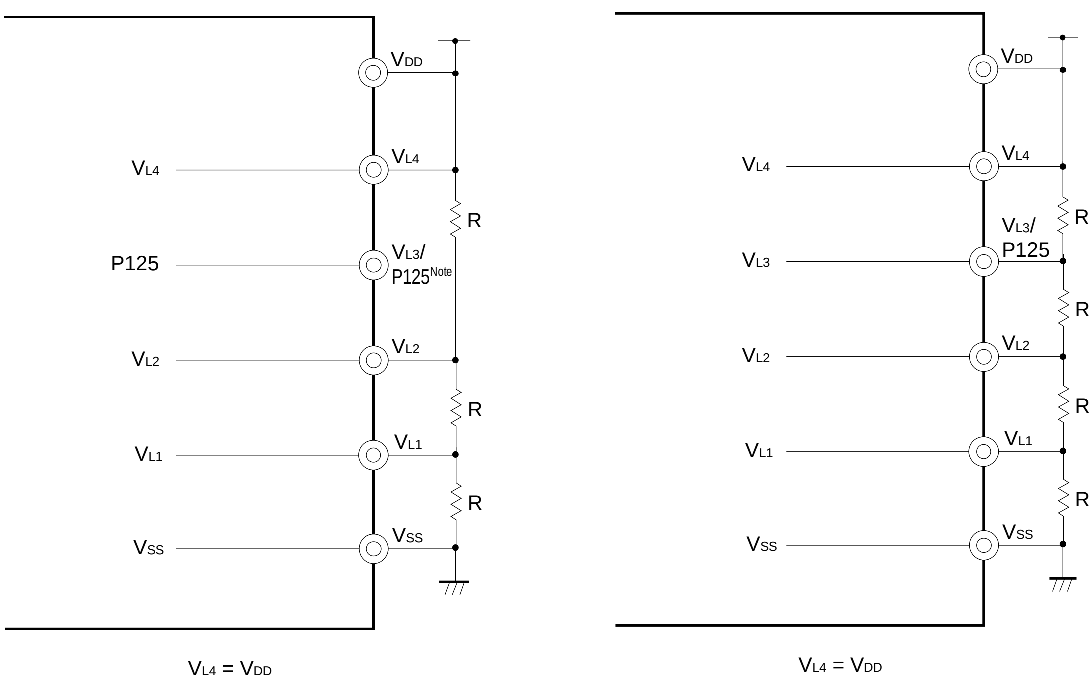
(a) Static display mode **(b) 1/2 bias method**



Notes 1. Connect V_{L1} and V_{L2} to GND or leave open.

2. V_{L3} can be used as port (P125).

Figure 24-20. Examples of LCD Drive Power Connections (External Resistance Division Method) (2/2)



(c) 1/3 bias method

(d) 1/4 bias method

Note V_{L3} can be used as port (P125).

Caution The reference resistance “R” value for external resistance division is 10 kΩ to 1 MΩ. Also, to stabilize the potential of the V_{L1} to V_{L4} pins, connect a capacitor between each of pins V_{L1} to V_{L4} 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.

24.8.2 Internal voltage boosting method

The RL78/I1C (512 KB) contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors (0.47 μF±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode or 1/4 bias mode can be set for the internal voltage boosting method.

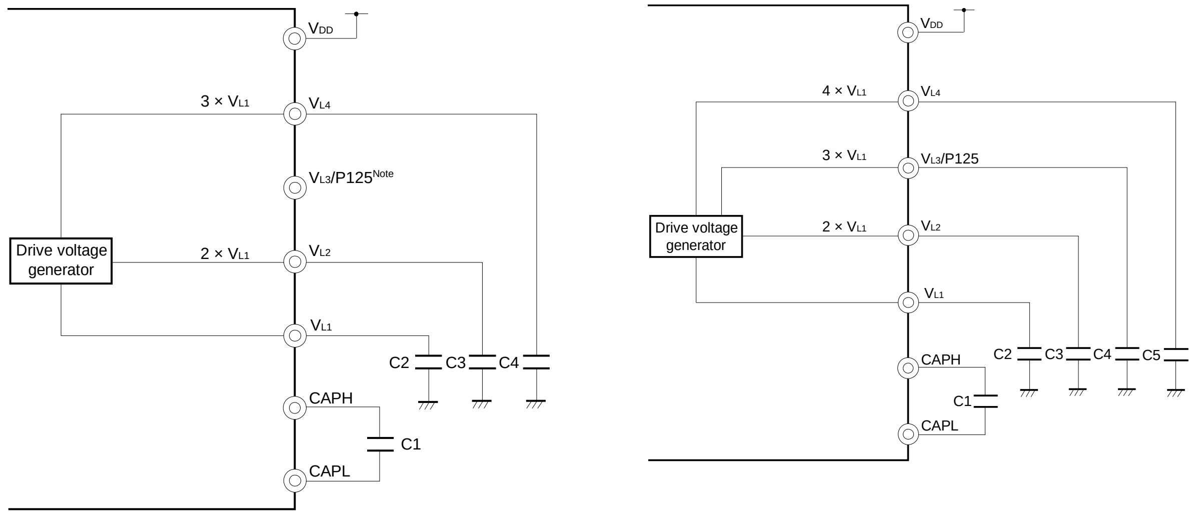
The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in V_{DD}, because it is a power supply separate from the main unit.

In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

Table 24-11. LCD Drive Voltages (Internal Voltage Boosting Method)

Bias Method LCD Drive Voltage Pin	1/3 Bias Method	1/4 Bias Method
V_{L4}	3 × V_{L1}	4 × V_{L1}
V_{L3}	–	3 × V_{L1}
V_{L2}	2 × V_{L1}	2 × V_{L1}
V_{L1}	LCD reference voltage	LCD reference voltage

Figure 24-21. Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)



(a) 1/3 bias method

(b) 1/4 bias method

Note V_{L3} can be used as port (P125).

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

24.8.3 Capacitor split method

The RL78/I1C (512 KB) contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors (0.47 μF±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

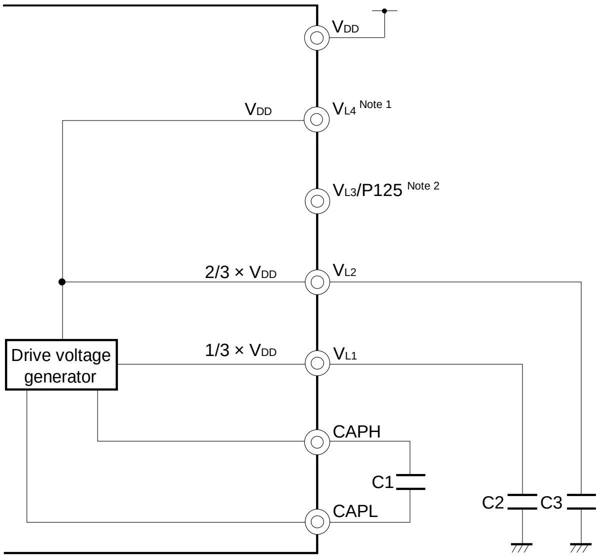
Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

Table 24-12. LCD Drive Voltages (Capacitor Split Method)

Bias Method LCD Drive Voltage Pin	1/3 Bias Method
V_{L4}	V_{DD}
V_{L3}	–
V_{L2}	2/3 × V_{L4}
V_{L1}	1/3 × V_{L4}

Figure 24-22. Examples of LCD Drive Power Connections (Capacitor Split Method)

- **1/3 bias method**



Notes 1. When switching to internal voltage boosting method, connect capacitor C4 as shown in **Figure 24-21 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**.

2. V_{L3} can be used as port (P125).

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

24.9 Common and Segment Signals

24.9.1 Normal liquid crystal waveform

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 24-13**. 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.

Use the COM4 to COM7 pins other than in the six-time-slice mode and eight-time-slice mode, and COM6, COM7 pins in the six-time-slice mode as open or segment pins.

Table 24-13. COM Signals

<table>
<tr><th>COM Signal
Number of Time Slices</th><th>COM0</th><th>COM1</th><th>COM2</th><th>COM3</th><th>COM4</th><th>COM5</th><th>COM6</th><th>COM7</th></tr>
<tr><td>Static display mode</td><td>→</td><td>→</td><td>→</td><td>→</td><td>Note</td><td>Note</td><td>Note</td><td>Note</td></tr>
<tr><td>Two-time-slice mode</td><td colspan="2">→</td><td>Open</td><td>Open</td><td>Note</td><td>Note</td><td>Note</td><td>Note</td></tr>
<tr><td>Three-time-slice mode</td><td colspan="3">→</td><td>Open</td><td>Note</td><td>Note</td><td>Note</td><td>Note</td></tr>
<tr><td>Four-time-slice mode</td><td colspan="4">→</td><td>Note</td><td>Note</td><td>Note</td><td>Note</td></tr>
<tr><td>Six-time-slice mode</td><td colspan="6">→</td><td>Note</td><td>Note</td></tr>
<tr><td>Eight-time-slice mode</td><td colspan="8">→</td></tr>
</table>

Note Use the pins as open or segment pins.

(2) Segment signals

The segment signals correspond to the LCD display data register (see **24.4 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. 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 SEG41).

When the number of time slices is number other than 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 SEG41).

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.

● 80-pin product:	SEG0 to SEG27, SEG32 to SEG37
● 100-pin product:	SEG0 to SEG41

(3) Output waveforms of common and segment signals

The voltages listed in **Table 24-14** are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of ±V_{LCD} is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 24-14. LCD Drive Voltage

(a) Static display mode

Segment Signal	Select Signal Level	Deselect Signal Level
Common Signal	V_{SS}/V_{L4}	V_{L4}/V_{SS}
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
Common Signal		V_{SS}/V_{L4}	V_{L4}/V_{SS}
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
Common Signal		V_{SS}/V_{L4}	V_{L2}/V_{L1}
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}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

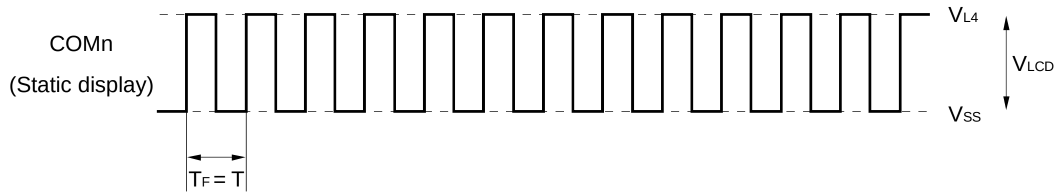
(d) 1/4 bias method (waveform A or B)

Segment Signal		Select Signal Level	Deselect Signal Level
Common Signal		V_{SS}/V_{L4}	V_{L2}
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 24-23 shows the common signal waveforms, and **Figure 24-24** shows the voltages and phases of the common and segment signals.

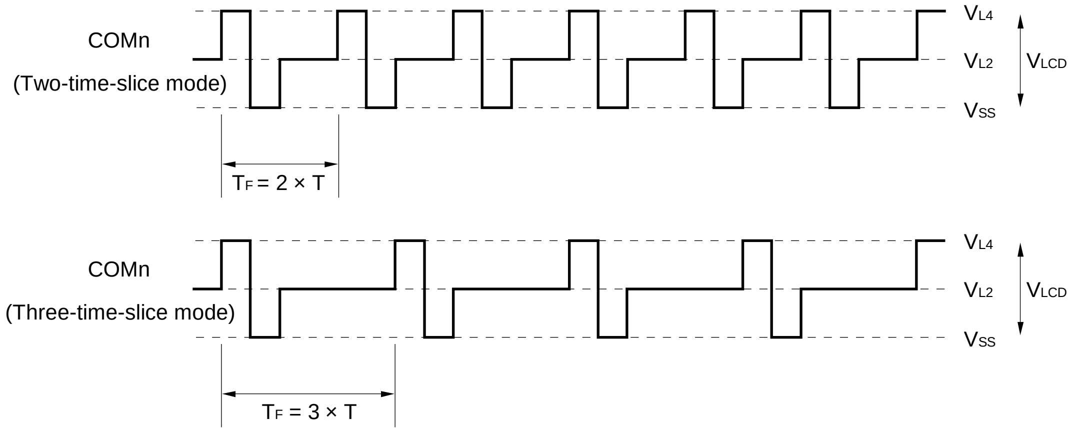
Figure 24-23. Common Signal Waveforms (1/3)

(a) Static display mode



T: One LCD clock period TF: Frame frequency

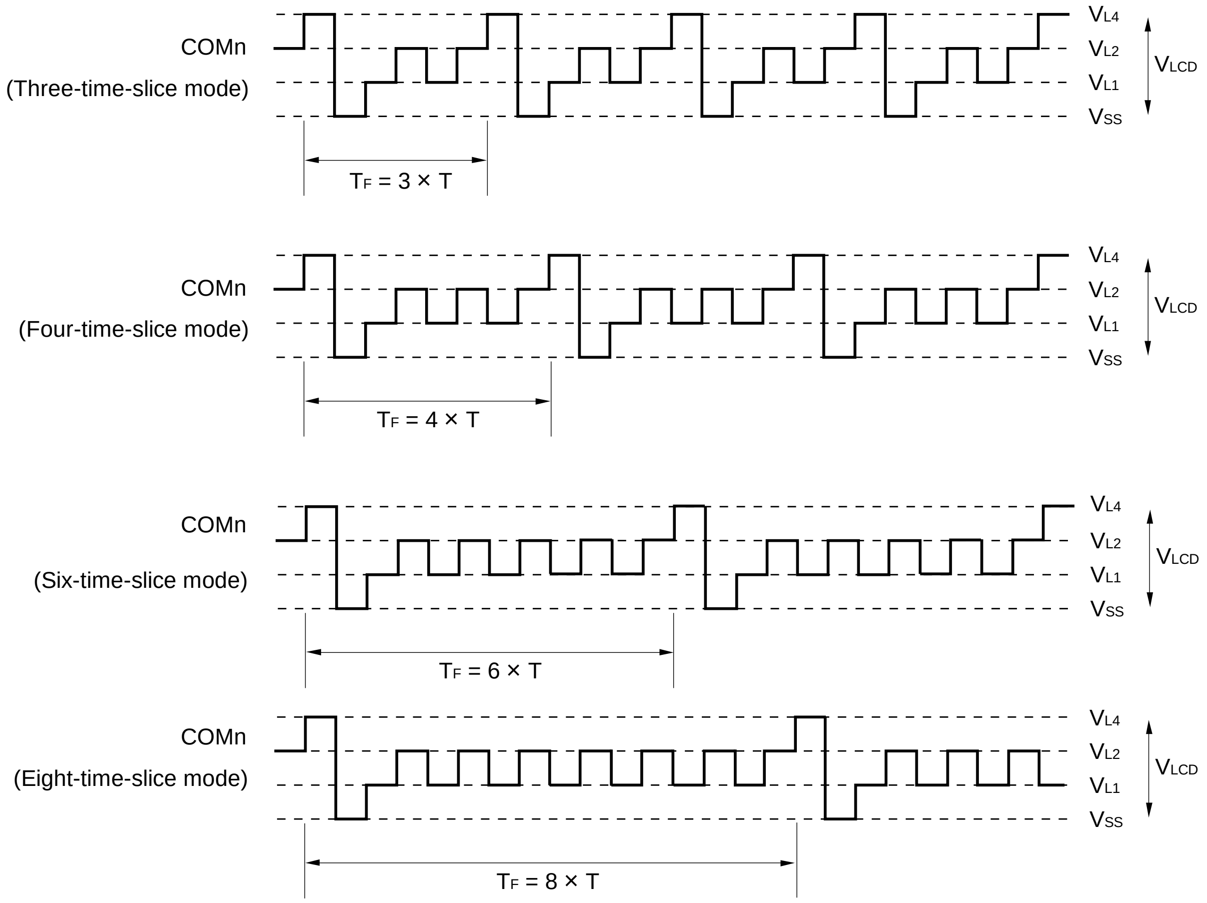
(b) 1/2 bias method



T: One LCD clock period TF: Frame frequency

Figure 24-23. Common Signal Waveforms (2/3)

(c) 1/3 bias method



T: One LCD clock period   TF: Frame frequency

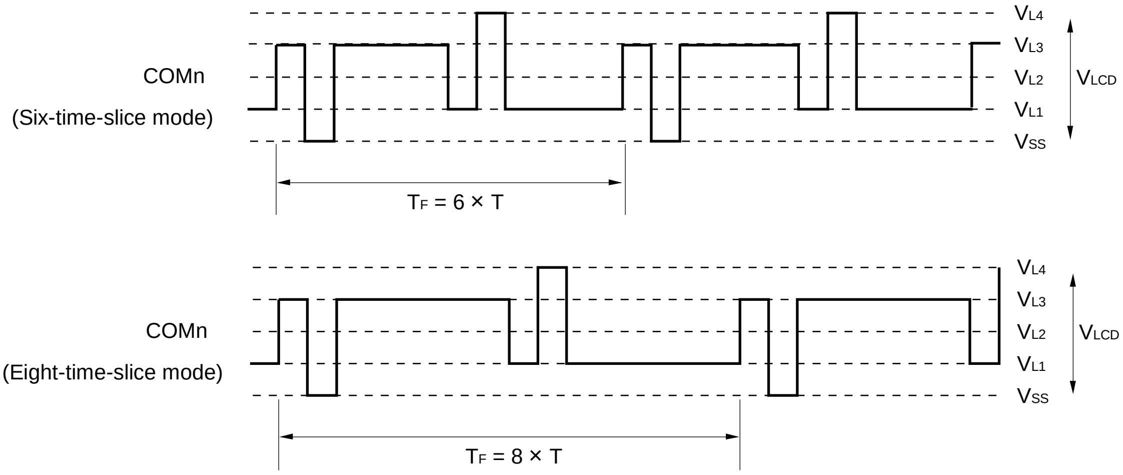
< Example of calculation of LCD frame frequency (When four-time-slice mode is used) >

LCD clock: $32768/2^7$ = 256 Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 64 Hz

Figure 24-23. Common Signal Waveforms (3/3)

(d) 1/4 bias method



T: One LCD clock period TF: Frame frequency

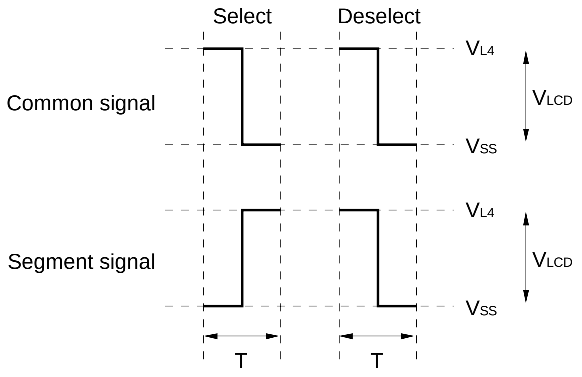
< Example of calculation of LCD frame frequency (When eight-time-slice mode is used) >

LCD clock: $32768/2^7$ = 256 Hz (When setting to LCDC0 = 06H)

LCD frame frequency: 32 Hz

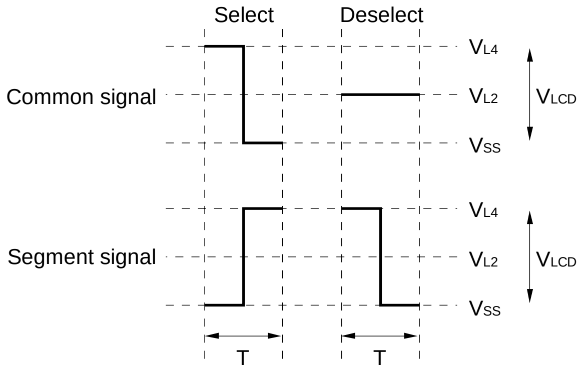
Figure 24-24. Voltages and Phases of Common and Segment Signals (1/3)

(a) Static display mode (waveform A)



T: One LCD clock period

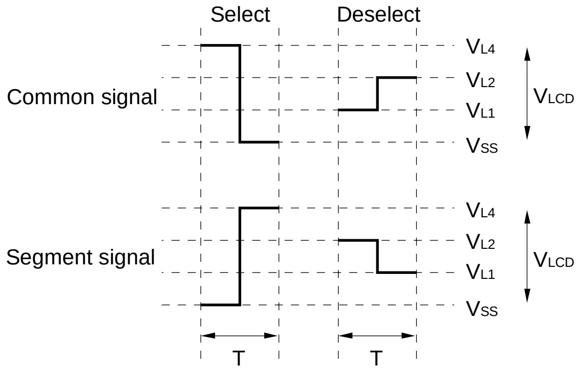
(b) 1/2 bias method (waveform A)



T: One LCD clock period

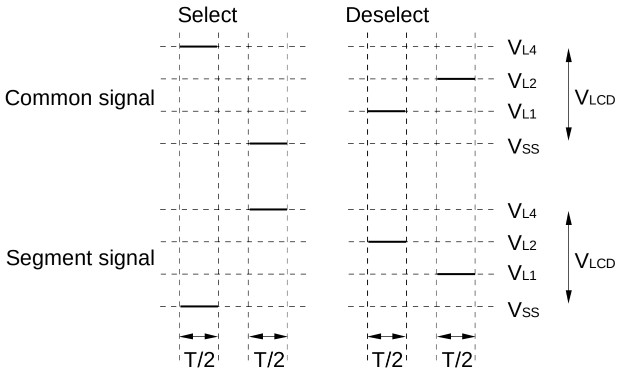
Figure 24-24. Voltages and Phases of Common and Segment Signals (2/3)

(c) 1/3 bias method (waveform A)



T: One LCD clock period

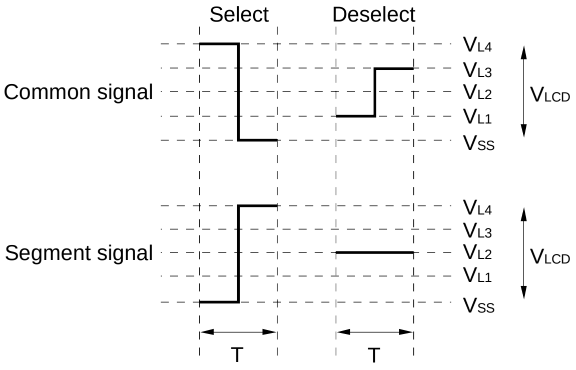
(d) 1/3 bias method (waveform B)



T: One LCD clock period

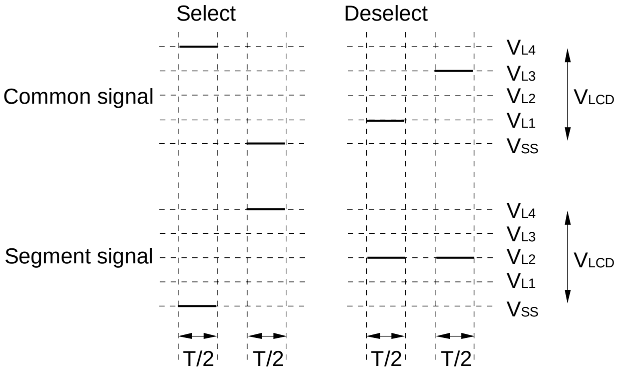
Figure 24-24. Voltages and Phases of Common and Segment Signals (3/3)

(e) 1/4 bias method (waveform A)



T: One LCD clock period

(f) 1/4 bias method (waveform B)



T: One LCD clock period

24.10 Display Modes

24.10.1 Static display example

Figure 24-26 shows how the three-digit LCD panel having the display pattern shown in **Figure 24-25** 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 24-15** at the timing of the common signal COM0; see **Figure 24-25** for the relationship between the segment signals and LCD segments.

Table 24-15. Select and Deselect Voltages (COM0)

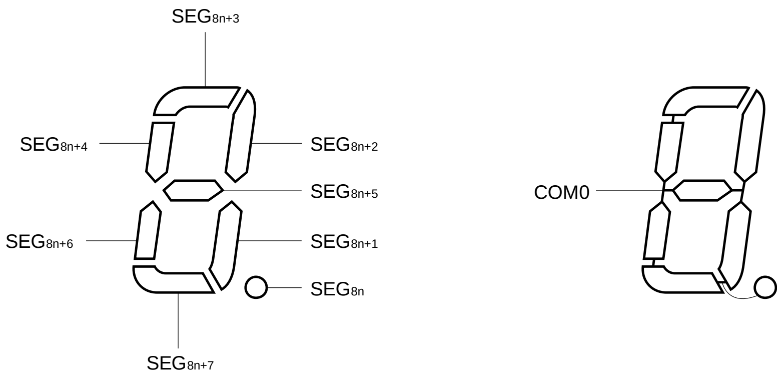
Segment Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to **Table 24-15**, it is determined that the bit-0 pattern of the display data register locations (F0408H to F040FH) must be 10110111.

Figure 24-27 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, +V_{LCD}/–V_{LCD}, is generated to turn on the corresponding LCD segment.

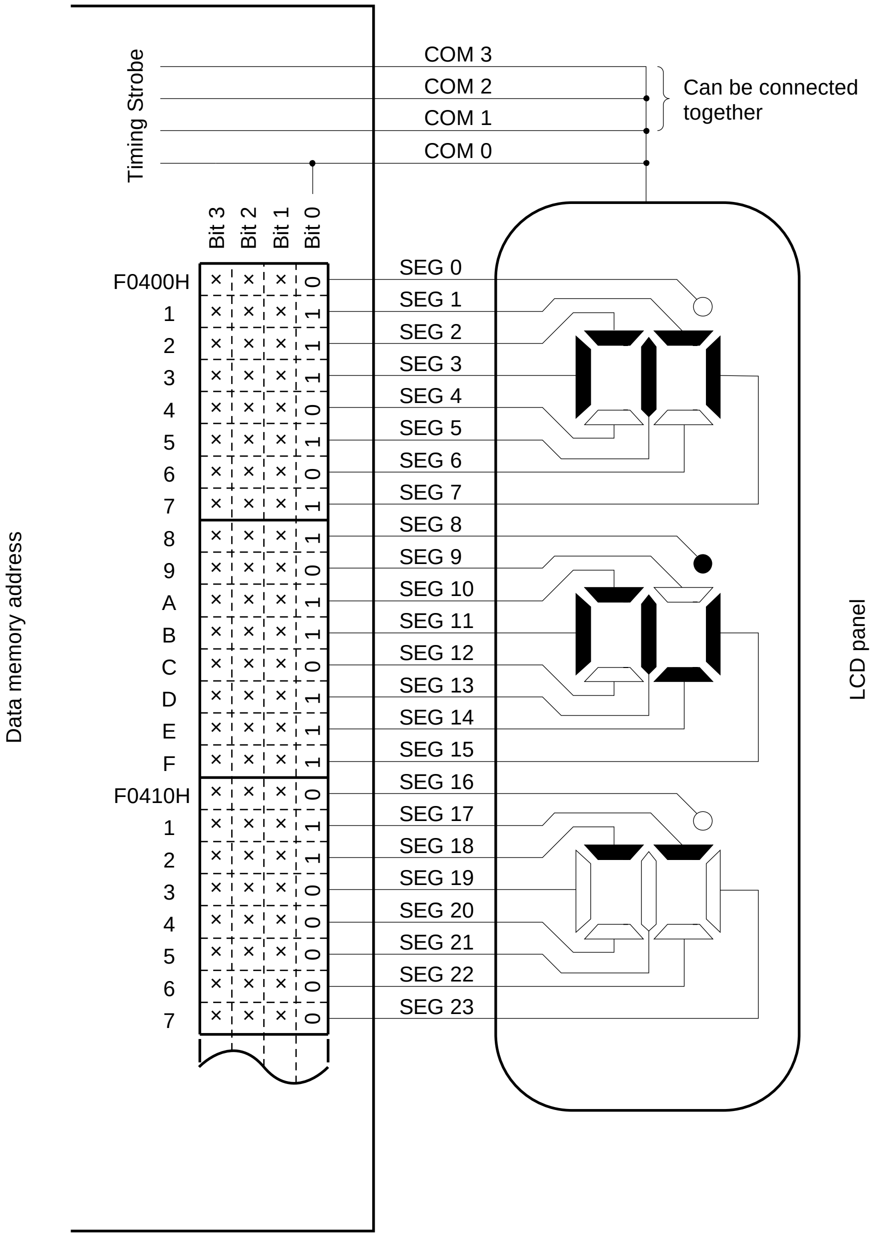
COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 24-25. Static LCD Display Pattern and Electrode Connections



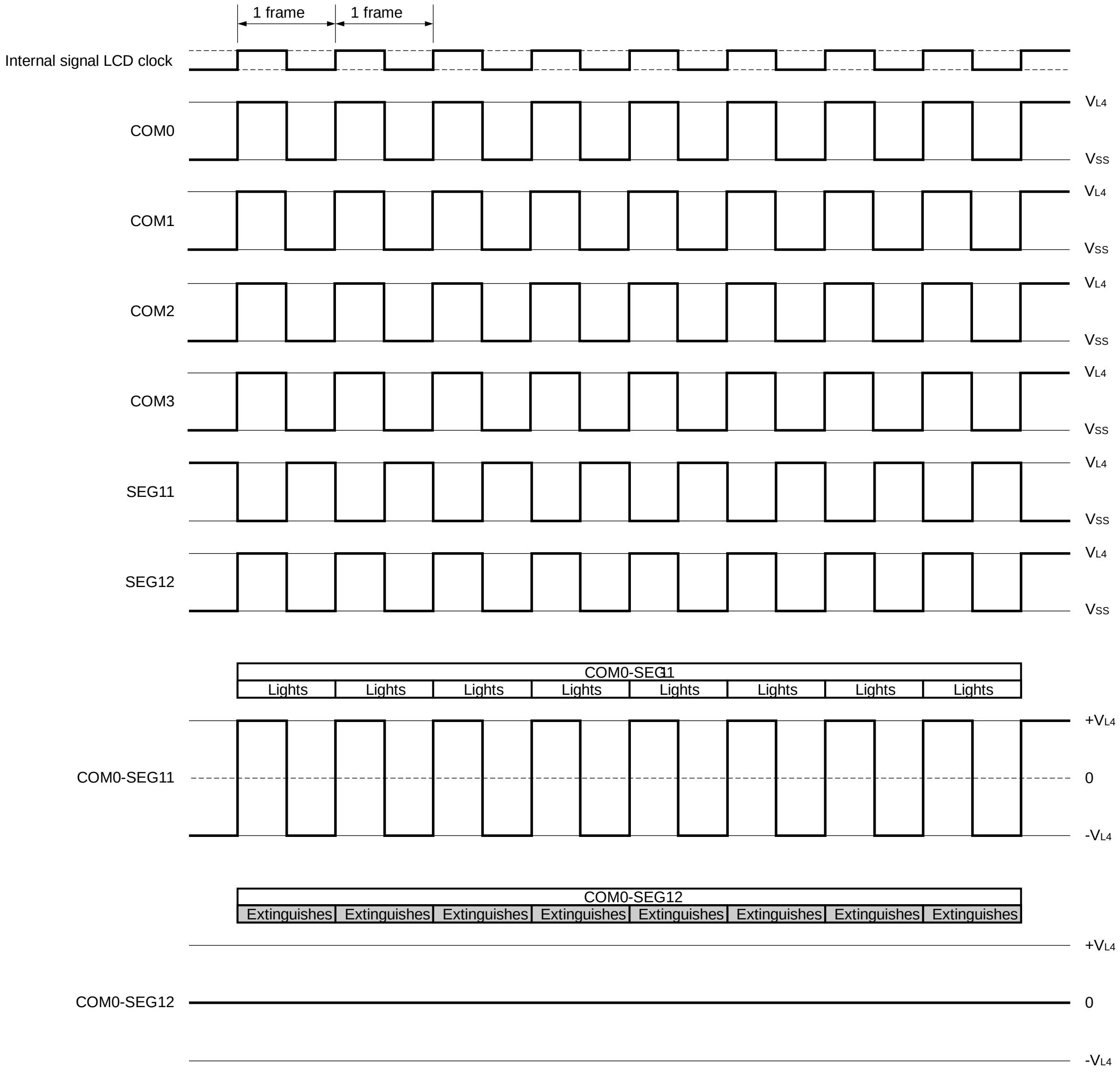
Remark 100-pin product: n = 0 to 4

Figure 24-26. Example of Connecting Static LCD Panel



	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	×	×	×	0
1	×	×	×	1
2	×	×	×	1
3	×	×	×	1
4	×	×	×	0
5	×	×	×	1
6	×	×	×	0
7	×	×	×	1
8	×	×	×	1
9	×	×	×	0
A	×	×	×	1
B	×	×	×	1
C	×	×	×	0
D	×	×	×	1
E	×	×	×	1
F	×	×	×	1
F0410H	×	×	×	0
1	×	×	×	1
2	×	×	×	1
3	×	×	×	0
4	×	×	×	0
5	×	×	×	0
6	×	×	×	0
7	×	×	×	0

Figure 24-27. Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0



24.10.2 Two-time-slice display example

Figure 24-29 shows how the 6-digit LCD panel having the display pattern shown in **Figure 24-28** 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 24-16** at the timing of the common signals COM0 and COM1; see **Figure 24-28** for the relationship between the segment signals and LCD segments.

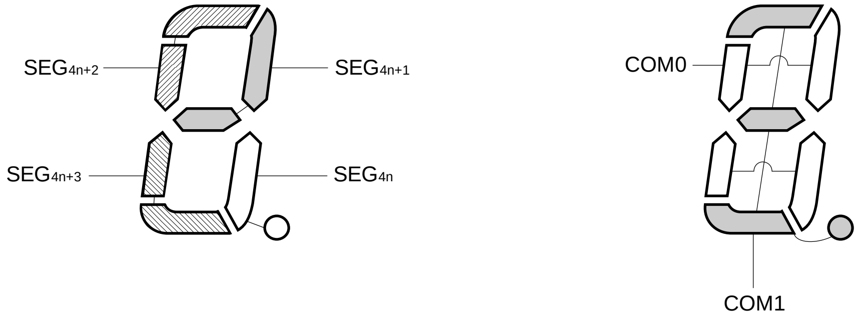
Table 24-16. Select and Deselect Voltages (COM0 and COM1)

Segment / Common	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to **Table 24-16**, it is determined that the display data register location (F040FH) that corresponds to SEG15 must contain xx10.

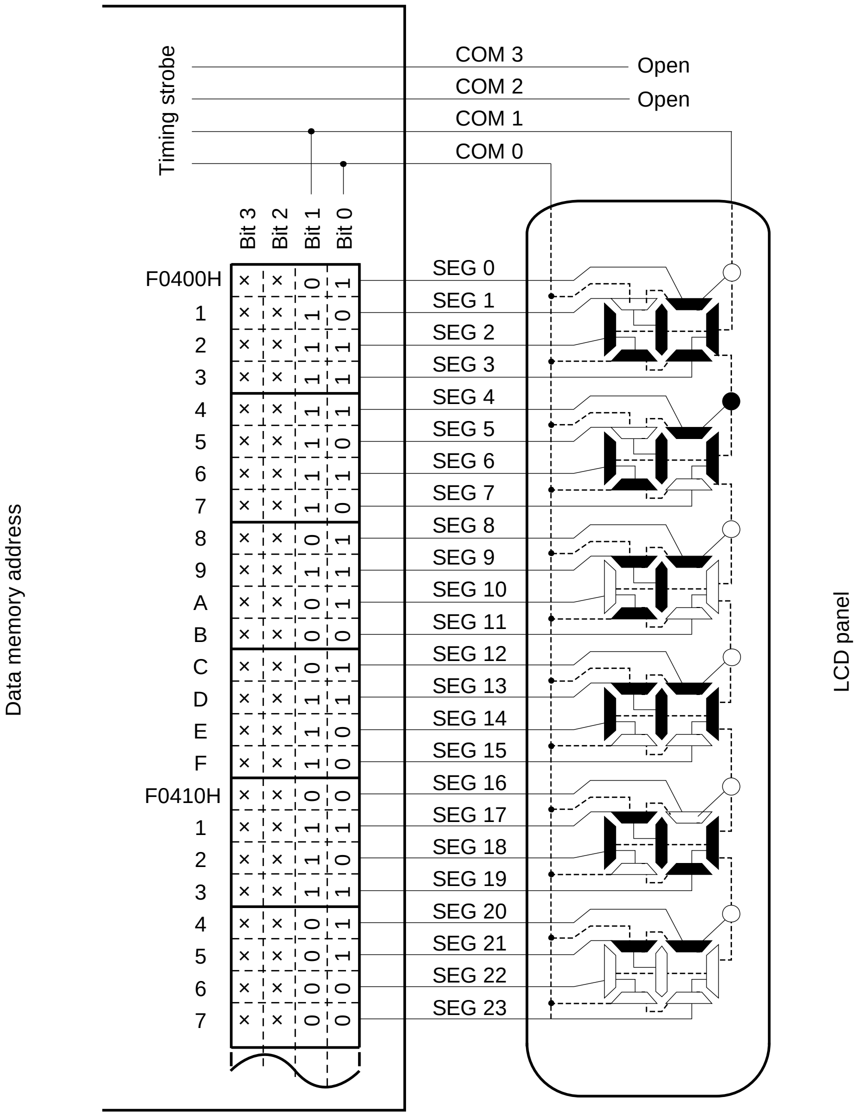
Figure 24-30 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +V$_{LCD}$/–V$_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 24-28. Two-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin product: n = 0 to 9

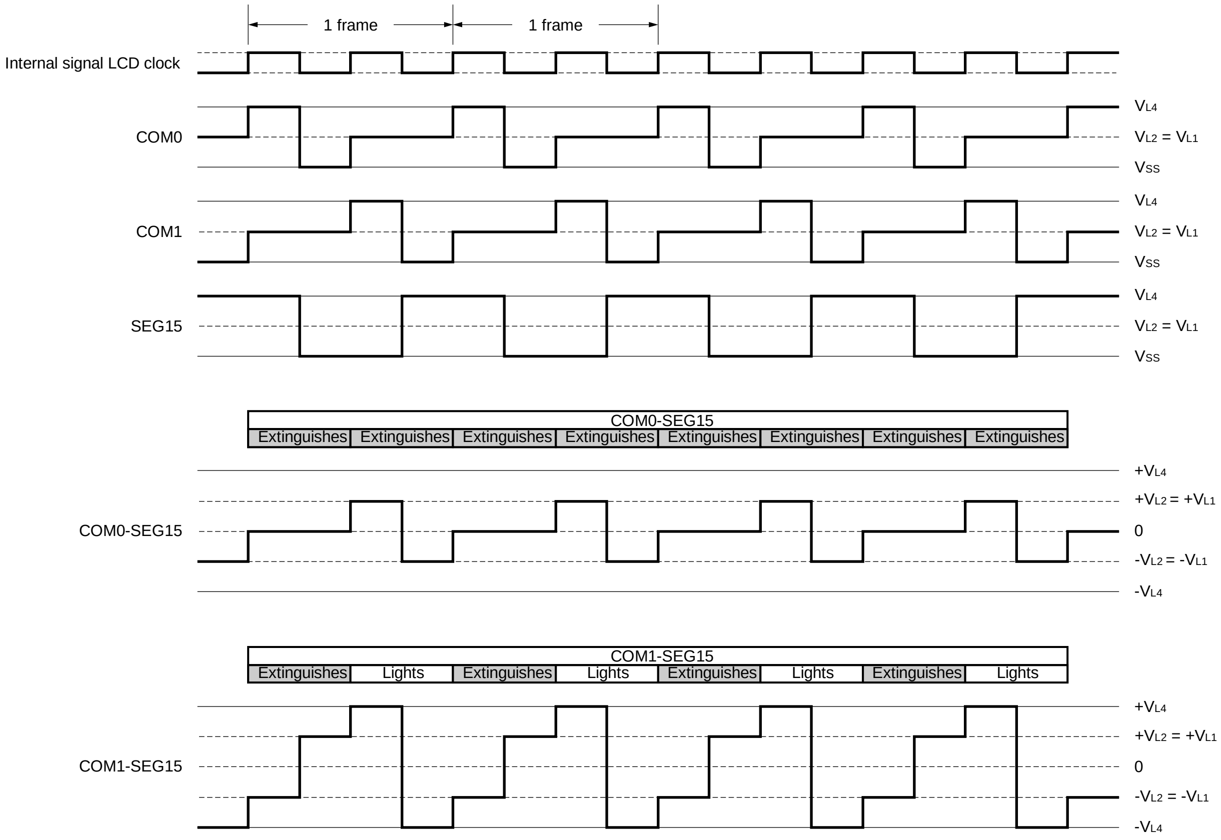
Figure 24-29. Example of Connecting Two-Time-Slice LCD Panel



Data memory address	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	×	×	0	1
1	×	×	1	0
2	×	×	1	1
3	×	×	1	1
4	×	×	1	1
5	×	×	1	0
6	×	×	1	1
7	×	×	1	0
8	×	×	0	1
9	×	×	1	1
A	×	×	0	1
B	×	×	0	0
C	×	×	0	1
D	×	×	1	1
E	×	×	1	0
F	×	×	1	0
F0410H	×	×	0	0
1	×	×	1	1
2	×	×	1	0
3	×	×	1	1
4	×	×	0	1
5	×	×	0	1
6	×	×	0	0
7	×	×	0	0

×: Can always be used to store any data because the two-time-slice mode is being used.

Figure 24-30. Two-Time-Slice LCD Drive Waveform Examples Between SEG15 and Each Common Signals (1/2 Bias Method)



24.10.3 Three-time-slice display example

Figure 24-32 shows how the 8-digit LCD panel having the display pattern shown in **Figure 24-31** 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 24-17** at the timing of the common signals COM0 to COM2; see **Figure 24-31** for the relationship between the segment signals and LCD segments.

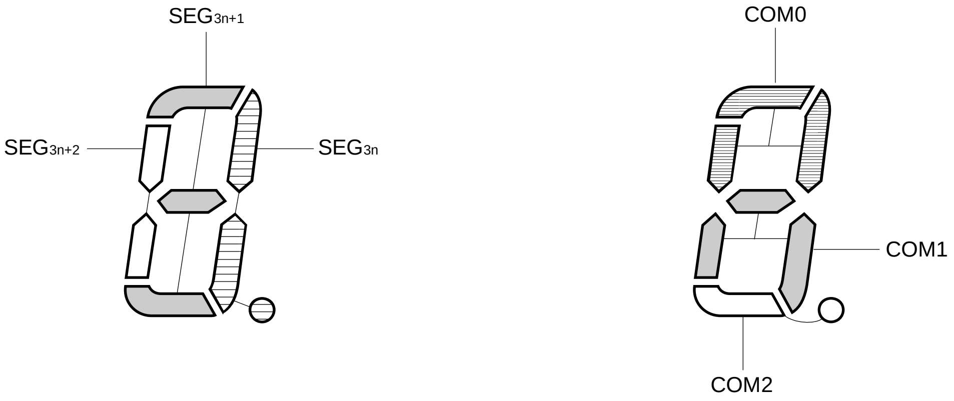
Table 24-17. Select and Deselect Voltages (COM0 to COM2)

Segment / Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to **Table 24-17**, it is determined that the display data register location (F0406H) that corresponds to SEG6 must contain x110.

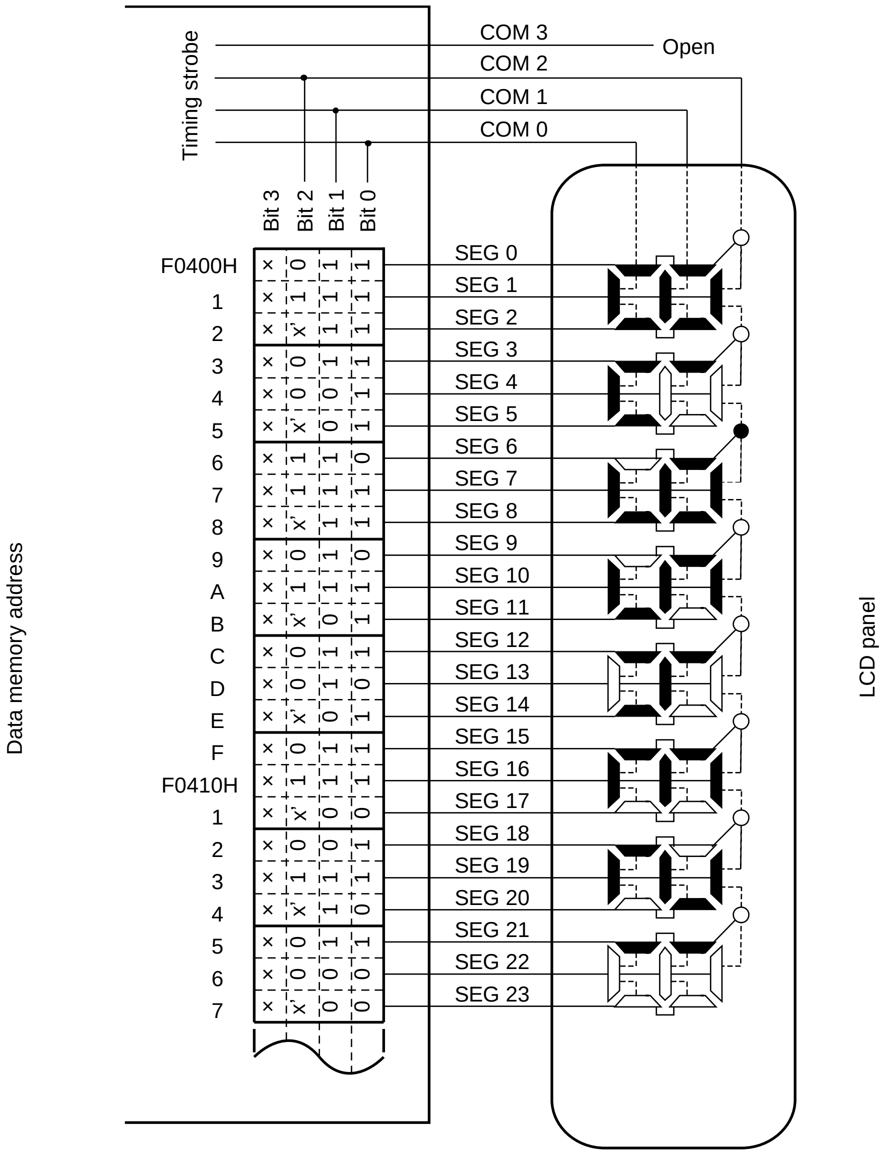
Figures 24-33 and **24-34** show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +V_{LCD}/–V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 24-31. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin product: n = 0 to 13

Figure 24-32. Example of Connecting Three-Time-Slice LCD Panel



Address	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	×	0	1	1
1	×	1	1	1
2	×	×'	1	1
3	×	0	1	1
4	×	0	0	1
5	×	×'	0	1
6	×	1	1	0
7	×	1	1	1
8	×	×'	1	1
9	×	0	1	0
A	×	1	1	1
B	×	×'	0	1
C	×	0	1	1
D	×	0	1	0
E	×	×'	0	1
F	×	0	1	1
F0410H	×	1	1	1
1	×	×'	0	0
2	×	0	0	1
3	×	1	1	1
4	×	×'	1	0
5	×	0	1	1
6	×	0	0	0
7	×	×'	0	0

×': 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 24-33. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/2 Bias Method)

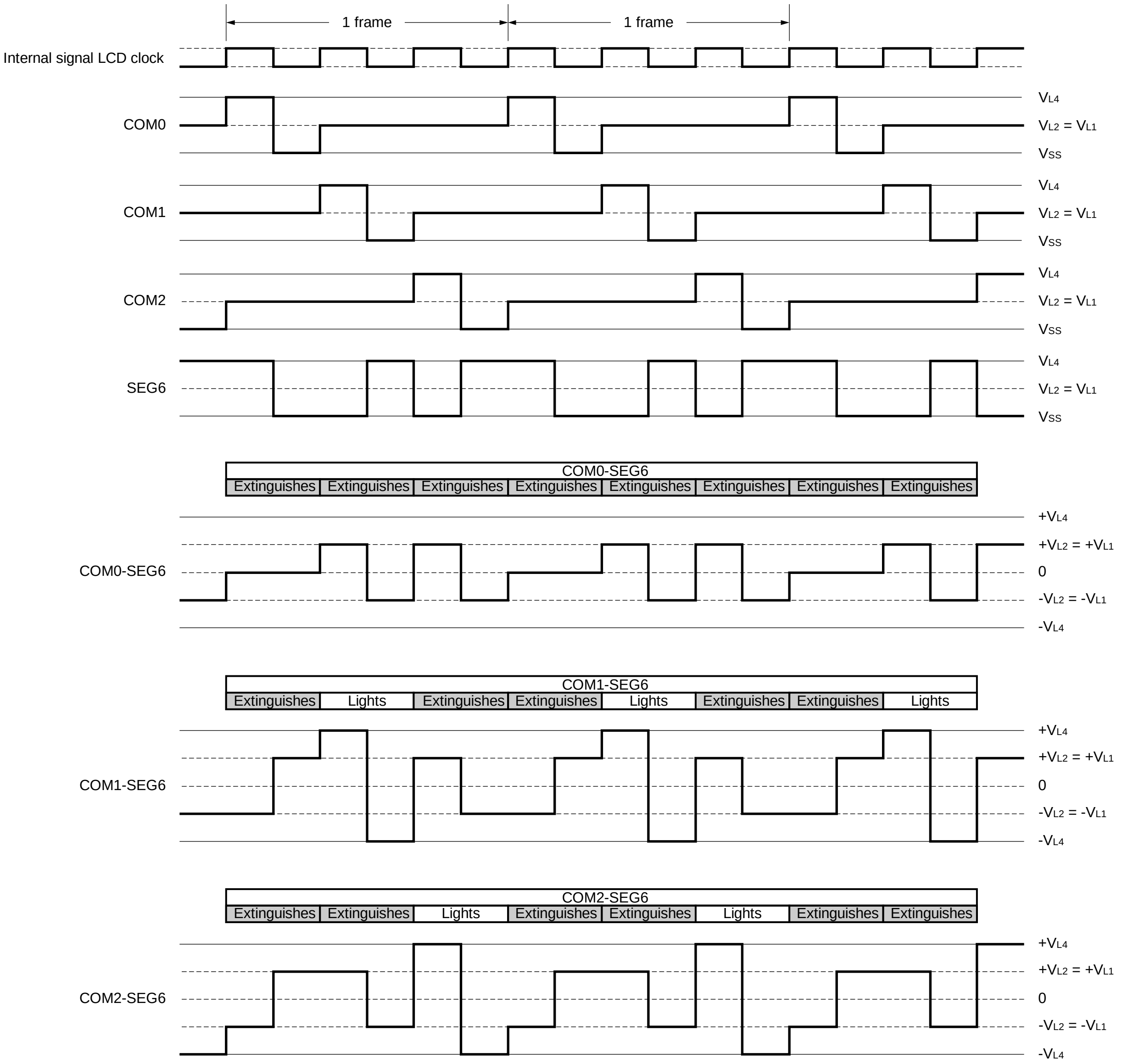
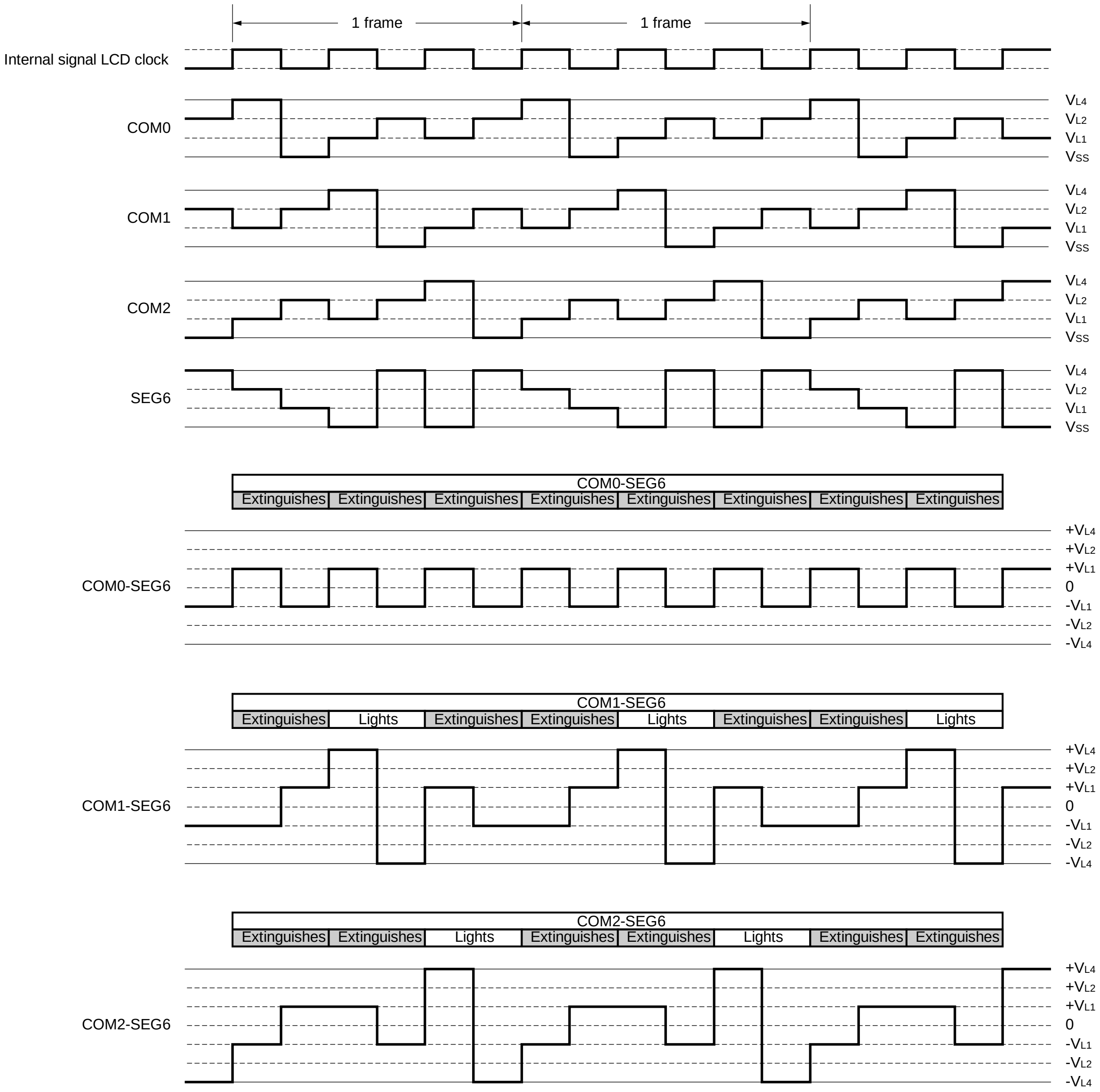

1 frame
1 frame
Internal signal LCD clock
COM0
COM1
COM2
SEG6
V_{L4}
$V_{L2} = V_{L1}$
V_{SS}
COM0-SEG6
Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes
COM1-SEG6
Extinguishes | Lights | Extinguishes | Extinguishes | Lights | Extinguishes | Extinguishes | Lights
COM2-SEG6
Extinguishes | Extinguishes | Lights | Extinguishes | Extinguishes | Lights | Extinguishes | Extinguishes
$+V_{L4}$
$+V_{L2} = +V_{L1}$
0
$-V_{L2} = -V_{L1}$
$-V_{L4}$

Figure 24-34. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/3 Bias Method)


1 frame
1 frame
Internal signal LCD clock
COM0
COM1
COM2
SEG6
VL4
VL2
VL1
VSS
COM0-SEG6
Extinguishes Extinguishes Extinguishes Extinguishes Extinguishes Extinguishes Extinguishes Extinguishes
+VL4
+VL2
+VL1
0
-VL1
-VL2
-VL4
COM1-SEG6
Extinguishes Lights Extinguishes Extinguishes Lights Extinguishes Extinguishes Lights
COM2-SEG6
Extinguishes Extinguishes Lights Extinguishes Extinguishes Lights Extinguishes Extinguishes

24.10.4 Four-time-slice display example

Figure 24-36 shows how the 12-digit LCD panel having the display pattern shown in **Figure 24-35** 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 24-18** at the timing of the common signals COM0 to COM3; see **Figure 24-35** for the relationship between the segment signals and LCD segments.

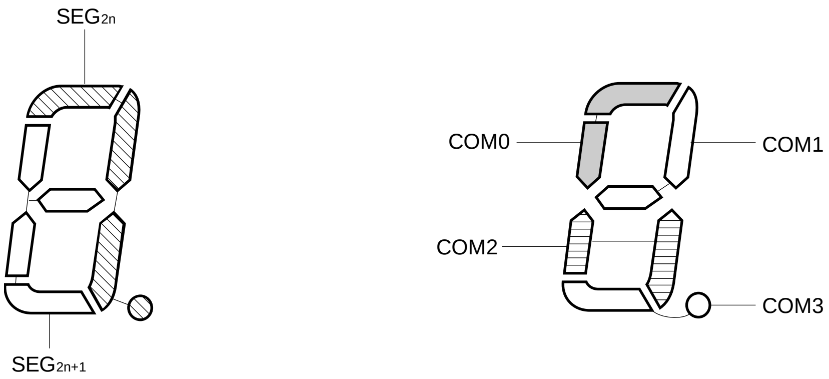
Table 24-18. Select and Deselect Voltages (COM0 to COM3)

Segment Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to **Table 24-18**, it is determined that the display data register location (F040CH) that corresponds to SEG12 must contain 1101.

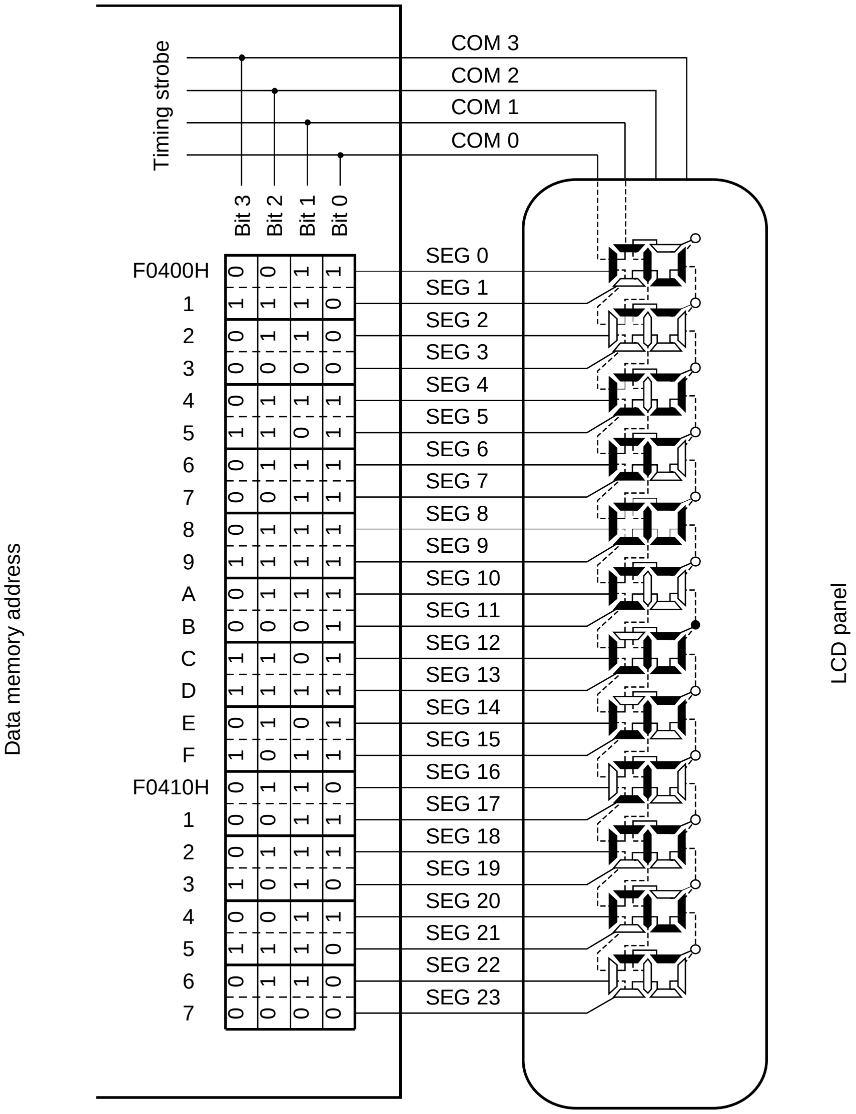
Figure 24-37 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}$/$–V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 24-35. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin product: n = 0 to 20

Figure 24-36. Example of Connecting Four-Time-Slice LCD Panel



	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	0	0	1	1
1	1	1	1	0
2	0	1	1	0
3	0	0	0	0
4	0	1	1	1
5	1	1	0	1
6	0	1	1	1
7	0	0	1	1
8	0	1	1	1
9	1	1	1	1
A	0	1	1	1
B	0	0	0	1
C	1	1	0	1
D	1	1	1	1
E	0	1	0	1
F	1	0	1	1
F0410H	0	1	1	0
1	0	0	1	1
2	0	1	1	1
3	1	0	1	0
4	0	0	1	1
5	1	1	1	0
6	0	1	1	0
7	0	0	0	0

Figure 24-37. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (1/2)

(a) Waveform A

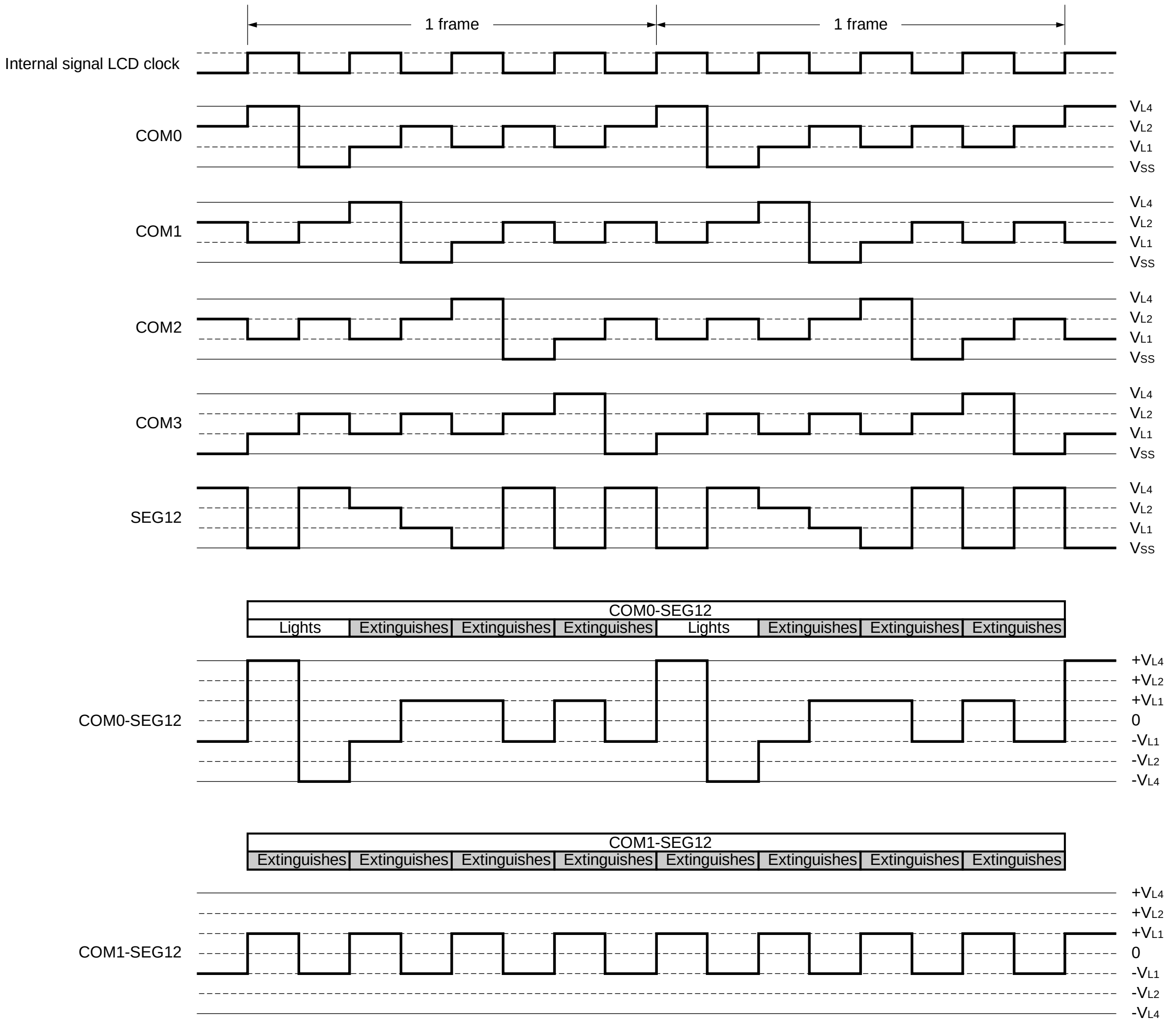
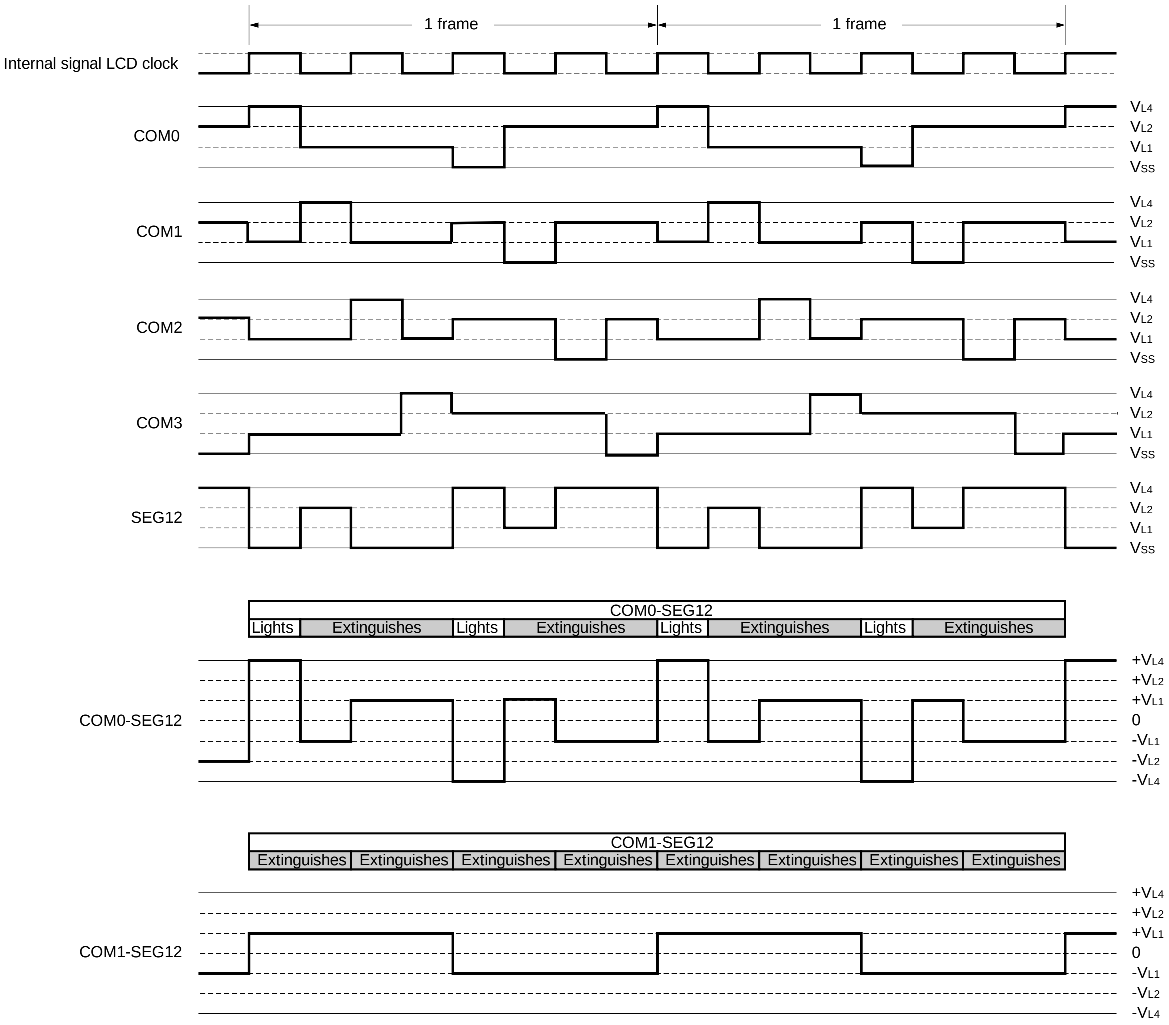

1 frame
1 frame
Internal signal LCD clock
COM0
COM1
COM2
COM3
SEG12
VL4
VL2
VL1
VSS
COM0-SEG12
Lights | Extinguishes | Extinguishes | Extinguishes | Lights | Extinguishes | Extinguishes | Extinguishes
COM0-SEG12
+VL4
+VL2
+VL1
0
-VL1
-VL2
-VL4
COM1-SEG12
Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes | Extinguishes
COM1-SEG12

Figure 24-37. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (2/2)

(b) Waveform B



24.10.5 Six-time-slice display example

Figure 24-39 shows how the 15x6 dot LCD panel having the display pattern shown in **Figure 24-38** 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 24-19** at the timing of the common signals COM0 to COM5; see **Figure 24-38** for the relationship between the segment signals and LCD segments.

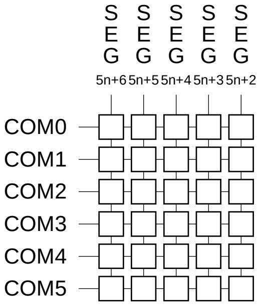
Table 24-19. Select and Deselect Voltages (COM0 to COM5)

Segment Common	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 **Table 24-19**, it is determined that the display data register location (F0402H) that corresponds to SEG2 must contain 010001.

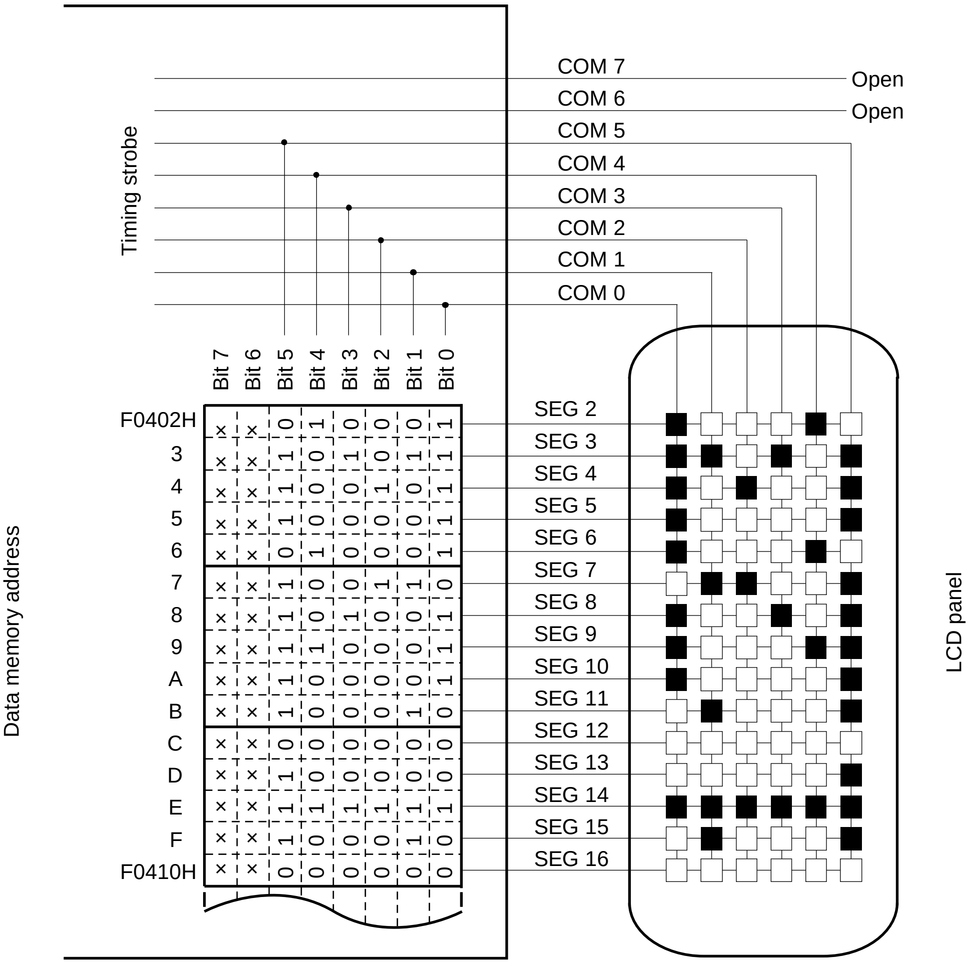
Figure 24-40 shows examples of LCD drive waveforms between the SEG2 signal and each common signal. When the select voltage is applied to SEG2 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 24-38. Six-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin product: n = 0 to 7

Figure 24-39. Example of Connecting Six-Time-Slice LCD Panel

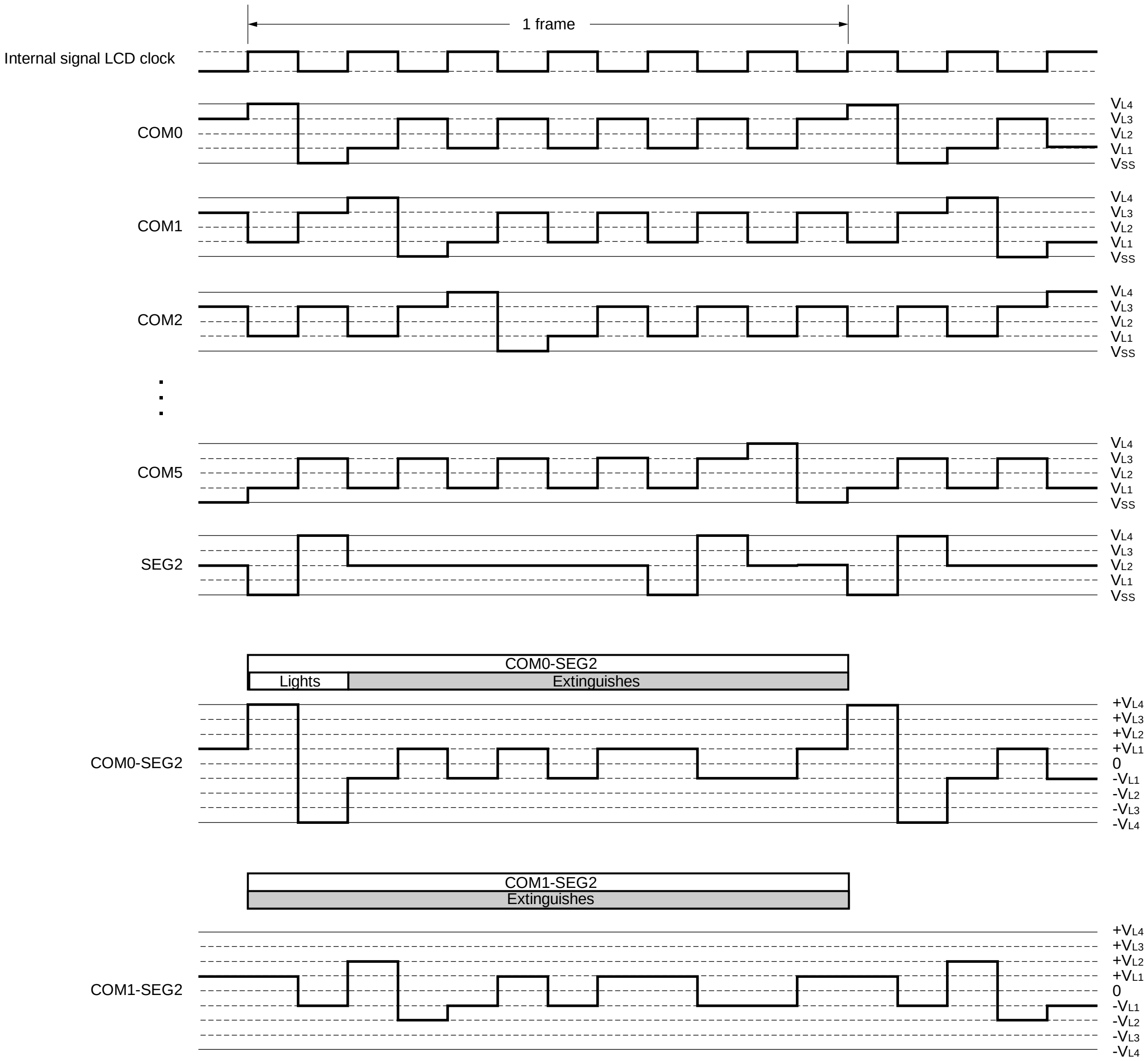


Data memory address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F0402H	×	×	0	1	0	0	0	1
3	×	×	1	0	1	0	1	1
4	×	×	1	0	0	1	0	1
5	×	×	1	0	0	0	0	1
6	×	×	0	1	0	0	0	1
7	×	×	1	0	0	1	1	0
8	×	×	1	0	1	0	0	1
9	×	×	1	1	0	0	0	1
A	×	×	1	0	0	0	0	1
B	×	×	1	0	0	0	1	0
C	×	×	0	0	0	0	0	0
D	×	×	1	0	0	0	0	0
E	×	×	1	1	1	1	1	1
F	×	×	1	0	0	0	1	0
F0410H	×	×	0	0	0	0	0	0

×: Can always be used to store any data because the six-time-slice mode is being used.

Figure 24-40. Six-Time-Slice LCD Drive Waveform Examples Between SEG2 and Each Common Signals (1/4 Bias Method)

(a) Waveform A



24.10.6 Eight-time-slice display example

Figure 24-42 shows how the 15x8 dot LCD panel having the display pattern shown in **Figure 24-41** 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 24-20** at the timing of the common signals COM0 to COM7; see **Figure 24-41** for the relationship between the segment signals and LCD segments.

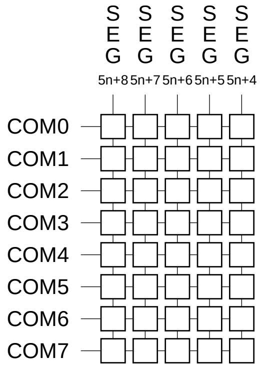
Table 24-20. Select and Deselect Voltages (COM0 to COM7)

Segment Common	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 **Table 24-20**, it is determined that the display data register location (F0404H) that corresponds to SEG4 must contain 00110001.

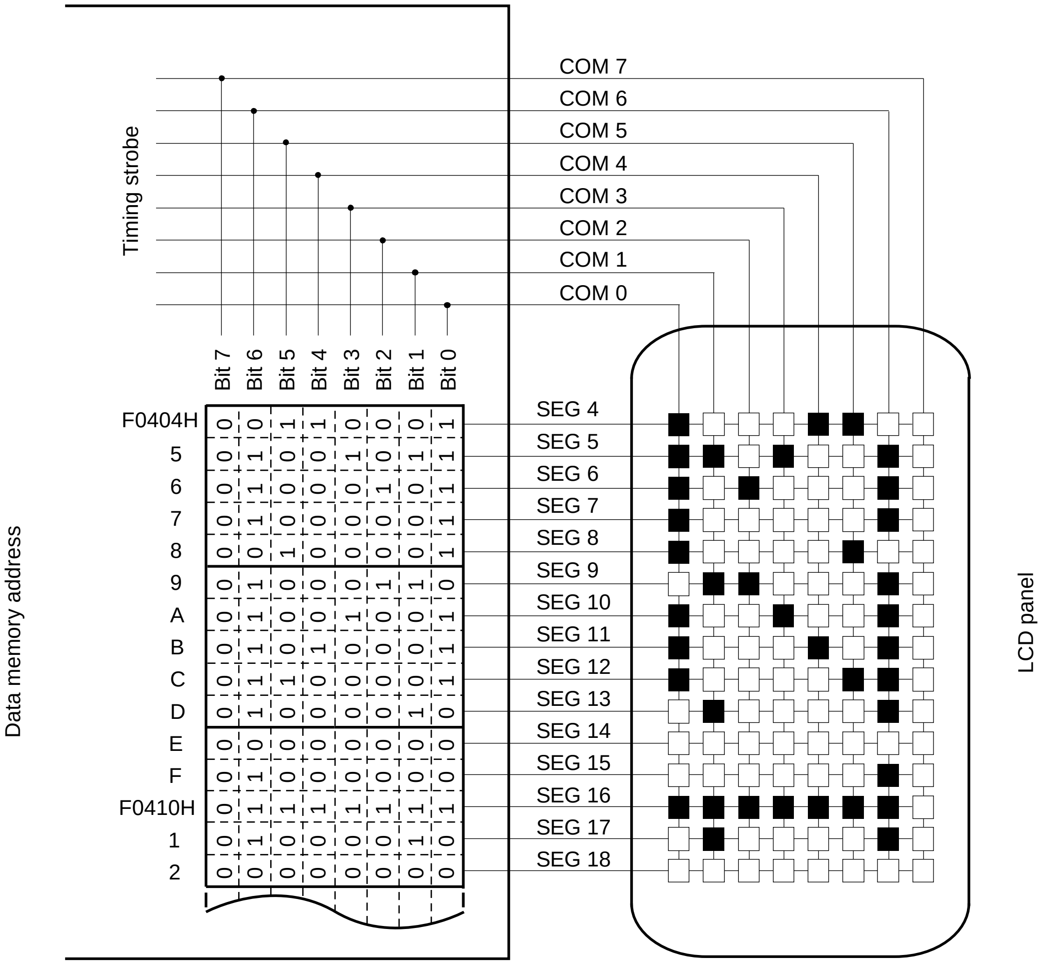
Figure 24-43 shows examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 24-41. Eight-Time-Slice LCD Display Pattern and Electrode Connections



Remark 100-pin product: n = 0 to 6

Figure 24-42. Example of Connecting Eight-Time-Slice LCD Panel



Data memory address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F0404H	0	0	1	1	0	0	0	1
5	0	1	0	0	1	0	1	1
6	0	1	0	0	0	1	0	1
7	0	1	0	0	0	0	0	1
8	0	0	1	0	0	0	0	1
9	0	1	0	0	0	1	1	0
A	0	1	0	0	1	0	0	1
B	0	1	0	1	0	0	0	1
C	0	1	1	0	0	0	0	1
D	0	1	0	0	0	0	1	0
E	0	0	0	0	0	0	0	0
F	0	1	0	0	0	0	0	0
F0410H	0	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0

Figure 24-43. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (1/2)

(a) Waveform A

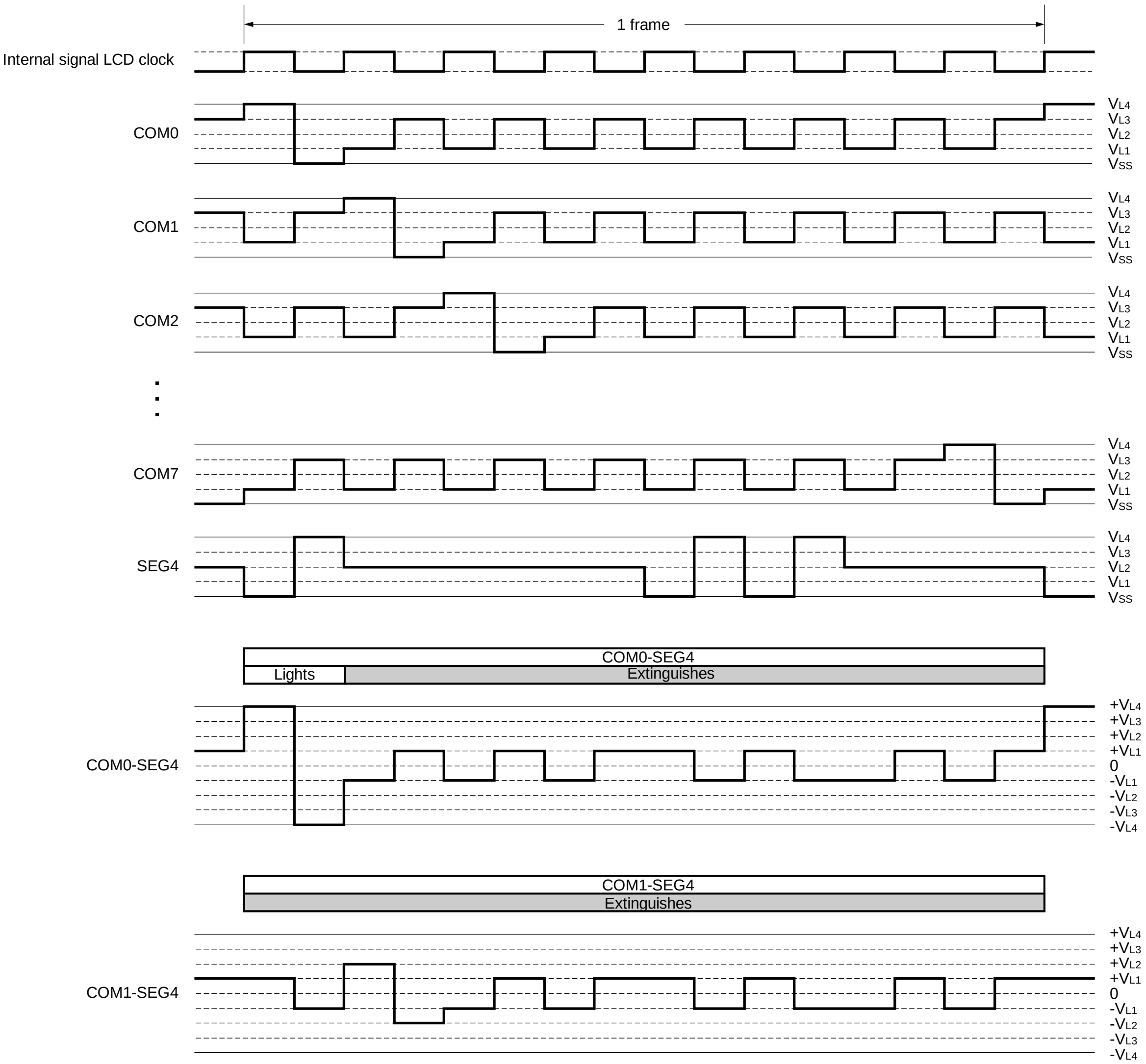
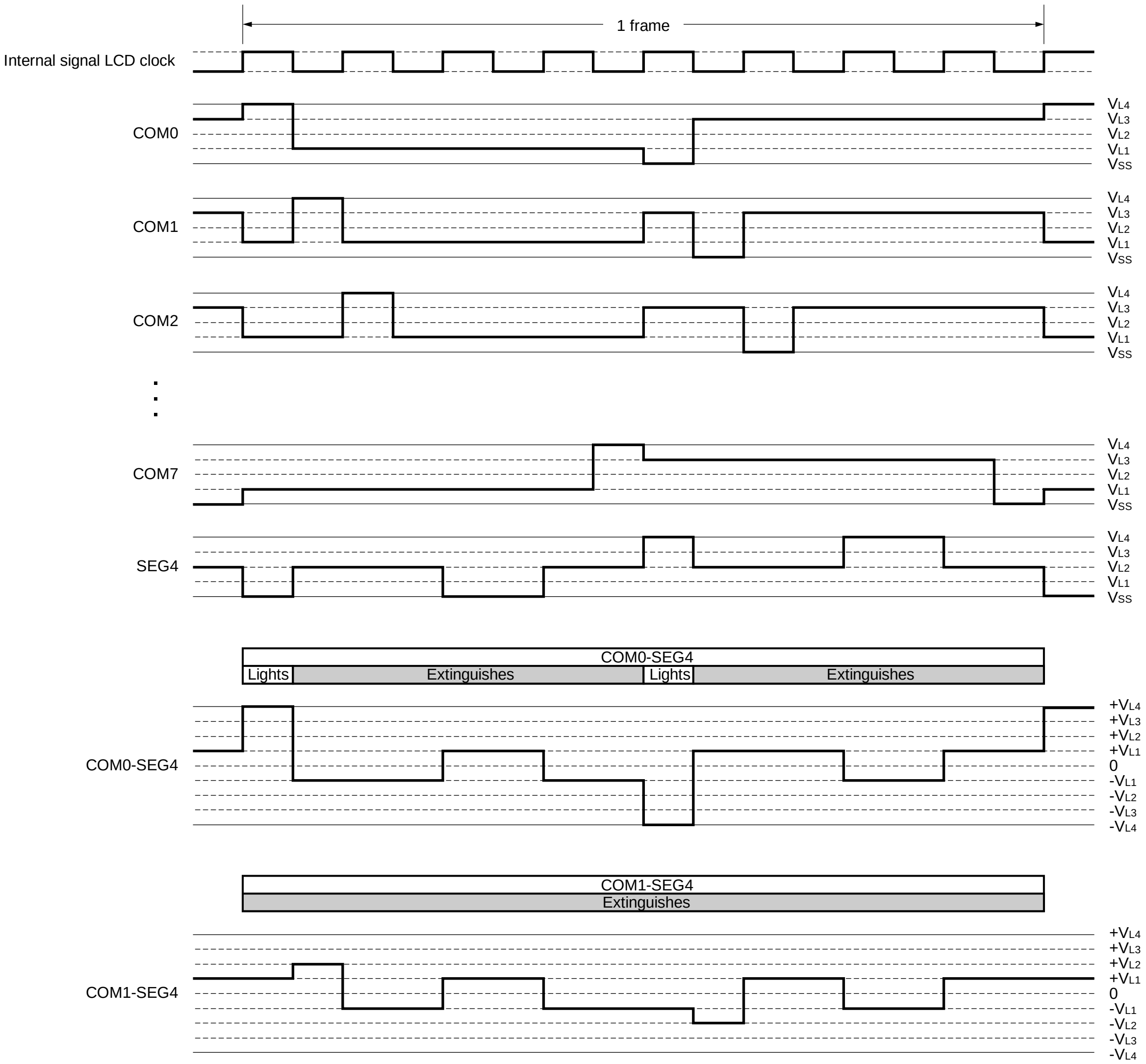


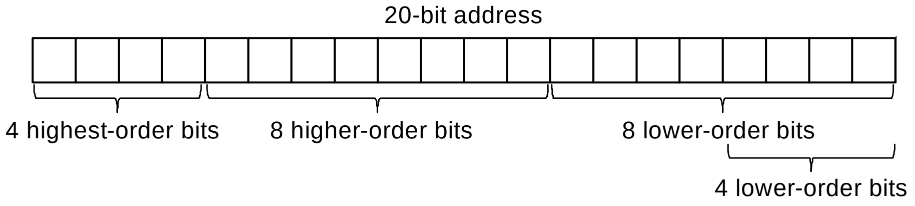
Figure 24-43. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (2/2)

(b) Waveform B



CHAPTER 25 DATA TRANSFER CONTROLLER (DTC)

The term “8 higher-order bits of the address” in this chapter 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).

25.1 Functions of DTC

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 25-1 lists the DTC specifications.

Table 25-1. DTC Specifications

Item		Specification
Activation sources		80-pin product: 46 sources, 100-pin product: 50 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), mirror area[Note], data flash memory area [Note], extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
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		See **Table 25-5 DTC Activation Sources and Vector Addresses**.
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 transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	● When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). ● When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	● When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). ● When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

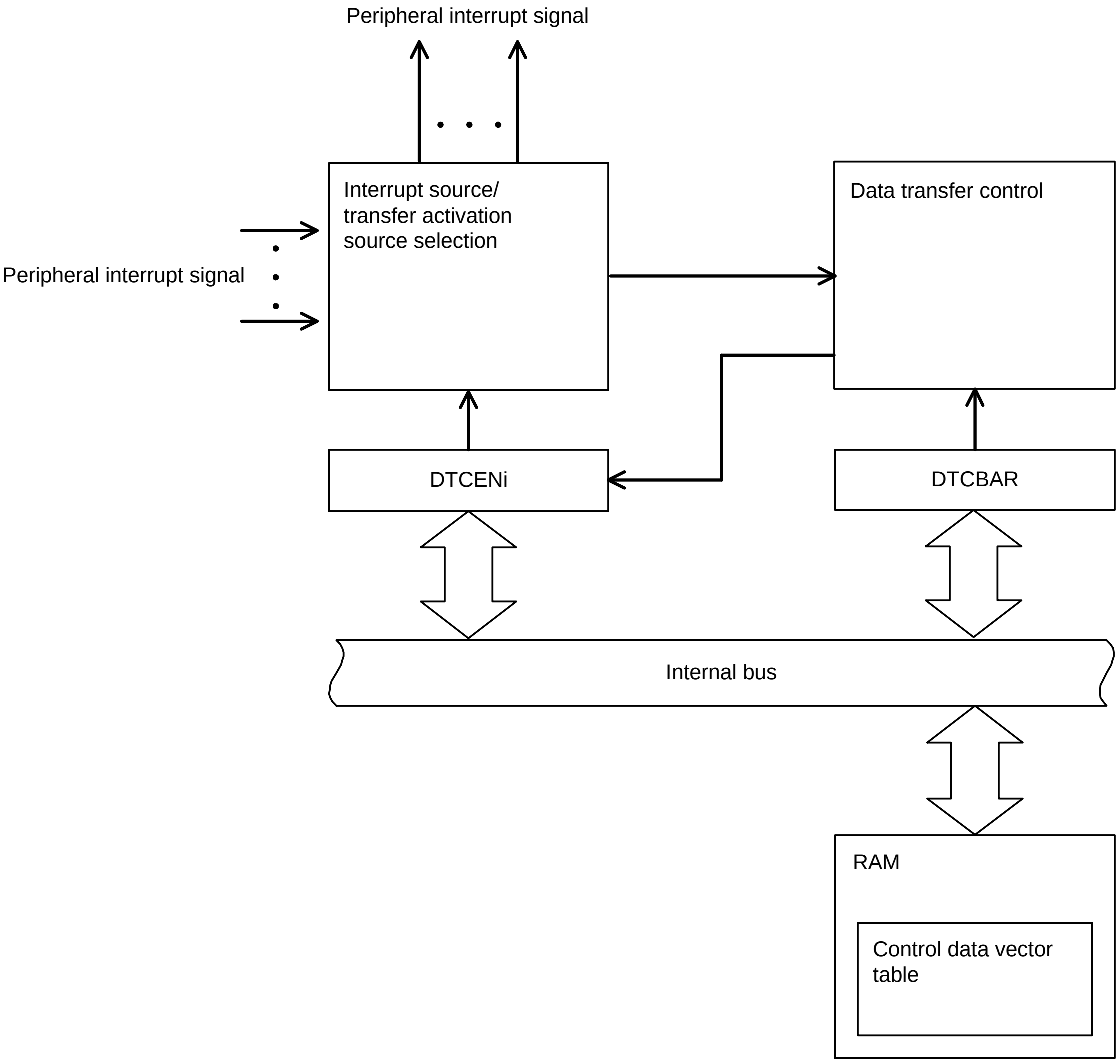
Note In the HALT and SNOOZE modes, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 6, j = 0 to 23

25.2 Configuration of DTC

Figure 25-1 shows the DTC block diagram.

Figure 25-1. DTC Block Diagram


Peripheral interrupt signal
Interrupt source/ transfer activation source selection
Data transfer control
Peripheral interrupt signal
DTCENi
DTCBAR
Internal bus
RAM
Control data vector table

25.3 Registers Controlling DTC

Table 25-2 lists the registers controlling DTC.

Table 25-2. Registers Controlling DTC

Register Name	Symbol
Peripheral Enable Register 1	PER1
DTC Activation Enable Register 0	DTCEN0
DTC Activation Enable Register 1	DTCEN1
DTC Activation Enable Register 2	DTCEN2
DTC Activation Enable Register 3	DTCEN3
DTC Activation Enable Register 4	DTCEN4
DTC Activation Enable Register 5	DTCEN5
DTC Activation Enable Register 6	DTCEN6
DTC Base Address Register	DTCBAR

Table 25-3 lists DTC control data.

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.

Table 25-3. DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

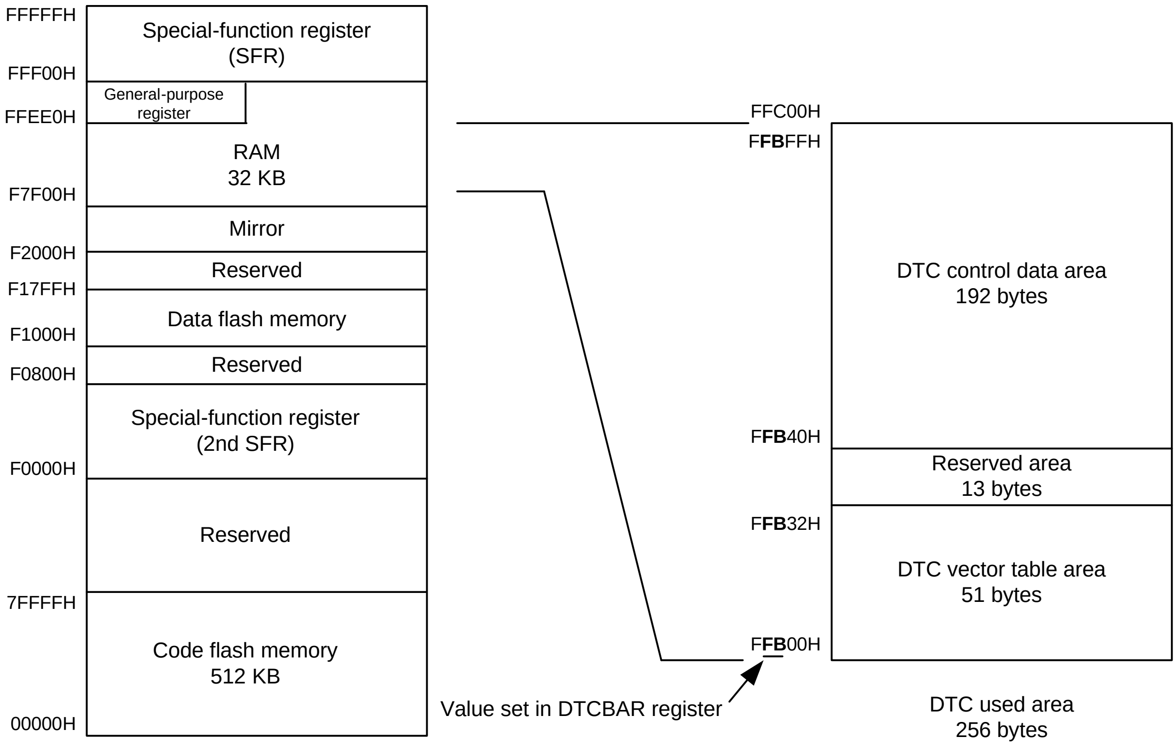
25.3.1 Allocation of DTC control data area and DTC vector table area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 25-2 shows a memory map example when 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 25-2. Memory Map Example When DTCBAR Register Is Set to FBH (R5F10NPLDFB, R5F10NMLDFB)



The areas where the DTC control data and vector table can be allocated differ depending on the product.

Cautions 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.

2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

3. The following RAM area cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.

F7F00H to F8309H

4. The following RAM area cannot be used as the DTC control data area or DTC vector table area when using the trace function of on-chip debugging.

F8300H to F86FFH

25.3.2 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).

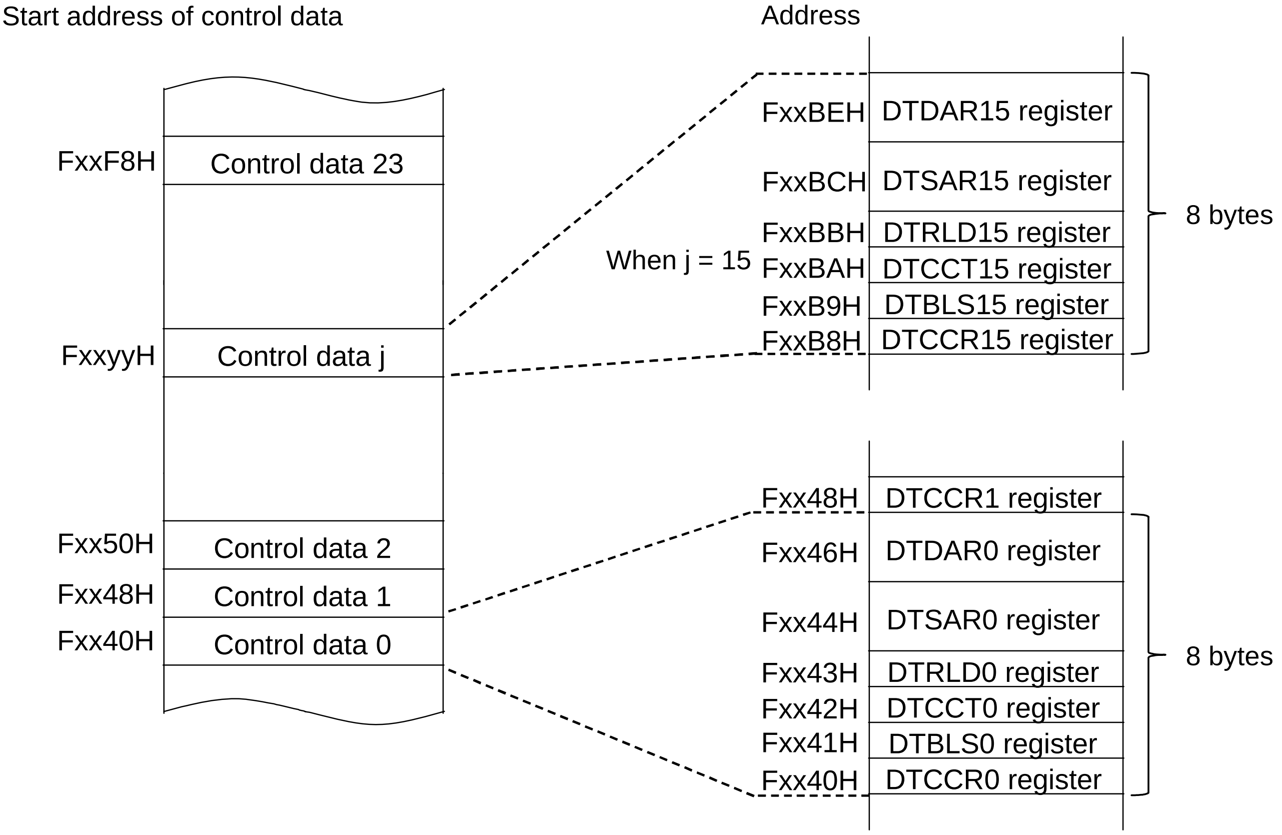
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.

Figure 25-3 shows control data allocation.

Cautions 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 6) in the DTCENi register is set to 0 (DTC activation disabled).

2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 25-3. Control Data Allocation



Remark xx: Value set in DTCBAR register

Table 25-4. Start Address of Control Data

j	Address	j	Address
11	Fxx98H	23	FxxF8H
10	Fxx90H	22	FxxF0H
9	Fxx88H	21	FxxE8H
8	Fxx80H	20	FxxE0H
7	Fxx78H	19	FxxD8H
6	Fxx70H	18	FxxD0H
5	Fxx68H	17	FxxC8H
4	Fxx60H	16	FxxC0H
3	Fxx58H	15	FxxB8H
2	Fxx50H	14	FxxB0H
1	Fxx48H	13	FxxA8H
0	Fxx40H	12	FxxA0H

Remark xx: Value set in DTCBAR register

25.3.3 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 25-5 lists the 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 higher 8 bits for the vector address are set by the DTCBAR register, and 00H to 32H are allocated to the lower 8 bits corresponding to the activation source.

Caution Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 6) in the DTCENi register is set to 0 (activation disabled).

Figure 25-4. Start Address of Control Data and Vector Table

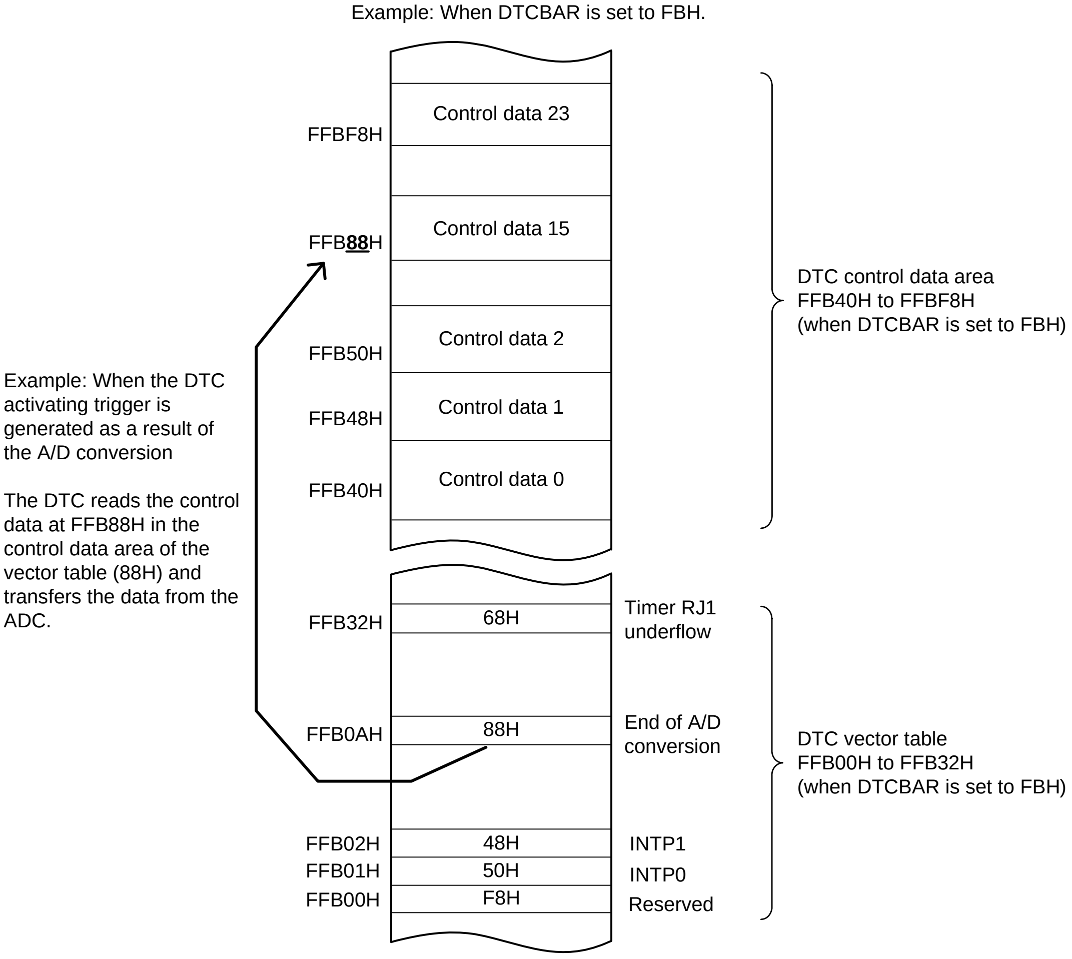

Example: When DTCBAR is set to FBH.
FFBF8H
Control data 23
FFB88H
Control data 15
FFB50H
Control data 2
FFB48H
Control data 1
FFB40H
Control data 0
DTC control data area
FFB40H to FFBF8H
(when DTCBAR is set to FBH)
Example: When the DTC activating trigger is generated as a result of the A/D conversion
The DTC reads the control data at FFB88H in the control data area of the vector table (88H) and transfers the data from the ADC.
FFB32H
68H
Timer RJ1 underflow
FFB0AH
88H
End of A/D conversion
DTC vector table
FFB00H to FFB32H
(when DTCBAR is set to FBH)
FFB02H
48H
INTP1
FFB01H
50H
INTP0
FFB00H
F8H
Reserved

Table 25-5. DTC Activation Sources and Vector Addresses (1/2)

Interrupt Request Source	Source No.	Vector Address	Priority
Reserved	0	Address set in DTCBAR register +00H	Highest
INTP0	1	Address set in DTCBAR register +01H	
INTP1	2	Address set in DTCBAR register +02H	
INTP2	3	Address set in DTCBAR register +03H	
INTP3	4	Address set in DTCBAR register +04H	
INTP4	5	Address set in DTCBAR register +05H	
INTP5	6	Address set in DTCBAR register +06H	
INTP6	7	Address set in DTCBAR register +07H	
INTP7	8	Address set in DTCBAR register +08H	
Key return signal detection	9	Address set in DTCBAR register +09H	
24-bit ΔΣ-type A/D converter	10	Address set in DTCBAR register +0AH	
12-bit successive-approximation A/D scan end	11	Address set in DTCBAR register +0BH	
UART0 reception transfer end	12	Address set in DTCBAR register +0CH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	13	Address set in DTCBAR register +0DH	
UART1 reception transfer end	14	Address set in DTCBAR register +0EH	
UART1 transmission transfer end/CSI10 transfer end or IIC10 transfer end	15	Address set in DTCBAR register +0FH	
UART2 reception transfer end	16	Address set in DTCBAR register +10H	
UART2 transmission transfer end	17	Address set in DTCBAR register +11H	
UART3 reception transfer end [Note]	18	Address set in DTCBAR register +12H	
UART3 transmission transfer end/CSI30 transfer end or IIC30 transfer end [Note]	19	Address set in DTCBAR register +13H	
End of channel 0 of timer array unit 0 count or capture	20	Address set in DTCBAR register +14H	
End of channel 1 of timer array unit 0 count or capture	21	Address set in DTCBAR register +15H	
End of channel 2 of timer array unit 0 count or capture	22	Address set in DTCBAR register +16H	
End of channel 3 of timer array unit 0 count or capture	23	Address set in DTCBAR register +17H	
End of channel 4 of timer array unit 0 count or capture	24	Address set in DTCBAR register +18H	
End of channel 5 of timer array unit 0 count or capture	25	Address set in DTCBAR register +19H	
End of channel 6 of timer array unit 0 count or capture	26	Address set in DTCBAR register +1AH	
End of channel 7 of timer array unit 0 count or capture	27	Address set in DTCBAR register +1BH	
8-bit interval timer 00	28	Address set in DTCBAR register +1CH	
8-bit interval timer 01	29	Address set in DTCBAR register +1DH	
8-bit interval timer 10	30	Address set in DTCBAR register +1EH	
8-bit interval timer 11	31	Address set in DTCBAR register +1FH	
12-bit interval timer detection	32	Address set in DTCBAR register +20H	
AES encryption/decryption end	33	Address set in DTCBAR register +21H	
External interrupt (RTCIC2)/INTP12	34	Address set in DTCBAR register +22H	
External interrupt (RTCIC1)/INTP13	35	Address set in DTCBAR register +23H	
External interrupt (RTCIC0)/INTP14	36	Address set in DTCBAR register +24H	
INTP8	37	Address set in DTCBAR register +25H	
INTP9	38	Address set in DTCBAR register +26H	Lowest

Note 100-pin product only

Table 25-5. DTC Activation Sources and Vector Addresses (2/2)

Interrupt Request Source	Source No.	Vector Address	Priority
UART4 reception transfer end[Note]	39	Address set in DTCBAR register +27H	Highest
UART4 transmission transfer end[Note]	40	Address set in DTCBAR register +28H	↕
UARTMG0 reception transfer end	41	Address set in DTCBAR register +29H	
UARTMG0 transmission transfer end or buffer empty	42	Address set in DTCBAR register +2AH	
UARTMG1 reception transfer end	43	Address set in DTCBAR register +2BH	
UARTMG1 transmission transfer end or buffer empty	44	Address set in DTCBAR register +2CH	
8-bit interval timer 20	45	Address set in DTCBAR register +2DH	
8-bit interval timer 21	46	Address set in DTCBAR register +2EH	
8-bit interval timer 30	47	Address set in DTCBAR register +2FH	
8-bit interval timer 31	48	Address set in DTCBAR register +30H	
Timer RJ0 underflow	49	Address set in DTCBAR register +31H	
Timer RJ1 underflow	50	Address set in DTCBAR register +32H	Lowest

Note 100-pin product only

25.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When 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.

Reset signal generation clears this register to 00H.

Figure 25-5. Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	0	FMCEN	SMOTD1EN	SMOTD0EN	DTCEN	TRJ1EN	TRJ0EN	DSADCEN

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. ● DTC cannot run.
1	Enables input clock supply. ● DTC can run.

Caution Be sure to clear bit 7 to “0”.

25.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 25-6. Format of DTC Control Register j (DTCCRj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Transfer data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

25.3.6 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 25-7. Format of DTC Block Size Register j (DTBLSj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined 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.

25.3.7 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 25-8. Format of DTC Transfer Count Register j (DTCCTj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined 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.

25.3.8 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 25-9. Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined 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.

25.3.9 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 bit is ignored and the address is handled as an even address.

Figure 25-10. Format of DTC Source Address Register j (DTSARj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTSARj15	DTSARj14	DTSARj13	DTSARj12	DTSARj11	DTSARj10	DTSARj9	DTSARj8	DTSARj7	DTSARj6	DTSARj5	DTSARj4	DTSARj3	DTSARj2	DTSARj1	DTSARj0

Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.

2. Do not access the DTSARj register using a DTC transfer.

25.3.10 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 bit is ignored and the address is handled as an even address.

Figure 25-11. Format of DTC Destination Address Register j (DTDARj)

Address: See **25.3.2 Control data allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTDARj15	DTDARj14	DTDARj13	DTDARj12	DTDARj11	DTDARj10	DTDARj9	DTDARj8	DTDARj7	DTDARj6	DTDARj5	DTDARj4	DTDARj3	DTDARj2	DTDARj1	DTDARj0

Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.

2. Do not access the DTDARj register using a DTC transfer.

25.3.11 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 25-6** lists the correspondence between interrupt sources and bits DTCENi0 to DTCENi7.

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

Cautions 1. Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.

2. Do not access the DTCENi register using a DTC transfer.

Figure 25-12. DTC Activation Enable Register i (DTCENi) (i = 0 to 6)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2),
F02EBH (DTCEN3), F02ECH (DTCEN4), F02EDH (DTCEN5), F02EEH (DTCEN6)

After reset: 00H 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.	

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 25-6. Correspondences Between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	INTP7	Key return signal detection	24-bit ΔΣ-type A/D converter	12-bit successive-approximation A/D scan end	UART0 reception transfer end	UART0 transmission transfer end /CSI00 transfer end or buffer empty /IIC00 transfer end	UART1 reception transfer end	UART1 transmission transfer end /CSI10 transfer end or IIC10 transfer end
DTCEN2	UART2 reception transfer end	UART2 transmission transfer end	UART3 reception transfer end Note	UART3 transmission transfer end /CSI30 transfer end or IIC30 transfer end Note	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
DTCEN3	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	End of channel 7 of timer array unit 0 count or capture	8-bit interval timer 00	8-bit interval timer 01	8-bit interval timer 10	8-bit interval timer 11
DTCEN4	12-bit interval timer detection	AES encryption/ decryption end	External interrupt (RTCIC2) /INTP12	External interrupt (RTCIC1) /INTP13	External interrupt (RTCIC0) /INTP14	INTP8	INTP9	UART4 reception transfer end Note
DTCEN5	UART4 transmission transfer end Note	UARTMG0 reception transfer end	UARTMG0 transmission transfer end or buffer empty	UARTMG1 reception transfer end	UARTMG1 transmission transfer end or buffer empty	8-bit interval timer 20	8-bit interval timer 21	8-bit interval timer 30
DTCEN6	8-bit interval timer 31	Timer RJ0 underflow	Timer RJ1 underflow	Reserved	Reserved	Reserved	Reserved	Reserved

Note 100-pin product only

Remark i = 0 to 6

25.3.12 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 higher 8 bits to generate a 16-bit address.

Cautions 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.

2. Do not rewrite the DTCBAR register more than once.

3. Do not access the DTCBAR register using a DTC transfer.

4. For the allocation of the DTC control data area and the DTC vector table area, see the cautions on 25.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 25-13. Format of DTC Base Address Register (DTCBAR)

Address: F02E0H After reset: FDH R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

25.4 DTC Operation

When the DTC is activated, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back 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.

25.4.1 Activation sources

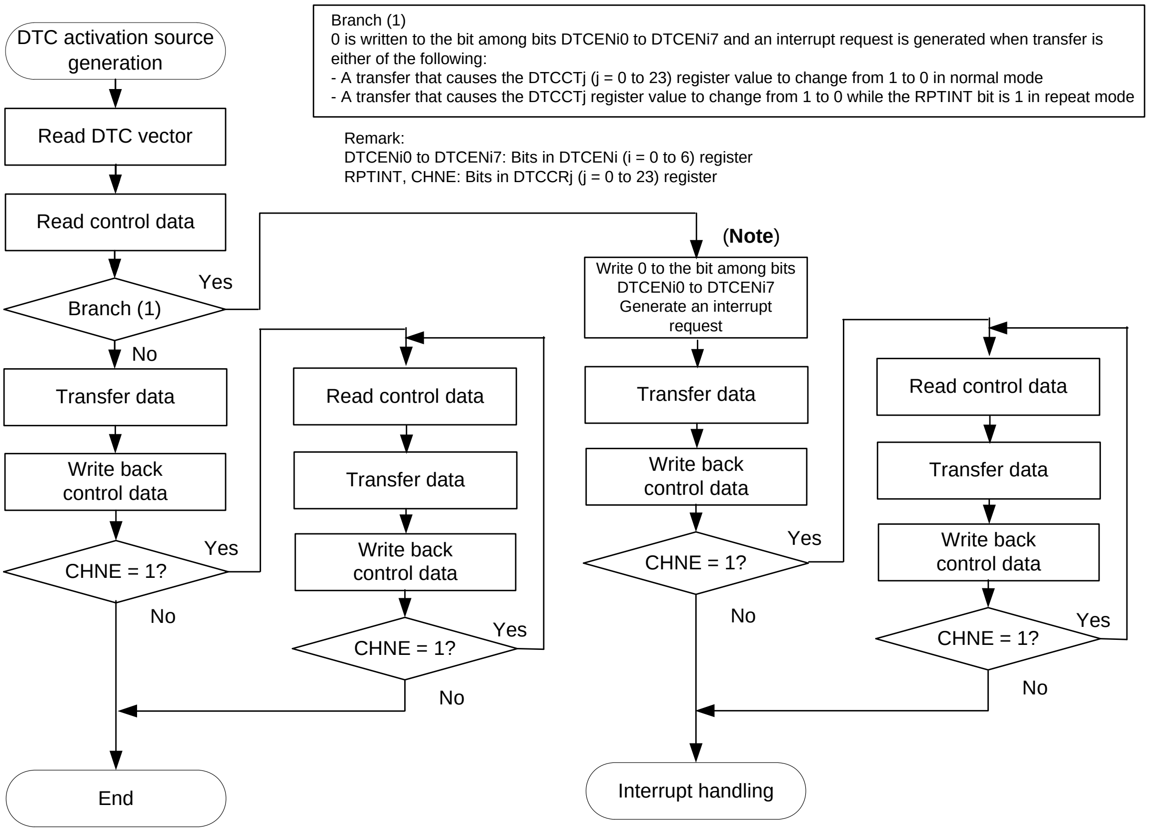
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 6) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 25-14 shows the DTC internal operation flowchart.

Figure 25-14. DTC Internal Operation Flowchart



Branch (1)
0 is written to the bit among bits DTCENi0 to DTCENi7 and an interrupt request is generated when transfer is either of the following:
- A transfer that causes the DTCCTj (j = 0 to 23) register value to change from 1 to 0 in normal mode
- A transfer that causes the DTCCTj register value to change from 1 to 0 while the RPTINT bit is 1 in repeat mode

Remark:
DTCENi0 to DTCENi7: Bits in DTCENi (i = 0 to 6) register
RPTINT, CHNE: Bits in DTCCRj (j = 0 to 23) register

Note 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

25.4.2 Normal mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 6) in the DTCENi register to 0 (activation disabled).

Table 25-7 shows register functions in normal mode. **Figure 25-15** shows data transfers in normal mode.

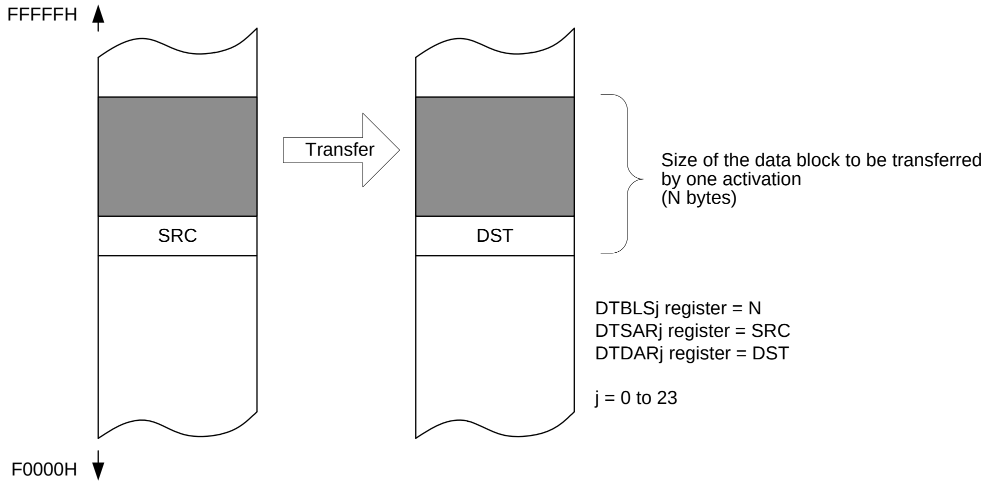
Table 25-7. Register Functions in Normal Mode

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 25-15. Data Transfers in Normal Mode


FFFFFH
F0000H
Transfer
SRC
DST
Size of the data block to be transferred by one activation (N bytes)

DTBLSj register = N
DTSARj register = SRC
DTDARj register = DST

j = 0 to 23

DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremented	Fixed	SRC + N	DST
1	0	X	0	Fixed	Incremented	SRC	DST + N
1	1	X	0	Incremented	Incremented	SRC + N	DST + N

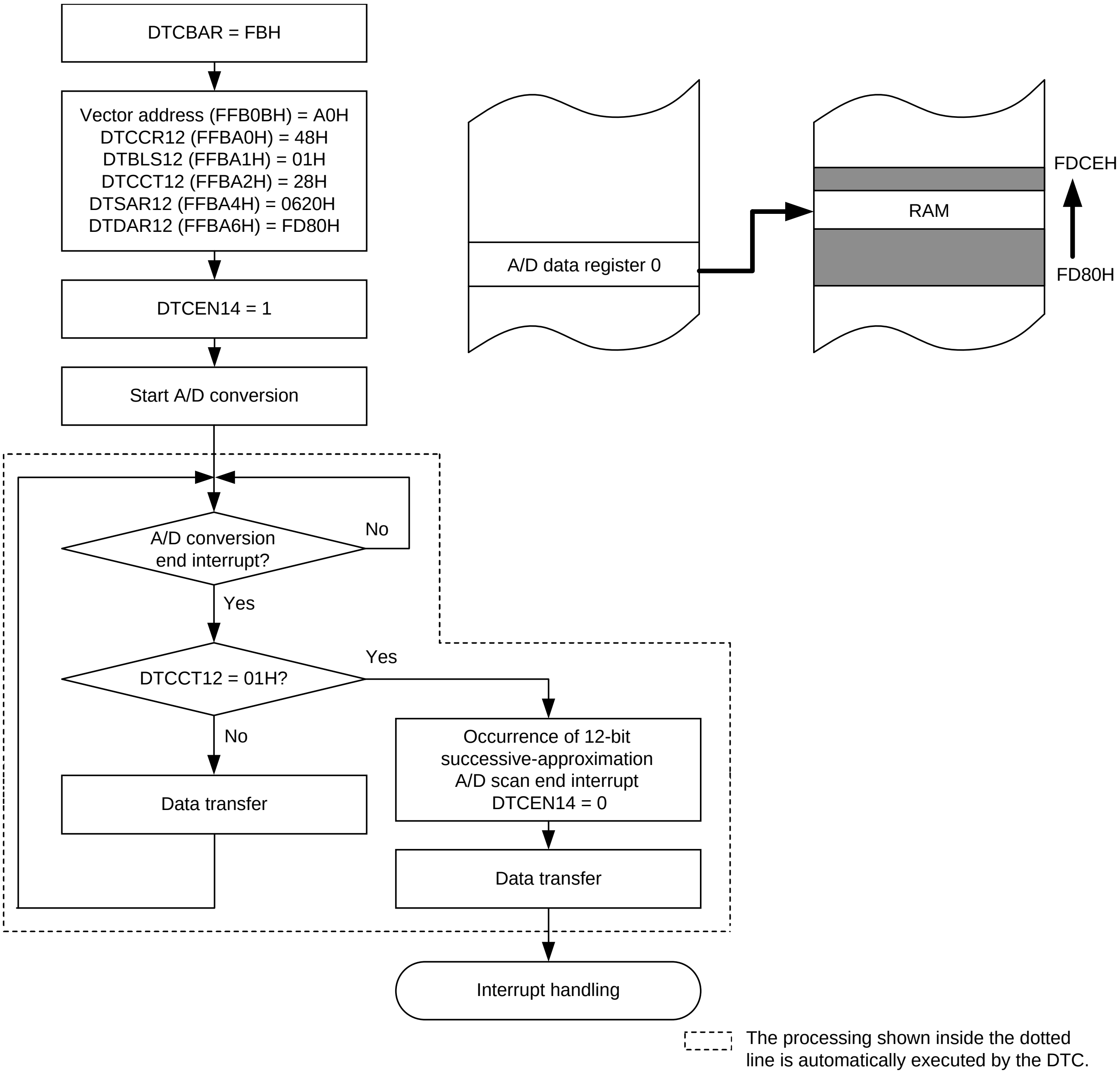
X: 0 or 1

(1) Example 1 of using normal mode: Capturing consecutive 12-bit results of A/D conversion

The DTC is activated by a 12-bit successive-approximation A/D scan end interrupt and transfers the value in A/D data register 0 to RAM.

- The vector address is FFB0BH and control data are allocated to FFBA0H to FFBA7H (control data 12).
- Data are transferred from the two halves of A/D data register 0 (at F0620H and F0621H) in two-byte units 40 times to fill the 80 bytes of RAM from FFD80H to FFDCFH.

Figure 25-16. Example 1 of Using Normal Mode: Capturing Consecutive 12-bit Results of A/D Conversion



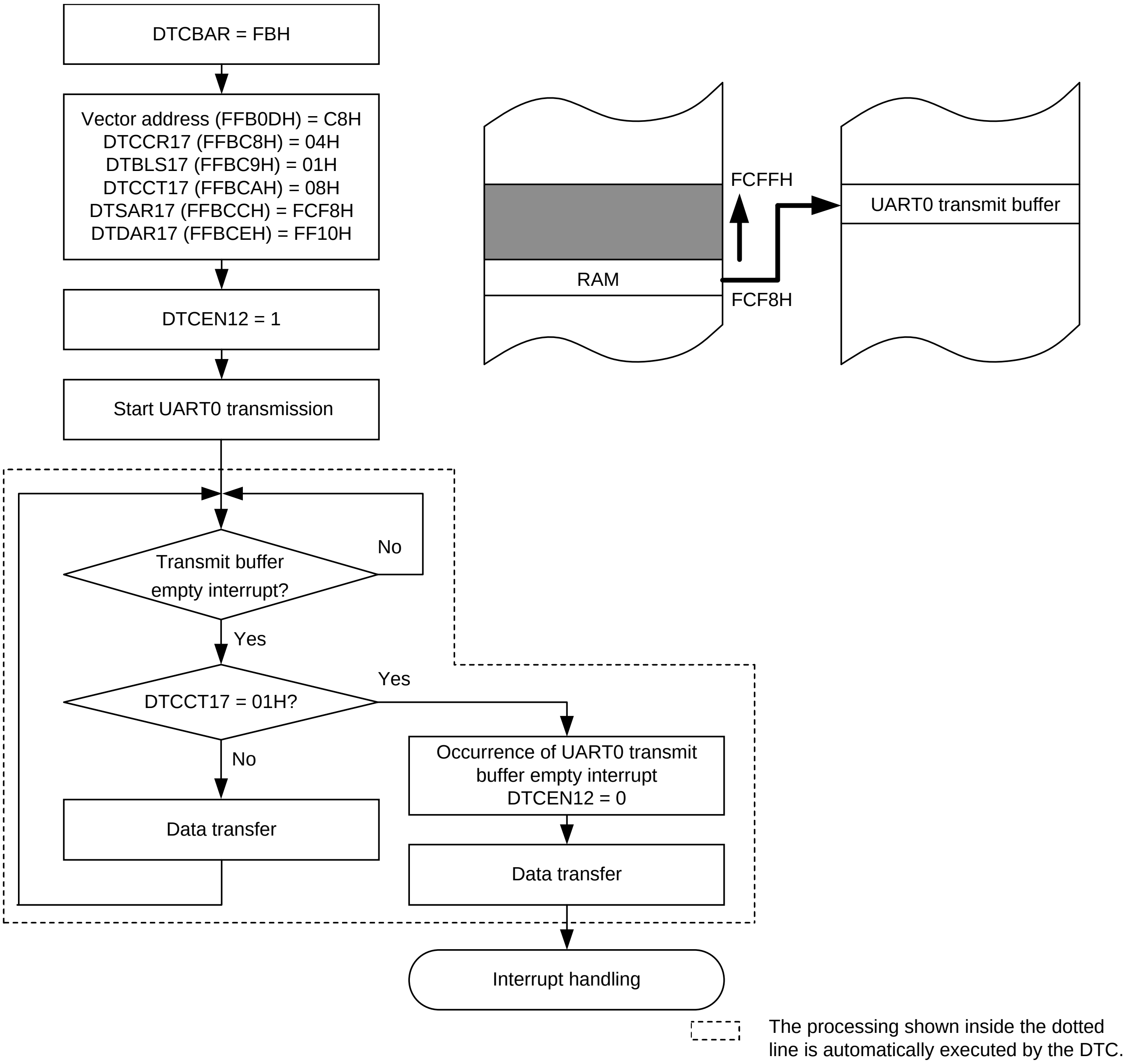
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(2) Example 2 of using normal mode: UART0 consecutive transmission

The DTC is activated by a UART0 transmit buffer empty interrupt and transfers the value of RAM to the UART0 transmit buffer.

- The vector address is FFB0DH and control data are allocated to FFBC8H to FFBCFH (control data 17).
- Transfers 8 bytes of FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H).

Figure 25-17. Example 2 of Using Normal Mode: UART0 Consecutive Transmission



The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.
Start the first UART0 transmission by software. The second and subsequent transmissions are automatically sent when the DTC is activated by a transmit buffer empty interrupt.

25.4.3 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 (j = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 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 DTCENi0 to DTCENi7 are not set to 0.

Table 25-8 lists register functions in repeat mode. **Figure 25-18** shows data transfers in repeat mode.

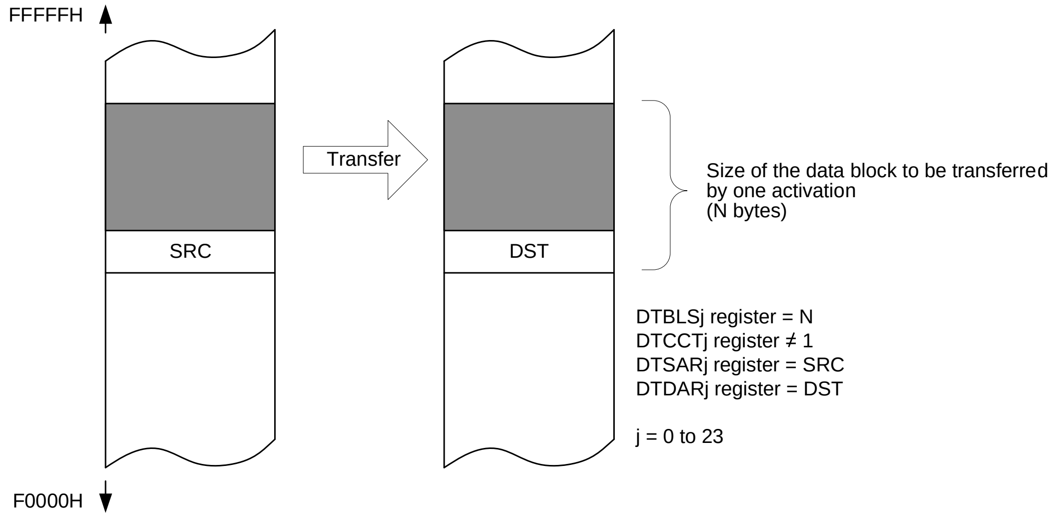
Table 25-8. Register Functions in Repeat Mode

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 25-18. Data Transfers in Repeat Mode

DTCCTj register ≠ 1



Size of the data block to be transferred
by one activation
(N bytes)

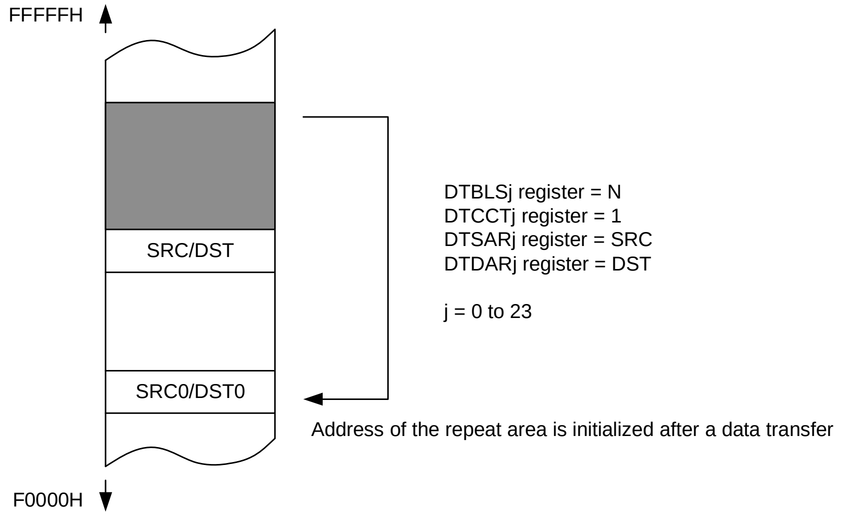
DTBLSj register = N
DTCCTj register ≠ 1
DTSARj register = SRC
DTDARj register = DST

j = 0 to 23

DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address After Transfer	Destination Address After Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC + N	DST
1	X	1	1	Repeat area	Incremented	SRC + N	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST + N
X	1	0	1	Incremented	Repeat area	SRC + N	DST + N

X: 0 or 1

DTCCTj register = 1



DTBLSj register = N
DTCCTj register = 1
DTSARj register = SRC
DTDARj register = DST

j = 0 to 23

Address of the repeat area is initialized after a data transfer

DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address After Transfer	Destination Address After Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC0	DST
1	X	1	1	Repeat area	Incremented	SRC0	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST0
X	1	0	1	Incremented	Repeat area	SRC + N	DST0

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

Cautions 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.

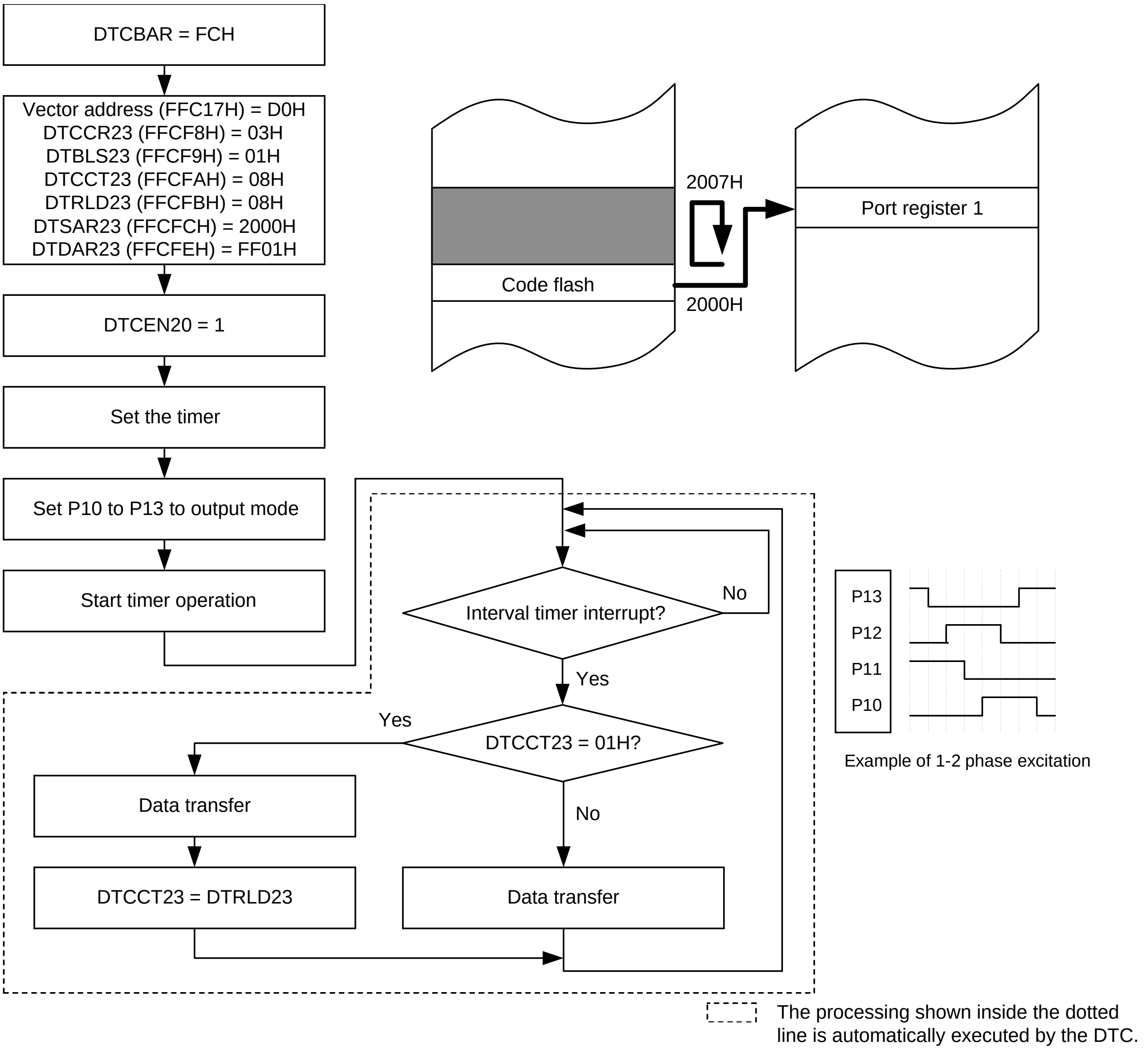
2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(1) Example of using repeat mode: Outputting a stepping motor control pulse using ports

The DTC is activated by an interval timer interrupt and transfers the pattern of the motor control pulse stored in the code flash memory to general-purpose ports.

- The vector address is FFC17H and control data are allocated to FFCF8H to FFCFFH (control data 23).
- Transfers 8-byte data of 02000H to 02007H of the code flash memory from the mirror space (F2000H to F2007H) to port register 1 (FFF01H).
- A repeat mode interrupt is disabled.

Figure 25-19. Example 1 of Using Repeat Mode: Outputting a Stepping Motor Control Pulse Using Ports



To stop the output, stop the timer first and then clear DTCEN20.

25.4.4 Chain transfers

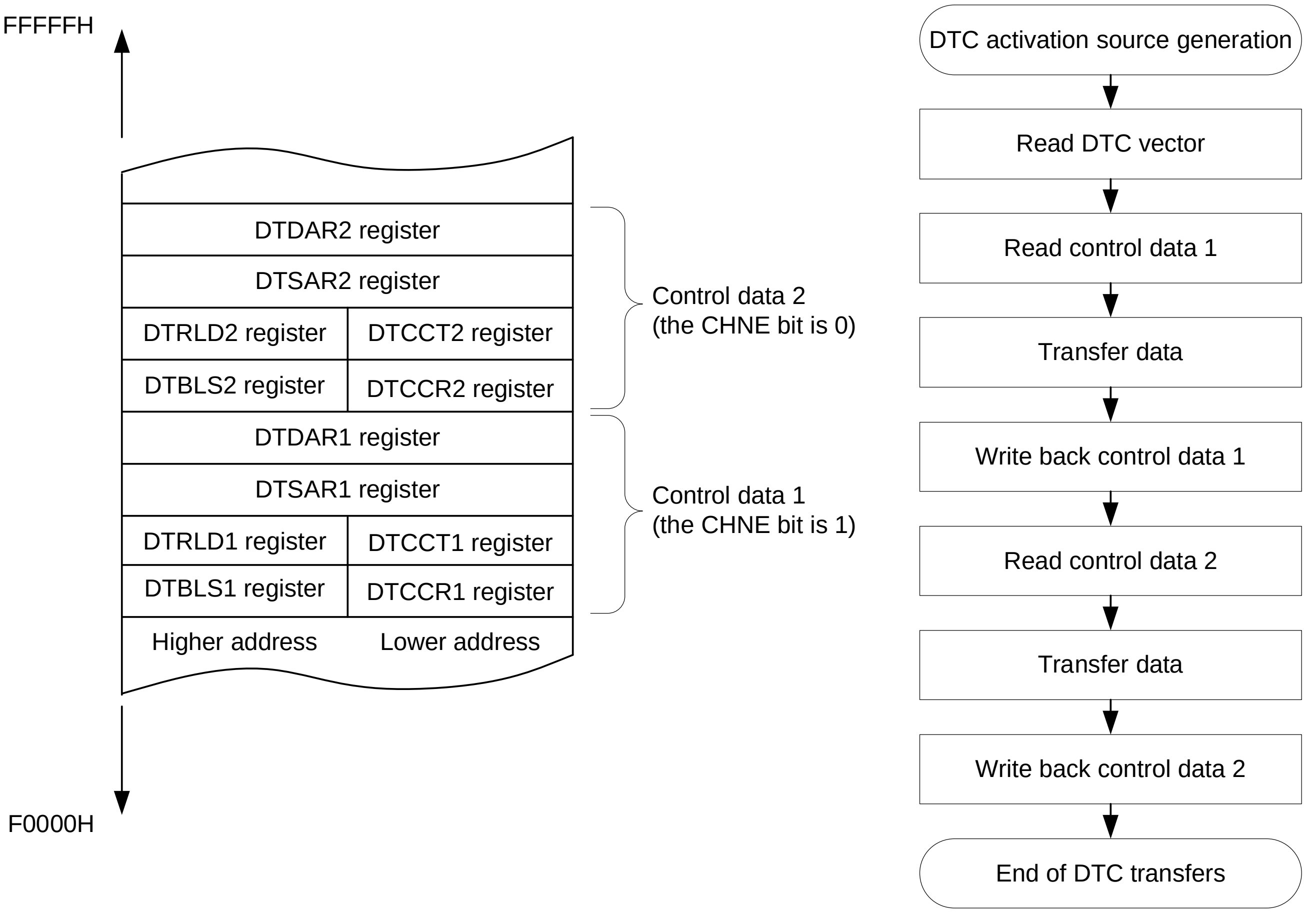
When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

When the DTC is activated, one control data is selected according to the data read from the DTC 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 and transferred 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 25-20 shows data transfers during chain transfers.

Figure 25-20. Data Transfers During Chain Transfers



Cautions 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

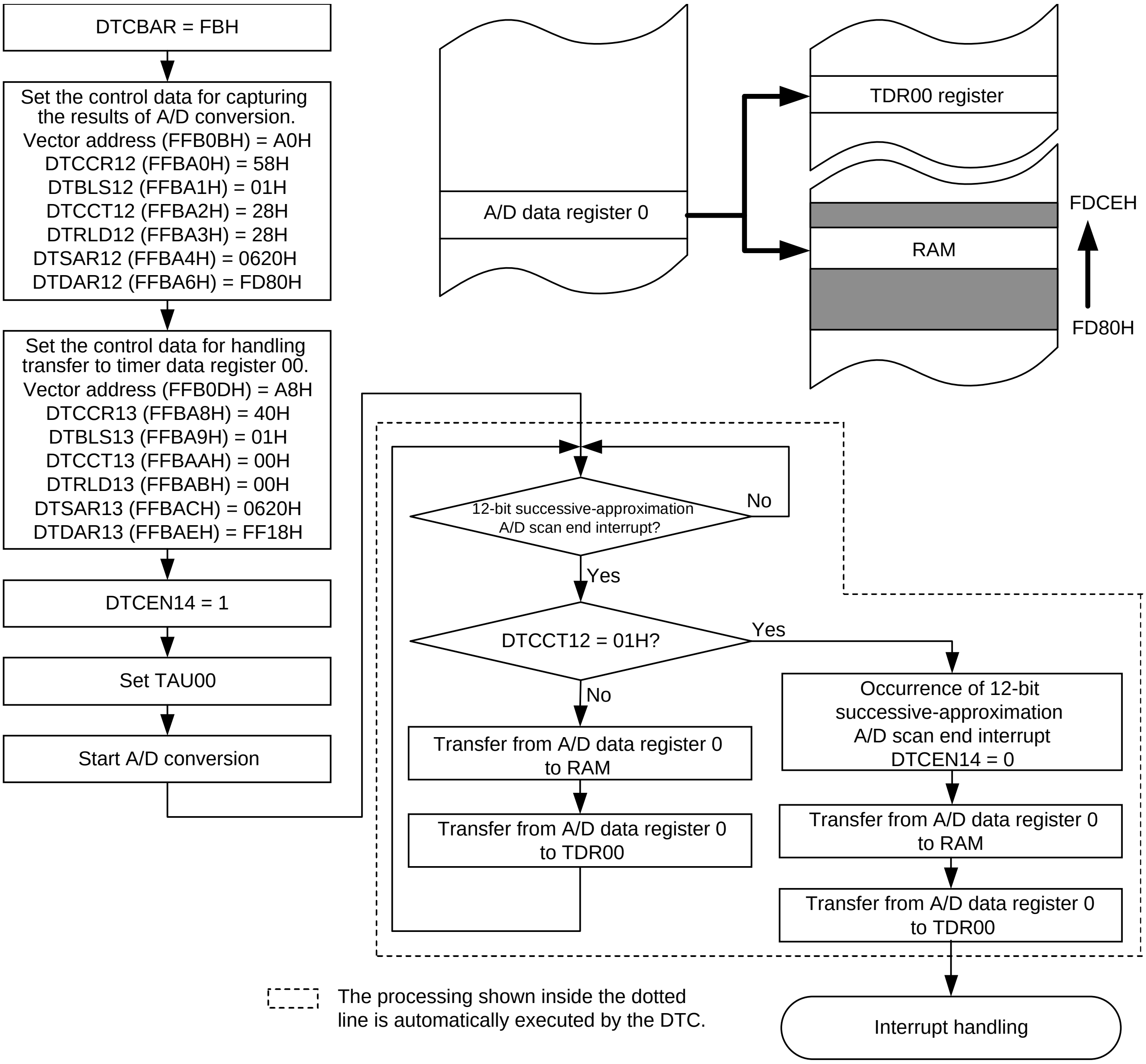
2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 6) in the DTCENi register are not set to 0 (DTC activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

(1) Example of using chain transfer: Capturing consecutive 12-bit results of A/D conversion and transferring them to timer data register 00 (TDR00)

The DTC is activated by a 12-bit successive-approximation A/D scan end interrupt and transfers the value in A/D data register 0 to RAM and then to timer data register 00 (TDR00).

- The vector address is FFB0BH.
- Control data for capturing the results of A/D conversion are allocated to FFBA0H to FFBA7H (control data 12).
- Control data for handling transfer to TDR00 are allocated to FFBA8H to FFBAFH.
- Data are transferred from the two halves of A/D data register 0 (at F0620H and F0621H) in two-byte units to the range from FFD80H to FFDCFH in RAM and then to TDR00 (at FFF18H and FFF19H).

Figure 25-21. Example of Using Chain Transfer: Capturing Consecutive Results of A/D Conversion and Transferring Them to Timer Data Register 00 (TDR00)


DTCBAR = FBH
Set the control data for capturing the results of A/D conversion.
Vector address (FFB0BH) = A0H
DTCCR12 (FFBA0H) = 58H
DTBLS12 (FFBA1H) = 01H
DTCCT12 (FFBA2H) = 28H
DTRLD12 (FFBA3H) = 28H
DTSAR12 (FFBA4H) = 0620H
DTDAR12 (FFBA6H) = FD80H
Set the control data for handling transfer to timer data register 00.
Vector address (FFB0DH) = A8H
DTCCR13 (FFBA8H) = 40H
DTBLS13 (FFBA9H) = 01H
DTCCT13 (FFBAAH) = 00H
DTRLD13 (FFBABH) = 00H
DTSAR13 (FFBACH) = 0620H
DTDAR13 (FFBAEH) = FF18H
DTCEN14 = 1
Set TAU00
Start A/D conversion
A/D data register 0
TDR00 register
RAM
FDCEH
FD80H
12-bit successive-approximation A/D scan end interrupt?
No
Yes
DTCCT12 = 01H?
Yes
No
Transfer from A/D data register 0 to RAM
Transfer from A/D data register 0 to TDR00
Occurrence of 12-bit successive-approximation A/D scan end interrupt DTCEN14 = 0
Transfer from A/D data register 0 to RAM
Transfer from A/D data register 0 to TDR00
The processing shown inside the dotted line is automatically executed by the DTC.
Interrupt handling

25.5 Notes on DTC

25.5.1 Setting DTC control data and vector table

- Do not access the DTC SFRs, the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 6) register is 0 (DTC activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 6) register is 0 (DTC activation disabled).
- Do not allocate RAM addresses which are used as a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

25.5.2 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.

- 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.
- The following RAM area cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.

  F7F00H to F8309H
- The following RAM area cannot be used as the DTC control data area or DTC vector table area when using the on-chip trace function.

  F8300H to F86FFH
- 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.

25.5.3 DTC pending instruction

Even if a DTC transfer request is generated, data 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
- Instruction of Multiply, Divide, Multiply & Accumulate (excluding MULU)

Cautions 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

25.5.4 Operation when accessing data flash memory space

When accessing the data flash space after an instruction execution from the start of DTC data transfer, a wait of three clock cycles will be inserted to the next instruction.

Instruction 1
DTC data transfer
Instruction ← The wait of three clock cycles occurs.
MOV A, ! Data Flash space

25.5.5 Number of DTC execution clock cycles

Table 25-9 lists the operations following DTC activation and required number of clock cycles for each operation.

Table 25-9. 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**

Notes 1. For the number of clock cycles required for control data write-back, see **Table 25-10 Number of Clock Cycles Required for Control Data Write-Back Operation**.

2. For the number of clock cycles required for data read/write, see **Table 25-11 Number of Clock Cycles Required for Data Read/Write Operation**.

Table 25-10. 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 25-11. Number of Clock Cycles Required for Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	SFR	2nd SFR	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait states[Note]
Data write	1	-	-	1	1	1 + number of wait states[Note]

Note The number of wait states differs depending on the specifications of the register allocated to the second SFR to be accessed.

25.5.6 DTC response time

Table 25-12 lists the DTC response time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts, excluding the number of DTC execution clocks.

Table 25-12. DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 clocks	19 clocks

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 conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clocks
- When executing a DTC pending instruction (see **25.5.3 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 that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

25.5.7 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, see **25.3.3 Vector table**.

25.5.8 Operation in standby mode status

Status	DTC Operation
HALT mode	Operable (Operation is disabled while in the low power consumption RTC mode)
STOP mode	DTC activation sources can be accepted^{Note 2}
SNOOZE mode	Operable^{Notes 1, 3, 4}

Notes 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or the middle-speed on-chip oscillator clock (f_{IM}) is selected as f_{CLK}.

2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the HALT or SNOOZE mode, the flash memory cannot be set as the transfer source.

3. When a transfer end interrupt is set as a DTC activation source from the CSIp SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set CSIp reception again (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm0 bit).

4. When a transfer end interrupt is set as a DTC activation source from the UARTq SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set UARTq reception again (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm1 bit).

Caution The SNOOZE function for the DTC and the SNOOZE function for UART cannot be used at the same time.

Remark p = 00; q = 0; m = 0

CHAPTER 26 EVENT LINK CONTROLLER (ELC)

26.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

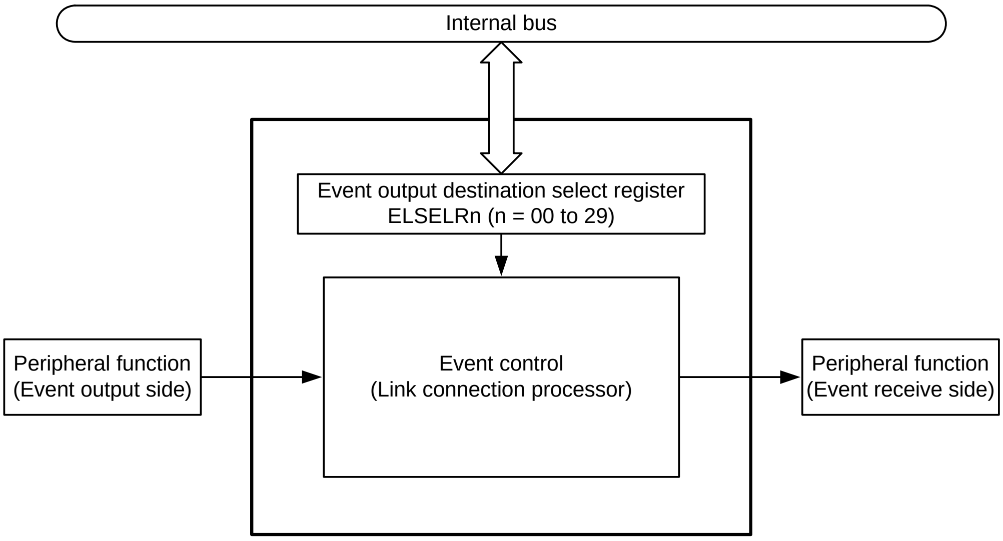
The ELC has the following functions.

- Capable of directly linking event signals from 30 types of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of 7 types of peripheral functions

26.2 Configuration of ELC

Figure 26-1 shows the ELC block diagram.

Figure 26-1. ELC Block Diagram



26.3 Registers Controlling ELC

Table 26-1 lists the registers controlling ELC.

Table 26-1. Registers Controlling ELC

Register Name	Symbol
Event output destination select register 00	ELSELR00
Event output destination select register 01	ELSELR01
Event output destination select register 02	ELSELR02
Event output destination select register 03	ELSELR03
Event output destination select register 04	ELSELR04
Event output destination select register 05	ELSELR05
Event output destination select register 06	ELSELR06
Event output destination select register 07	ELSELR07
Event output destination select register 08	ELSELR08
Event output destination select register 09	ELSELR09
Event output destination select register 10	ELSELR10
Event output destination select register 11	ELSELR11
Event output destination select register 12	ELSELR12
Event output destination select register 13	ELSELR13
Event output destination select register 14	ELSELR14
Event output destination select register 15	ELSELR15
Event output destination select register 16	ELSELR16
Event output destination select register 17	ELSELR17
Event output destination select register 18	ELSELR18
Event output destination select register 19	ELSELR19
Event output destination select register 20	ELSELR20
Event output destination select register 21	ELSELR21
Event output destination select register 22	ELSELR22
Event output destination select register 23	ELSELR23
Event output destination select register 24	ELSELR24
Event output destination select register 25	ELSELR25
Event output destination select register 26	ELSELR26
Event output destination select register 27	ELSELR27
Event output destination select register 28	ELSELR28
Event output destination select register 29	ELSELR29
Timer input select register 0	TIS0

26.3.1 Event output destination select register n (ELSELRn) (n = 00 to 29)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side).[Note] The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function. Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 26-2 lists the correspondence between ELSELRn (n = 00 to 29) registers and peripheral functions, and **Table 26-3** lists the correspondence between values set to ELSELRn (n = 00 to 29) registers and operation of link destination peripheral functions at reception.

Note This restriction does not apply to the 12-bit A/D converter in that it can serve as the destination for multiple event inputs.

Figure 26-2. Format of Event Output Destination Select Register n (ELSELRn)

Address: F0240H (ELSELR00) to F025DH (ELSELR29) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	Event link selection
0	0	0	Event link disabled
0	0	1	Select operation of peripheral function 1 to link[Note]
0	1	0	Select operation of peripheral function 2 to link[Note]
0	1	1	Select operation of peripheral function 3 to link[Note]
1	0	0	Select operation of peripheral function 4 to link[Note]
1	0	1	Select operation of peripheral function 5 to link[Note]
1	1	0	Select operation of peripheral function 6 to link[Note]
1	1	1	Select operation of peripheral function 7 to link[Note]

Note See **Table 26-3 Correspondence Between Values Set to ELSELRn (n = 00 to 29) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 26-2. Correspondence Between ELSELRn (n = 00 to 29) Registers and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01	External interrupt edge detection 1	INTP1
ELSELR02	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	External interrupt edge detection 4	INTP4
ELSELR05	External interrupt edge detection 5	INTP5
ELSELR06	External interrupt edge detection 6	INTP6
ELSELR07	External interrupt edge detection 7	INTP7
ELSELR08	Key return signal detection	INTKR
ELSELR09	12-bit interval timer interval signal detection	INTIT
ELSELR10	8-bit interval timer channel 00 compare match or 8-bit interval timer channel 0 compare match (cascaded)	INTIT00
ELSELR11	8-bit interval timer channel 10 compare match or 8-bit interval timer channel 1 compare match (cascaded)	INTIT10
ELSELR12	Fixed-cycle signal of real-time clock	INTRTCPRD
ELSELR13	TAU channel 00 count end/capture end	INTTM00
ELSELR14	TAU channel 01 count end/capture end	INTTM01
ELSELR15	TAU channel 02 count end/capture end	INTTM02
ELSELR16	TAU channel 03 count end/capture end	INTTM03
ELSELR17	TAU channel 05 count end/capture end	INTTM05
ELSELR18	TAU channel 07 count end/capture end	INTTM07
ELSELR19	24-bit ΔΣ-type A/D conversion end	INTDSAD
ELSELR20	24-bit ΔΣ-type A/D conversion end	INTDSAD
ELSELR21	External interrupt edge detection 8	INTP8
ELSELR22	External interrupt edge detection 9	INTP9
ELSELR23	External interrupt edge detection 12	INTP12
ELSELR24	External interrupt edge detection 13	INTP13
ELSELR25	External interrupt edge detection 14	INTP14
ELSELR26	8-bit interval timer channel 20 compare match or 8-bit interval timer channel 2 compare match (cascaded)	INTIT20
ELSELR27	8-bit interval timer channel 30 compare match or 8-bit interval timer channel 3 compare match (cascaded)	INTIT30
ELSELR28	Timer RJ0 underflow	INTTRJ0
ELSELR29	Timer RJ1 underflow	INTTRJ1

Table 26-3. Correspondence Between Values Set to ELSELRn (n = 00 to 29) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELRn2 to ELSELRn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
001B	1	12-bit A/D converter	A/D conversion starts
010B	2	Timer input of timer array unit channel 0[Note 1]	Delay counter, input pulse interval measurement, external event counter
011B	3	Timer input of timer array unit channel 1[Note 2]	Delay counter, input pulse interval measurement, external event counter
100B	4	Timer input of timer array unit channel 5[Note 3]	Delay counter, input pulse interval measurement, external event counter
101B	5	Timer input of timer array unit channel 7[Note 4]	Delay counter, input pulse interval measurement, external event counter
110B	6	Timer RJ0	Count source
111B	7	Timer RJ1	Count source

Notes 1. To select the timer input of timer array unit channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

2. To select the timer input of timer array unit channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

3. To select the timer input of timer array unit channel 5 as the link destination peripheral function, set the operating clock for channel 5 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI05 pin to OFF (TNFEN05 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 5 to an event input signal from the ELC using timer input select register 0 (TIS0).

4. To select the timer input of timer array unit channel 7 as the link destination peripheral function, set the operating clock for channel 7 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI07 pin to OFF (TNFEN07 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 7 to an event input signal from the ELC using timer input select register 0 (TIS0).

26.3.2 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input for channels 0, 1, 5, 6, and 7 of the timer array unit (TAU0).

Figure 26-3. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TIS07	TIS06	Selection of timer input used with channel 7
0	0	Input signal of timer input pin (TI07)
0	1	RTCOUT output signal
1	0	RxD0 input pin
1	1	Event input signal from ELC

TIS04	Selection of timer input used with channel 1
0	Input signal of timer input pin (TI01)
1	Event input signal from ELC

TIS03	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI05)
0	1	1	Middle-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 above			Setting prohibited

26.3.3 A/D conversion start trigger select register (ADSTRGR)

The ADSTRGR register selects the A/D conversion start trigger.

This register can be set by 16-bit memory manipulation instructions.

Figure 26-4. Format of A/D Conversion Start Trigger Select Register (ADSTRGR)

Address: F0610h After reset: 0000h R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADSTRGR	0	0	TRSA[5:0]						0	0	0	0	0	0	0	0

TRSA[5:0]	A/D conversion start trigger select bit [Note]
110000	Event output signal from event link controller (ELCTRG0)
111111	Trigger source deselection
Other than above	Setting prohibited
The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. ● When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1. ● Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit and the TRSA[5:0] bits.	

Note The issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. See **17.3.4 Analog input sampling time and scan conversion time** for details.

Table 26-4 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 26-4. Selection of A/D Activation Sources by the TRSA[5:0] Bits

Peripheral Function	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			1	1	1	1	1	1
ELC	ELCTRG0	Event output signal from the event link controller	1	1	0	0	0	0

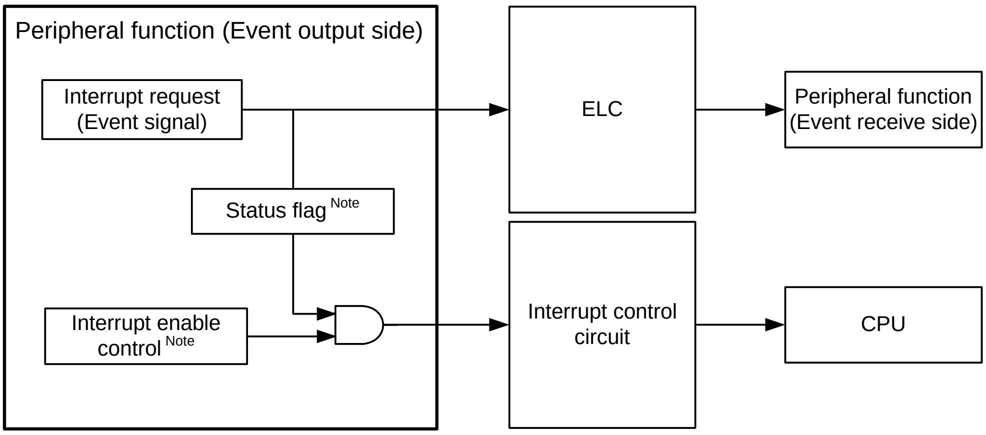
26.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 26-5 shows the relationship between interrupt handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (See **Table 26-3 Correspondence Between Values Set to ELSELRn (n = 00 to 29) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 26-5. Relationship Between Interrupt Handling and ELC


Peripheral function (Event output side)
Interrupt request (Event signal)
Status flag Note
Interrupt enable control Note
ELC
Interrupt control circuit
Peripheral function (Event receive side)
CPU

Note Not available depending on the peripheral function.

Table 26-5 lists the response of peripheral functions that receive events.

Table 26-5. Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	12-bit A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
3	Timer array unit Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
4	Timer array unit Timer input of channel 5	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
5	Timer array unit Timer input of channel 7	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
6	Timer RJ0	Count source	An event from the ELC is directly used as a count source of timer RJ0.
7	Timer RJ1	Count source	An event from the ELC is directly used as a count source of timer RJ1.

26.5 Points for Caution when Using the ELC

- Attempting to link multiple event inputs to a single event trigger is prohibited.[Note]
- Ensure that conditions for the generation of an event signal are not satisfied in a related peripheral module while setting control registers of the ELC.
- Setting the same module as the source of an event signal and destination for linkage is prohibited.

Note This restriction does not apply to the 12-bit A/D converter in that it can serve as the destination for multiple event inputs.

CHAPTER 27 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.

		80-pin	100-pin
Maskable interrupts	External	14	14
	Internal	41	47

27.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. Default priority, see **Table 27-1**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupts

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

27.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 27-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 27-1. 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
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time+1/2f_{IL})	Internal	0004H	(A)	√	√
	1	INTLVI	Voltage detection[Note 4]		0006H		√	√
	2	INTP0	Pin input edge detection	External	0008H	(B)	√	√
	3	INTP1			000AH		√	√
	4	INTP2			000CH		√	√
	5	INTP3			000EH		√	√
	6	INTP4			0010H		√	√
	7	INTP5			0012H		√	√
	8	INTST2	UART2 transmission transfer end or buffer empty interrupt	Internal	0014H	(A)	√	√
		INTSTMG0	UARTMG0 transmission transfer end or buffer empty interrupt				√	√
	9	INTSR2	UART2 reception transfer end		0016H		√	√
		INTSRMG0	UARTMG0 reception transfer end				√	√
	10	INTSRE2	UART2 reception communication error occurrence		0018H		√	√
		INTSREMG0	UARTMG0 reception transfer end communication error occurrence				√	√
	11	INTCR	End of high-speed on-chip oscillator clock frequency correction		001AH		√	√
		INTSMP00	Sampling detector detection 00				√	√
		INTSMP10	Sampling detector detection 10				√	√
	12	INTAES	AES encryption/decryption end		001CH		√	√
		INTAESF	AES encryption/decryption end of first block				√	√
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		001EH		√	√
	14	INTIICA0	End of IICA0 communication		0020H		√	√
	15	INTSR0	UART0 reception transfer end		0022H		√	√
	16	INTSRE0	UART0 reception communication error occurrence		0024H		√	√
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				√	√
	17	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		0026H		√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 27-1**.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 27-1. 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
		Name	Trigger					
Maskable	18	INTSR1	UART1 reception transfer end	Internal	0028H	(A)	√	√
	19	INTSRE1	UART1 reception communication error occurrence		002AH		√	√
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				√	√
	20	INTTM00	End of timer channel 00 count or capture		002CH		√	√
	21	INTRTCALM0	RTC alarm interrupt 0		002EH		√	√
		INTSMP01	Sampling detector detection 01				√	√
		INTSMP11	Sampling detector detection 11				√	√
	22	INTFM	End of frequency measurement		0030H		√	√
		INTSMP02	Sampling detector detection 02				√	√
		INTSMP12	Sampling detector detection 12				√	√
	23	INTTM01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)		0032H		√	√
	24	INTTM02	End of timer channel 02 count or capture		0034H		√	√
	25	INTTM03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)		0036H		√	√
	26	INTAD	12-bit successive-approximation A/D scan end interrupt		0038H		√	√
	27	INTRTCPRD	Fixed-cycle signal of realtime clock		003AH		√	√
		INTSMP03	Sampling detector detection 03				√	√
		INTSMP13	Sampling detector detection 13				√	√
	28	INTIT	Interval signal of 12-bit interval timer detection		003CH		√	√
	29	INTKR	Key return signal detection	External	003EH	(B)	√	√
		INTRTCALM1	RTC alarm interrupt 1	Internal		(A)	√	√
		INTSMP04	Sampling detector detection 04				√	√
		INTSMP14	Sampling detector detection 14				√	√
	30	INTST3/ INTCSI30/ INTIIC30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt/IIC30 transfer end		0040H		√	-
	31	INTSR3	UART3 reception transfer end		0042H		√	-
	32	INTDSAD	End of ΔΣ A/D conversion		0044H		√	√
		INTSMP05	Sampling detector detection 05				√	√
		INTSMP15	Sampling detector detection 15				√	√
	33	INTTM04	End of timer channel 04 count or capture		0046H		√	√
	34	INTTM05	End of timer channel 05 count or capture		0048H		√	√
		INTSMOTA0	Sampling output timer interval interrupt 0				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 27-1**.

Table 27-1. 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
Maskable		Name	Trigger					
Maskable	35	INTP6	Pin input edge detection	External	004AH	(B)	√	√
	36	INTP7	Pin input edge detection		004CH		√	√
	37	INTRTCIC2	Tamper detection of RTCIC2 pin		004EH		√	√
		INTP12	Pin input edge detection				√	√
	38	INTRTCIC1	Tamper detection of RTCIC1 pin		0050H		√	√
		INTP13	Pin input edge detection				√	√
	39	INTRTCIC0	Tamper detection of RTCIC0 pin		0052H		√	√
		INTP14	Pin input edge detection				√	√
	40	INTTM06	End of timer channel 06 count or capture	Internal	0054H	(A)	√	√
		INTSMOTB0	Sampling output timer compare match interrupt 0				√	√
	41	INTTM07	End of timer channel 07 count or capture		0056H		√	√
		INTSMOTA1	Sampling output timer interval interrupt 1				√	√
	42	INTIT00	8-bit interval timer channel 00/channel 0 (when cascade) compare match detection		0058H		√	√
	43	INTIT01	8-bit interval timer channel 01 compare match detection		005AH		√	√
	44	INTSRE3	UART3 reception communication error occurrence		005CH		√	-
	45	INTMACLOF	Multiply-accumulation overflow/underflow interrupt		005EH		√	√
		INTSMOTB1	Sampling output timer compare match interrupt 1				√	√
	46	INTOSDC	Oscillation stop detection		0060H		√	√
	47	INTFL	Reserved [Note 3]		0062H		√	√
	48	INTP8	Pin input edge detection		0064H		√	√
	49	INTP9	Pin input edge detection		0066H		√	√
	50	INTIT10	8-bit interval timer channel 10/channel 1 (when cascade) compare match detection		0068H		√	√
	51	INTIT11	8-bit interval timer channel 11 compare match detection		006AH		√	√
	52	INTLVDVDD	Voltage detection of V_{DD} pin		006CH		√	√
		INTIT20	8-bit interval timer channel 20/channel 2 (when cascade) compare match detection				√	√
	53	INTLVDVBAT	Voltage detection of VBAT pin		006EH		√	√
		INTIT21	8-bit interval timer channel 21 compare match detection				√	√
	54	INTLVDVRTC	Voltage detection of VRTC pin		0070H		√	√
		INTIT30	8-bit interval timer channel 30/channel 3 (when cascade) compare match detection				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 27-1**.

3. Be used at the flash self-programming library or the data flash library.

Table 27-1. 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
		Name	Trigger					
Maskable	55	INTLVDEXLVD	Voltage detection of EXLVD pin	Internal	0072H	(A)	√	√
		INTIT31	8-bit interval timer channel 31 compare match detection				√	√
	56	INTST4	UART4 transmission transfer end or buffer empty interrupt		0074H		√	-
		INTSTMG1	UARTMG1 transmission transfer end or buffer empty interrupt				√	√
	57	INTSR4	UART4 reception transfer end		0076H		√	-
		INTSRMG1	UARTMG1 reception transfer end				√	√
	58	INTSRE4	UART4 reception communication error occurrence		0078H		√	-
		INTSREMG1	UARTMG1 reception transfer end communication error occurrence				√	√
	59	INTTRJ0	Timer RJ0 interrupt		007AH		√	√
		INTDSADDEC	ΔΣ A/D decimation filter output end interrupt				√	√
	60	INTTRJ1	Timer RJ1 interrupt		007CH		√	√
Software	–	BRK	Execution of BRK instruction	–	007EH	(C)	√	√
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–	√	√
		POR	Power-on-reset				√	√
		LVD	Voltage detection[Note 3]				√	√
		WDT	Overflow of watchdog timer				√	√
		TRAP	Execution of illegal instruction[Note 4]				√	√
		IAW	Illegal-memory access				√	√
		RPE	RAM parity error				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

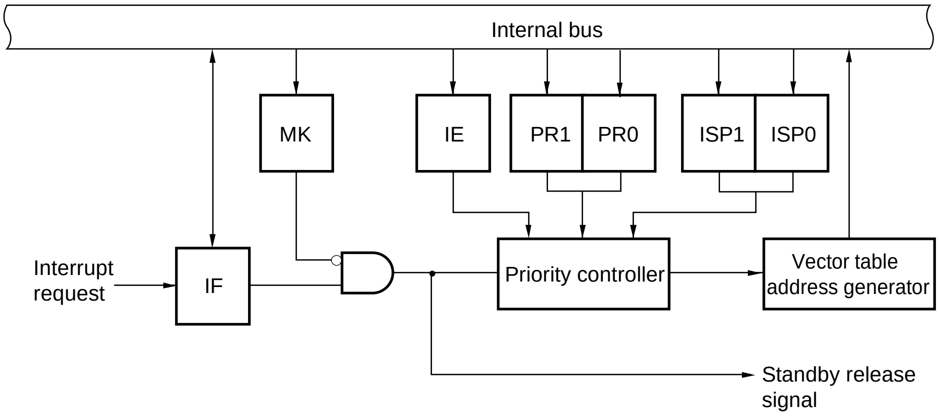
2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 27-1**.

3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

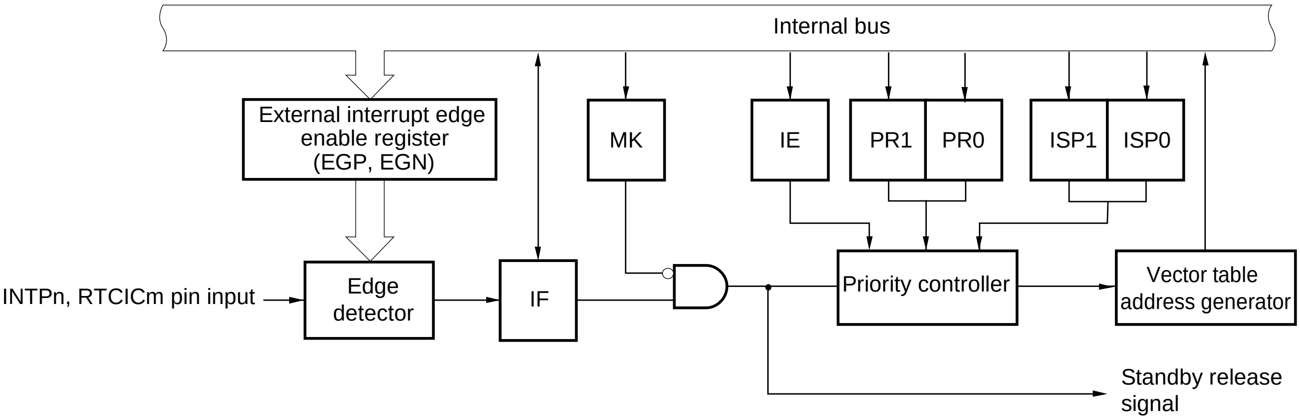
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 27-1. Basic Configuration of Interrupt Function

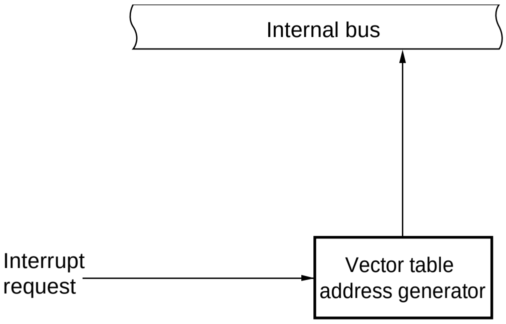
(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn, RTCICm)



(C) Software interrupt



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 n = 0 to 9, 12 to 14, m = 0 to 2

27.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- 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, EGP1)
- External interrupt falling edge enable register (EGN0, EGN1)
- Program status word (PSW)

Table 27-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 27-2. Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
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	
INTST2	STIF2	IF0H	STMK2	MK0H	STPR02, STPR12	PR00H, PR10H
INTSTMG0	STMGIF0		STMGMK0		STMGPR00, STMGPR10	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTSRMG0	SRMGIF0		SRMGMK0		SRMGPR00, SRMGPR10	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTSREMG0	SREMGIF0		SREMGMK0		SREMGPR00, SREMGPR10	
INTCR	CRIF		CRMK		CRPR0, CRPR1	
INTSMP00	SMPIF00		SMPMK00		SMPPR000, SMPPR100	
INTSMP10	SMPIF10		SMPMK10		SMPPR010, SMPPR110	
INTAES	AESIF		AESMK		AESPR0, AESPR1	
INTAESF	AESFIF		AESFMK		AESFPR0, AESFPR1	
INTST0	STIF0		STMK0		STPR00, STPR10	
INTCSI00	CSIIF00		CSIMK00		CSIPR000, CSIPR100	
INTIIC00	IICIF00		IICMK00		IICPR000, IICPR100	
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	

Table 27-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTSRE0	SREIF0	IF1L	SREMK0	MK1L	SREPR00, SREPR10	PR01L, PR11L
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H	
INTST1	STIF1		STMK1		STPR01, STPR11	
INTCSI10	CSIIF10		CSIMK10		CSIPR010, CSIPR110	
INTIIC10	IICIF10		IICMK10		IICPR010, IICPR110	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTRTCALM0	RTCAIF0		RTCAMK0		RTCAPR00, RTCAPR10	
INTSMP01	SMPIF01		SMPMK01		SMPPR001, SMPPR101	
INTSMP11	SMPIF11		SMPMK11		SMPPR011, SMPPR111	
INTFM	FMIF		FMMK		FMPR0, FMPR1	
INTSMP02	SMPIF02		SMPMK02		SMPPR002, SMPPR102	
INTSMP12	SMPIF12		SMPMK12		SMPPR012, SMPPR112	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02	IF1H	TMMK02	MK1H	TMPR002, TMPR102	PR01H, PR11H
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF		ADMK		ADPR0, ADPR1	
INTRTCPRD	RTCRIF		RTCRMK		RTCRPR0, RTCRPR1	
INTSMP03	SMPIF03		SMPMK03		SMPPR003, SMPPR103	
INTSMP13	SMPIF13		SMPMK13		SMPPR013, SMPPR113	
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTRTCALM1	RTCAIF1		RTCAMK1		RTCAPR01, RTCAPR11	
INTSMP04	SMPIF04		SMPMK04		SMPPR004, SMPPR104	
INTSMP14	SMPIF14		SMPMK14		SMPPR014, SMPPR114	
INTST3	STIF3		STMK3		STPR03, STPR13	
INTCSI30	CSIIF30		CSIMK30		CSIPR030, CSIPR130	
INTIIC30	IICIF30		IICMK30		IICPR030, IICPR130	
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTDSAD	DSAIF	IF2L	DSAMK	MK2L	DSAPR0, DSAPR1	PR02L, PR12L
INTSMP05	SMPIF05		SMPMK05		SMPPR005, SMPPR105	
INTSMP15	SMPIF15		SMPMK15		SMPPR015, SMPPR115	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05		TMMK05		TMPR005, TMPR105	
INTSMOTA0	SMOTAIF0		SMOTAMK0		SMOTAPR00, SMOTAPR10	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTRTCIC2	RTCIIF2		RTCIMK2		RTCIPR02, RTCIPR12	
INTP12	PIF12		PMK12		PPR012, PPR112	
INTRTCIC1	RTCIIF1		RTCIMK1		RTCIPR01, RTCIPR11	
INTP13	PIF13		PMK13		PPR013, PPR113	
INTRTCIC0	RTCIIF0		RTCIMK0		RTCIPR00, RTCIPR10	
INTP14	PIF14		PMK14		PPR014, PPR114	

Table 27-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTTM06	TMIF06	IF2H	TMMK06	MK2H	TMPR006, TMPR106	PR02H, PR12H
INTSMOTB0	SMOTBIF0		SMOTBMK0		SMOTBPR00, SMOTBPR10	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTSMOTA1	SMOTAIF1		SMOTAMK1		SMOTAPR01, SMOTAPR11	
INTIT00	ITIF00		ITMK00		ITPR000, ITPR100	
INTIT01	ITIF01		ITMK01		ITPR001, ITPR101	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTMACLOF	MACIF		MACMK		MACPR0, MACPR1	
INTSMOTB1	SMOTBIF1		SMOTBMK1		SMOTBPR01, SMOTBPR11	
INTOSDC	OSDIF		OSDMK		OSDPR0, OSDPR1	
INTFL	FLIF		FLMK		FLPR0, FLPR1	
INTP8	PIF8	IF3L	PMK8	MK3L	PPR08, PPR18	PR03L, PR13L
INTP9	PIF9		PMK9		PPR09, PPR19	
INTIT10	ITIF10		ITMK10		ITPR010, ITPR110	
INTIT11	ITIF11		ITMK11		ITPR011, ITPR111	
INTLVDVDD	LVDVDIF		LVDVDMK		LVDVDPR0, LVDVDPR1	
INTIT20	ITIF20		ITMK20		ITPR020, ITPR120	
INTLVDVBAT	LVDVBIF		LVDVBMK		LVDVBPR0, LVDVBPR1	
INTIT21	ITIF21		ITMK21		ITPR021, ITPR121	
INTLVDVRTC	LVDVRIF		LVDVRMK		LVDVRPR0, LVDVRPR1	
INTIT30	ITIF30		ITMK30		ITPR030, ITPR130	
INTLVDEXLVD	LVDEXIF		LVDEXMK		LVDEXPR0, LVDEXPR1	
INTIT31	ITIF31		ITMK31		ITPR031, ITPR131	
INTST4	STIF4	IF3H	STMK4	MK3H	STPR04, STPR14	PR03H, PR13H
INTSTMG1	STMGIF1		STMGMK1		STMGPR01, STMGPR11	
INTSR4	SRIF4		SRMK4		SRPR04, SRPR14	
INTSRMG1	SRMGIF1		SRMGMK1		SRMGPR01, SRMGPR11	
INTSRE4	SREIF4		SREMK4		SREPR04, SREPR14	
INTSREMG1	SREMGIF1		SREMGMK1		SREMGPR01, SREMGPR11	
INTTRJ0	TRJIF0		TRJMK0		TRJPR00, TRJPR10	
INTDSADDEC	DSADECIF		DSADECMK		DSADECPR0, DSADECPR1	
INTTRJ1	TRJIF1		TRJMK1		TRJPR01, TRJPR11	

27.3.1 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 an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

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, 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.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 27-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SRIF0	IICAIF0	STIF0 CSIIF00 IICIF00	AESIF AESFIF	CRIF SMPIF00 SMPIF10	SREIF2 SREMGIF0	SRIF2 SRMGIF0	STIF2 STMGIF0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF01	FMIF SMPIF02 SMPIF12	RTCAIF0 SMPIF01 SMPIF11	TMIF00	SREIF1 TMIF03H	SRIF1	STIF1 CSIIF10 IICIF10	SREIF0 TMIF01H

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	SRIF3	STIF3 CSIIF30 IICIF30	KRIF RTCAIF1 SMPIF04 SMPIF14	TMKAIF	RTCRIF SMPIF03 SMPIF13	ADIF	TMIF03	TMIF02

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	RTCIIF0 PIF14	RTCIIF1 PIF13	RTCIIF2 PIF12	PIF7	PIF6	TMIF05 SMOTAIF0	TMIF04	DSAIF SMPIF05 SMPIF15

Figure 27-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	FLIF	OSDIF	MACIF SMOTBIF1	SREIF3	ITIF01	ITIF00	TMIF07 SMOTAIF1	TMIF06 SMOTBIF0

Address: FFFD2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	LVDEXIF ITIF31	LVDVRIF ITIF30	LVDVBIF ITIF21	LVDVDIF ITIF20	ITIF11	ITIF10	PIF9	PIF8

Address: FFFD3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	TRJIF1	TRJIF0 DSADECIF	SREIF4 SREMGIF1	SRIF4 SRMGIF1	STIF4 STMGIF1

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. For details about the bits, see Table 27-2. Be sure to clear bits that are not available to 0.

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.

27.3.2 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, the MK2L and MK2H, 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.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 27-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (1/2)

Address: FFFE4H After reset: FFH 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

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SRMK0	IICAMK0	STMK0 CSIMK00 IICMK00	AESMK AESFMK	CRMK SMPMK00 SMPMK10	SREMK2 SREMGMK0	SRMK2 SRMGMK0	STMK2 STMGMK0

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK01	FMMK SMPMK02 SMPMK12	RTCAMK0 SMPMK01 SMPMK11	TMMK00	SREMK1 TMMK03H	SRMK1	STMK1 CSIMK10 IICMK10	SREMK0 TMMK01H

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	SRMK3	STMK3 CSIMK30 IICMK30	KRMK RTCAMK1 SMPMK04 SMPMK14	TMKAMK	RTCRMK SMPMK03 SMPMK13	ADMK	TMMK03	TMMK02

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	RTCIMK0 PMK14	RTCIMK1 PMK13	RTCIMK2 PMK12	PMK7	PMK6	TMMK05 SMOTAMK0	TMMK04	DSAMK SMPMK05 SMPMK15

Figure 27-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (2/2)

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	FLMK	OSDMK	MACMK SMOTBMK1	SREMK3	ITMK01	ITMK00	TMMK07 SMOTAMK1	TMMK06 SMOTBMK0

Address: FFFD6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	LVDEXMK ITMK31	LVDVRMK ITMK30	LVDVBMK ITMK21	LVDVDMK ITMK20	ITMK11	ITMK10	PMK9	PMK8

Address: FFFD7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	TRJMK1	TRJMK0 DSADECMK	SREMK4 SREMGMK1	SRMK4 SRMGMK1	STMK4 STMGMK1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution For details about the bits, see Table 27-2. Be sure to set bits that are not available to the initial value.

27.3.3 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.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 27-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (1/3)

Address: FFFE8H After reset: FFH 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

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

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SRPR00	IICAPR00	STPR00 CSIPR000 IICPR000	AESPR0 AESFPR0	CRPR0 SMPPR000 SMPPR010	SREPR02 SREMGPR00	SRPR02 SRMGPR00	STPR02 STMGPR00

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SRPR10	IICAPR10	STPR10 CSIPR100 IICPR100	AESPR1 AESFPR1	CRPR1 SMPPR100 SMPPR110	SREPR12 SREMGPR10	SRPR12 SRMGPR10	STPR12 STMGPR10

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR001	FMPR0 SMPPR002 SMPPR012	RTCAPR00 SMPPR001 SMPPR011	TMPR000	SREPR01 TMPR003H	SRPR01	STPR01 CSIPR010 IICPR010	SREPR00 TMPR001H

Figure 27-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (2/3)

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR101	FMPR1 SMPPR102 SMPPR112	RTCAPR10 SMPPR101 SMPPR111	TMPR100	SREPR11 TMPR103H	SRPR11	STPR11 CSIPR110 IICPR110	SREPR10 TMPR101H

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	SRPR03	STPR03 CSIPR030 IICPR030	KRPR0 RTCAPR01 SMPPR004 SMPPR014	TMKAPR0	RTCRPR0 SMPPR003 SMPPR013	ADPR0	TMPR003	TMPR002

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	SRPR13	STPR13 CSIPR130 IICPR130	KRPR1 RTCAPR11 SMPPR104 SMPPR114	TMKAPR1	RTCRPR1 SMPPR103 SMPPR113	ADPR1	TMPR103	TMPR102

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	RTCIPR00 PPR014	RTCIPR01 PPR013	RTCIPR02 PPR012	PPR07	PPR06	TMPR005 SMOTAPR00	TMPR004	DSAPR0 SMPPR005 SMPPR015

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	RTCIPR10 PPR114	RTCIPR11 PPR113	RTCIPR12 PPR112	PPR17	PPR16	TMPR105 SMOTAPR10	TMPR104	DSAPR1 SMPPR105 SMPPR115

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	FLPR0	OSDPR0	MACPR0 SMOTBPR01	SREPR03	ITPR001	ITPR000	TMPR007 SMOTAPR01	TMPR006 SMOTBPR00

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	FLPR1	OSDPR1	MACPR1 SMOTBPR11	SREPR13	ITPR101	ITPR100	TMPR107 SMOTAPR11	TMPR106 SMOTBPR10

Figure 27-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (3/3)

Address: FFFDAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	LVDEXPR0 ITPR031	LVDVRPR0 ITPR030	LVDVBPR0 ITPR021	LVDVDPR0 ITPR020	ITPR011	ITPR010	PPR09	PPR08

Address: FFFDEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	LVDEXPR1 ITPR131	LVDVRPR1 ITPR130	LVDVBPR1 ITPR121	LVDVDPR1 ITPR120	ITPR111	ITPR110	PPR19	PPR18

Address: FFFDBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	TRJPR01	TRJPR00 DSADECPR0	SREPR04 SREMGPR01	SRPR04 SRMGPR01	STPR04 STMGPR01

Address: FFFDFH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	TRJPR11	TRJPR10 DSADECPR1	SREPR14 SREMGPR11	SRPR14 SRMGPR11	STPR14 STMGPR11

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 For details about the bits, see Table 27-2. Be sure to set bits that are not available to the initial value.

27.3.4 External interrupt rising edge enable register (EGP0, EGP1), External interrupt falling edge enable register (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP9, INTP12 to INTP14, and RTCIC0 to RTCIC2.

The EGP0, EGP1, EGN0 and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 27-5. Format of External Interrupt Rising Edge Enable Register (EGP0, EGP1) and External Interrupt Falling Edge Enable Register (EGN0, EGN1)

Address: FFF38H After reset: 00H 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

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	EGP14	EGP13	EGP12	0	0	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	EGN14	EGN13	EGN12	0	0	EGN9	EGN8

EGPn	EGNn	INTP0 to INTP9, INTP12 to INTP14, and RTCIC0 to RTCIC2 pin valid edge selection (n = 0 to 9, 12 to 14)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 27-3 shows the ports corresponding to the EGPn and EGNn bits.

Table 27-3. Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Interrupt Request Signal	80, 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	√
EGP8	EGN8	INTP8	√
EGP9	EGN9	INTP9	√
EGP12	EGP12	RTCIC2/INTP12	√
EGP13	EGN13	RTCIC1/INTP13	√
EGP14	EGN14	RTCIC0/INTP14	√

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).

Remarks 1. For edge detection port, see **2.1 Port Function.**
2. n = 0 to 9, 12 to 14

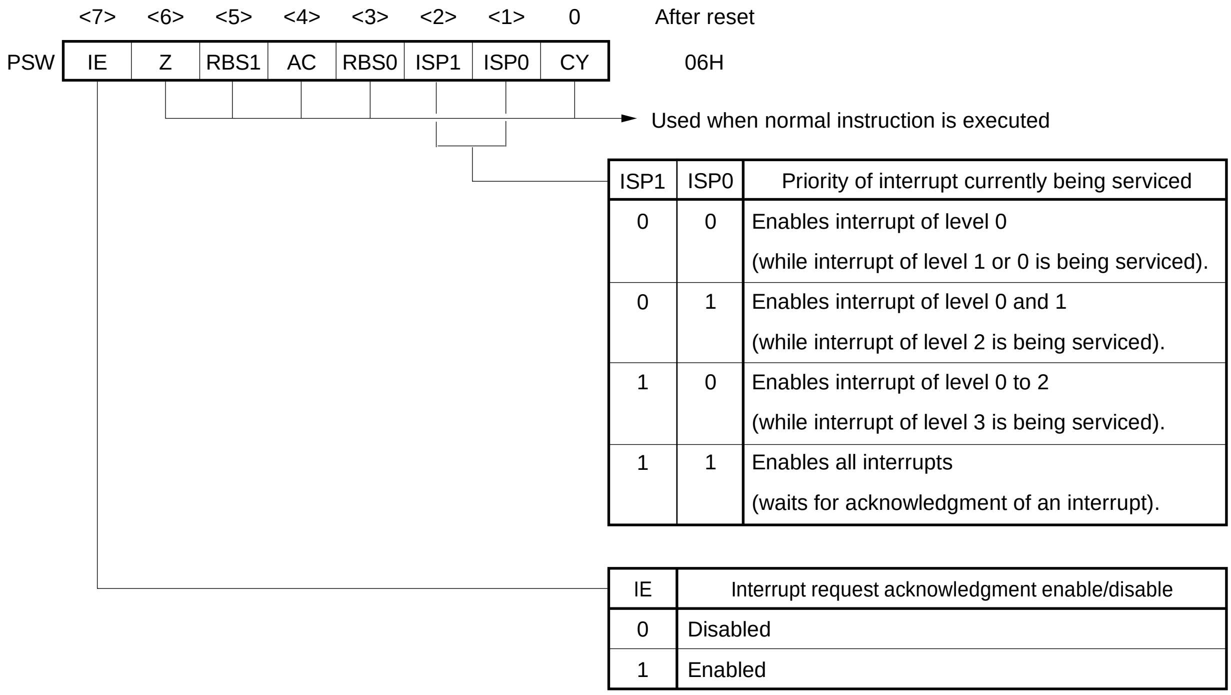
27.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls 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.

Reset signal generation sets PSW to 06H.

Figure 27-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

27.4 Interrupt Servicing Operations

27.4.1 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 from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in **Table 27-4** below.

For the interrupt request acknowledgment timing, see **Figures 27-8** and **27-9**.

Table 27-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

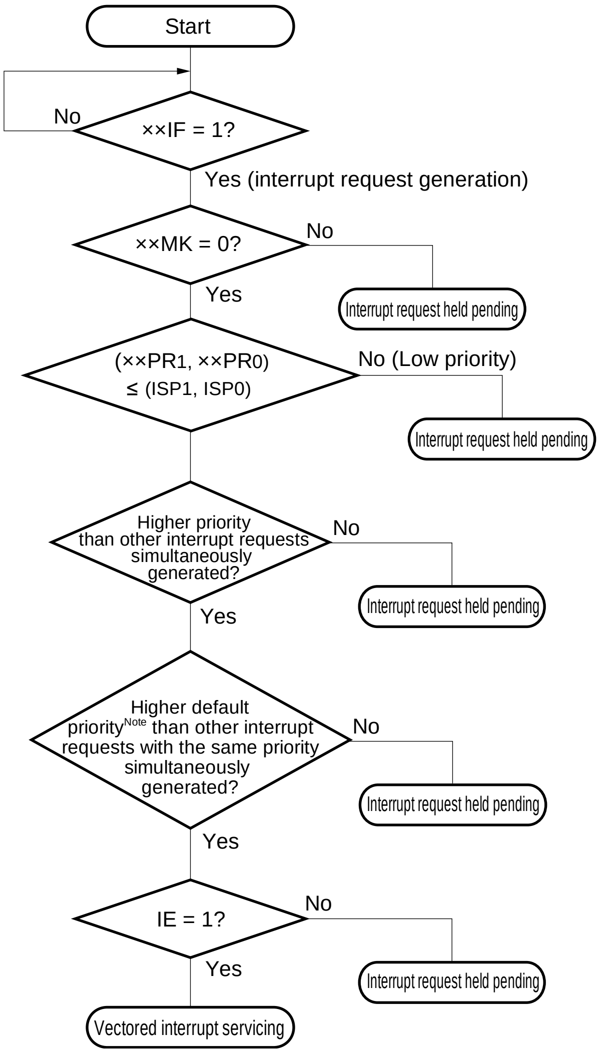
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 27-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

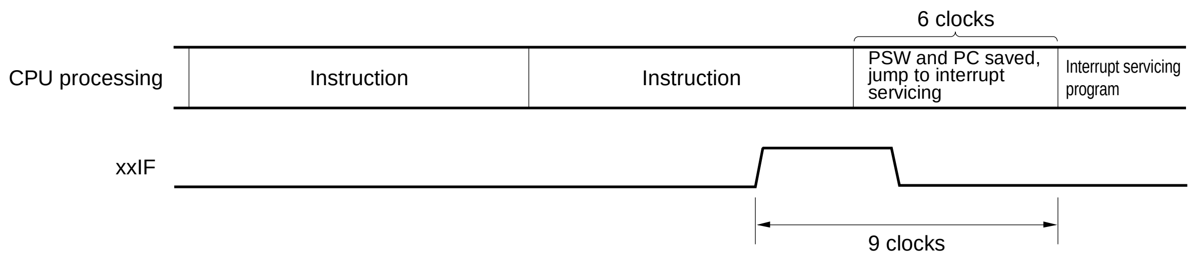
Figure 27-7. Interrupt Request Acknowledgment Processing Algorithm


Start
No
××IF = 1?
Yes (interrupt request generation)
××MK = 0?
No
Interrupt request held pending
Yes
(××PR1, ××PR0)
≤ (ISP1, ISP0)
No (Low priority)
Interrupt request held pending
Higher priority
than other interrupt requests
simultaneously
generated?
No
Interrupt request held pending
Yes
Higher default
priorityNote than other interrupt
requests with the same priority
simultaneously
generated?
No
Interrupt request held pending
Yes
IE = 1?
No
Interrupt request held pending
Yes
Vectored interrupt servicing

××IF:	Interrupt request flag
××MK:	Interrupt mask flag
××PR0:	Priority specification flag 0
××PR1:	Priority specification flag 1
IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1:	Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 27-6**)

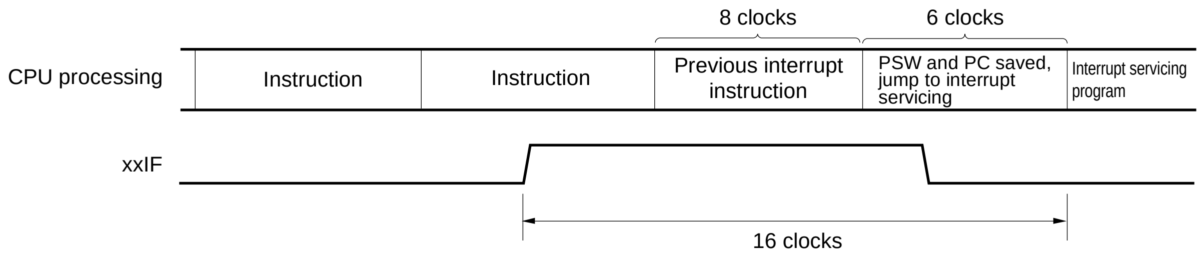
Note For the default priority, see **Table 27-1 Interrupt Source List**.

Figure 27-8. Interrupt Request Acknowledgment Timing (Minimum Time)


6 clocks
CPU processing
Instruction
Instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
xxIF
9 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 27-9. Interrupt Request Acknowledgment Timing (Maximum Time)


8 clocks
6 clocks
CPU processing
Instruction
Instruction
Previous interrupt instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
xxIF
16 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

27.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

27.4.3 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 (1) the IE flag 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. 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 execution.

Table 27-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 27-10** shows multiple interrupt servicing examples.

Table 27-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request Interrupt Being Serviced		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	
Maskable interrupt	ISP1 = 0 ISP0 = 0	O	×	×	×	×	×	×	×	O
	ISP1 = 0 ISP0 = 1	O	×	O	×	×	×	×	×	O
	ISP1 = 1 ISP0 = 0	O	×	O	×	O	×	×	×	O
	ISP1 = 1 ISP0 = 1	O	×	O	×	O	×	O	×	O
Software interrupt		O	×	O	×	O	×	O	×	O

Remarks 1. O: Multiple interrupt servicing enabled

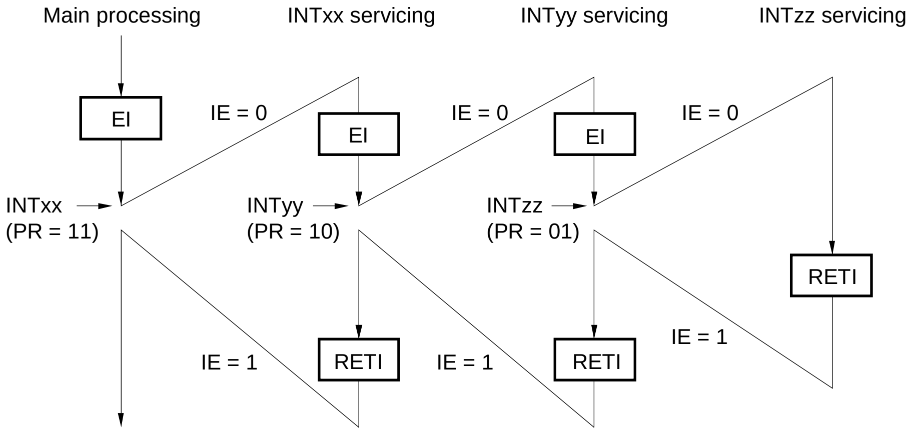
2. ×: Multiple interrupt servicing disabled

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.

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 ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

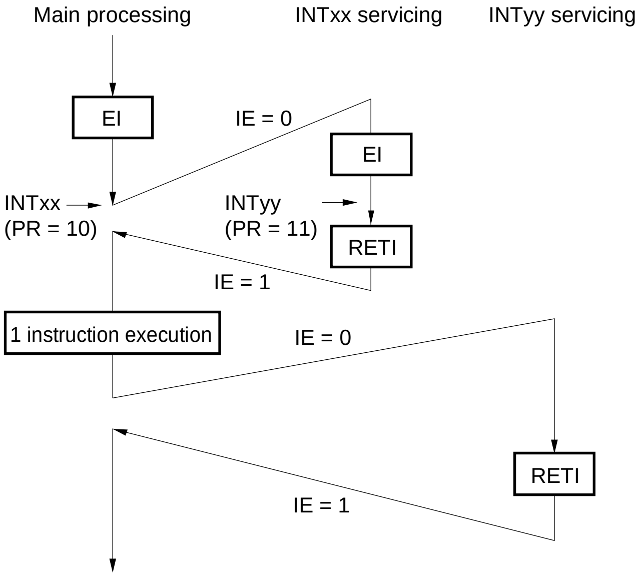
Figure 27-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

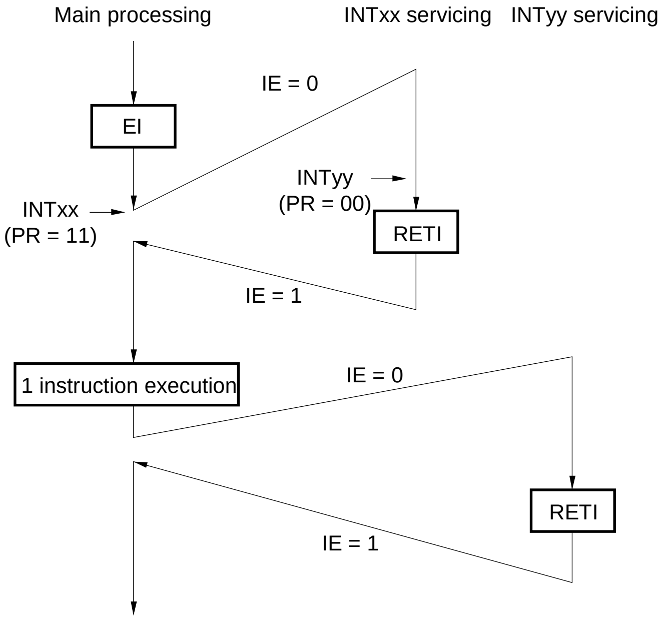


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00:	Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01:	Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10:	Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11:	Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0:	Interrupt request acknowledgment is disabled
IE = 1:	Interrupt request acknowledgment is enabled.

Figure 27-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

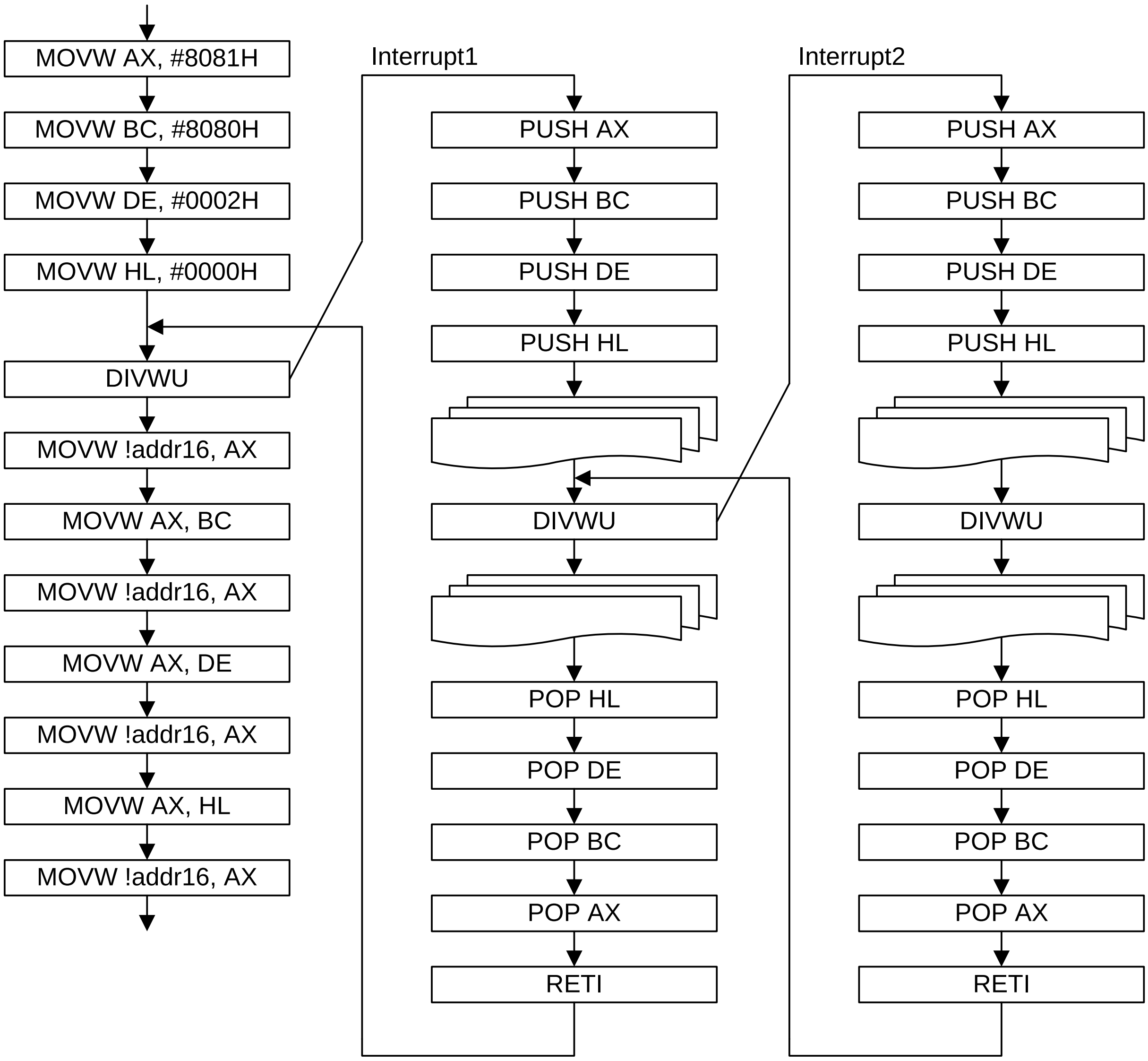
27.4.4 Interrupt servicing during division instruction

The RL78/I1C (512 KB) handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← (PC)S	(SP-2) ← (PC-3)S
(SP-3) ← (PC)H	(SP-3) ← (PC-3)H
(SP-4) ← (PC)L	(SP-4) ← (PC-3)L
PCS ← 0000	PCS ← 0000
PCH ← (Vector)	PCH ← (Vector)
PCL ← (Vector)	PCL ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



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. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

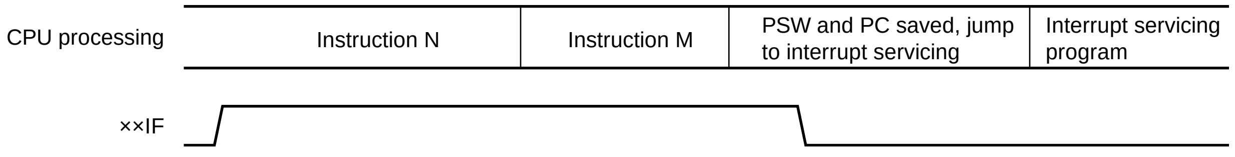
27.4.5 Interrupt request hold

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 (interrupt request hold instructions) 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
- MULHU
- MULH
- MACHU
- MACH
- 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 27-11 shows the timing at which interrupt requests are held pending.

Figure 27-11. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 28 KEY INTERRUPT FUNCTION

28.1 Functions of 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 28-1. Assignment of 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

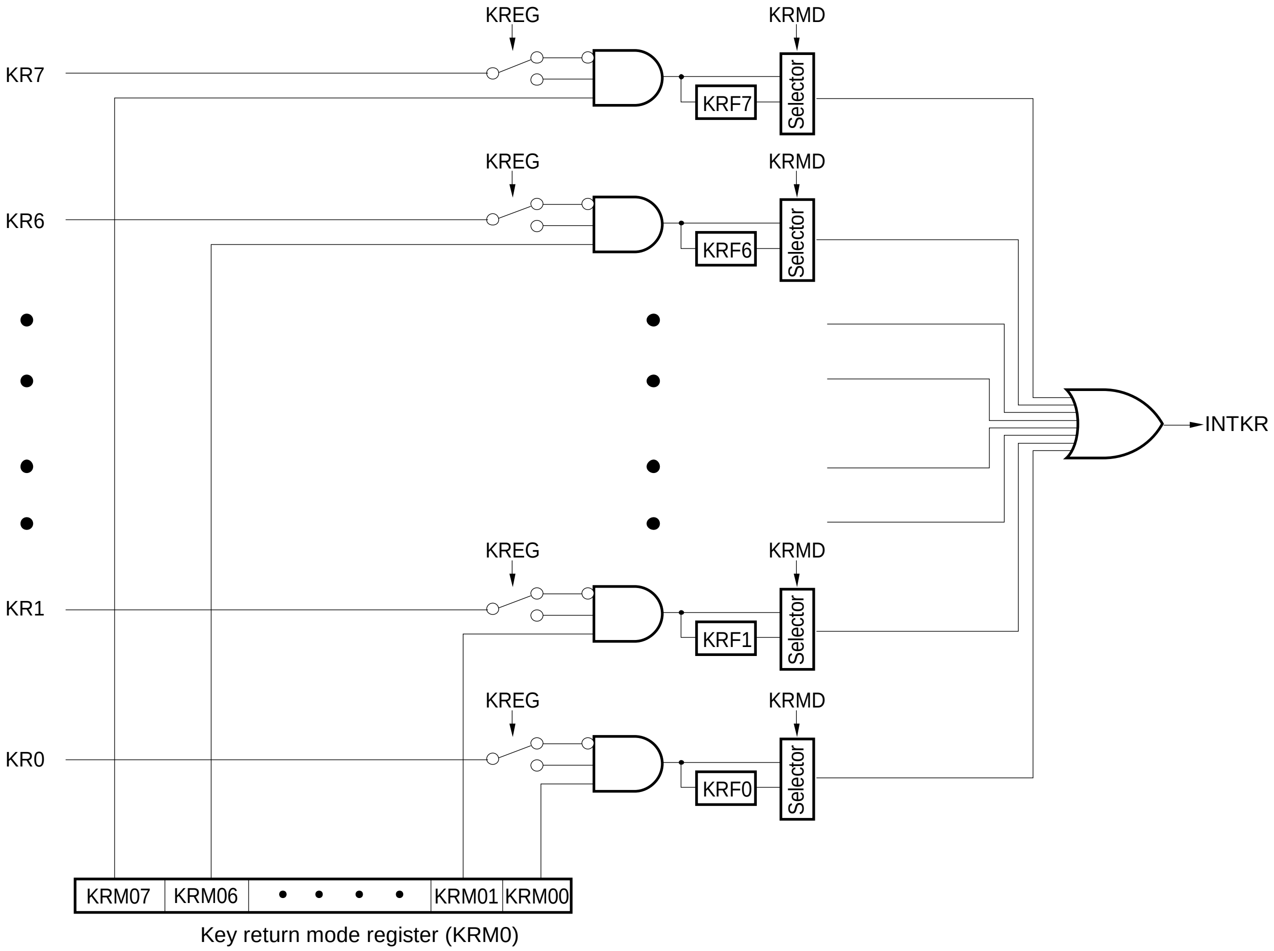
28.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 28-2. Configuration of Key Interrupt

Item	Configuration
Control registers	Key return control register (KRCTL) Key return mode register 0 (KRM0) Key return flag register (KRF) Port mode register 7 (PM7)

Figure 28-1. Block Diagram of Key Interrupt



28.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following registers.

- Key return control register (KRCTL)
- Key return mode register 0 (KRM0)
- Key return flag register (KRF)
- Port mode register 7 (PM7)

28.3.1 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 an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 28-2. Format of Key Return Control Register (KRCTL)

Address: FFF34H After reset: 00H 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

28.3.2 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 an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 28-3. Format of Key Return Mode Register 0 (KRM0)

Address: FFF37H After reset: 00H 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

Cautions 1. The on-chip pull-up resistors can be applied by setting the corresponding key interrupt input pins (bits) in pull-up resistor register 7 (PU7) to 1.

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 43.4 AC Characteristics).

3. The pins not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 7

28.3.3 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.

Reset signal generation clears this register to 00H.

Figure 28-4. Format of Key Return Flag Register (KRF)

Address: FFF35H After reset: 00H 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 to 1 is invalid. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

28.3.4 Port mode register 7 (PM7)

These registers set the input and output of port 7 in 1-bit units.

To use a key interrupt input (KR0 to KR7), set 1 to the bit of port mode register (PM7) corresponding to each port.

The PM7 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to FFH.

Figure 28-5. Format of Port Mode Register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode selection for PM7n pin (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 29 STANDBY FUNCTION

29.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation 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

STOP instruction execution sets 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 cleared 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

When reception of data through CSI0 or UART0 or activation of the DTC leads to release from STOP mode, reception of data through CSI0 or UART0 or activation of the DTC proceeds without CPU intervention. This mode is only available when the high-speed on-chip oscillator or middle-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

3. When using CSI0 or UART0 in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see 20.3 Registers Controlling Serial Array Unit.

4. It can be selected by the WDTON bit of the option byte and 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 (2) <2> Low-speed on-chip oscillator.

29.2 Registers Controlling Standby Function

The registers which control the standby function are described below.

- Subsystem clock supply option control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 6 CLOCK GENERATOR**. For registers which control the SNOOZE mode, see **CHAPTER 20 SERIAL ARRAY UNIT**.

29.3 Standby Function Operation

29.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 29-1. Operating Statuses in HALT Mode (1/4)

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Main System Clock				
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})	When CPU is Operating on PLL Clock Frequency (f_{PLL})
System clock			Clock supply to the CPU is stopped				
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	Operation disabled		Operation continues (cannot be stopped)
		f_{IM}	Operation disabled	Operation continues (cannot be stopped)			Operation disabled
		f_X	Operation disabled		Operation continues (cannot be stopped)	Cannot operate	
		f_{EX}			Cannot operate	Operation continues (cannot be stopped)	
		f_{PLL}	Operation disabled				Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained. (Operation disabled when RTC power-on-reset occurs)				
		f_{EXS}					
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply option control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited while the sub clock (f_{SX}) operates.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop				
CPU			Operation stopped				
Code flash memory							
Data flash memory							
RAM			Operation stopped (Operable while in the DTC is executed)				
Port (latch)			Status before HALT mode was set is retained. (Rewriting the port register by the DTC can change the port pin setting.)				
Timer array unit			Operable				
Independent power supply real-time clock (RTC)			Operable (Operation stopped when RTC power-on-reset occurs)				
Frequency measurement function			Operation disable	Operable			
High-speed on-chip oscillator clock frequency correction function			Operable (when f_{XT} or f_{EXS} is supplied)	Operation disabled			
Oscillation stop detection			Operable (only when f_{IL} is oscillating)				
12-bit interval timer			Operable				
8-bit interval timer							
Sampling output timer detector							
Timer RJ							
Watchdog timer			See **CHAPTER 16 WATCHDOG TIMER**.				
Clock output/buzzer output			Operable				
12-bit A/D converter							
24-bit ΔΣ A/D converter							
Temperature sensor 2							
Serial array unit (SAU)							
IrDA							
Serial interface (IICA)							
Serial interface UARTMG							
LCD controller/driver							
Data transfer controller (DTC)							
Event link controller (ELC)			Operable function blocks can be linked				

Remark Operation stopped: Operation is automatically stopped before switching to HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.

f_{IH}: High-speed on-chip oscillator clock	f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock	f_{PLL}: PLL clock frequency
f_{EX}: External main system clock	f_X: X1 clock
f_{EXS}: External subsystem clock	f_{XT}: XT1 clock

Table 29-1. Operating Statuses in HALT Mode (2/4)

HALT Mode Setting Item		When HALT Instruction is Executed While CPU is Operating on Main System Clock				
		When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})	When CPU is Operating on PLL Clock Frequency (f_{PLL})
32-bit multiplier and multiply accumulator		Operation disable				
AES circuit		Operable				
Power-on-reset function						
RTC power-on-reset function						
Voltage detection function	V_{DD}					
	V_{DD}, LVDVBAT, VRTC, EXLVD pin supply voltage					
External interrupt	INTP0 to INTP9, INTP12 to INTP14					
	RTCIC0 to RTCIC2					
Key interrupt function						
CRC operation function	High-speed CRC					
	General-purpose CRC	In the calculation of the RAM area, operable when DTC is executed only				
Illegal-memory access detection function		Operation stopped (Operable when DTC is executed.)				
RAM parity error detection function						
RAM guard function						
SFR guard function						

Remark Operation stopped: Operation is automatically stopped before switching to HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.

f_{IH}: High-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock

f_{IL}: Low-speed on-chip oscillator clock
f_{PLL}: PLL clock frequency
f_X: X1 clock
f_{XT}: XT1 clock

Table 29-1. Operating Statuses in HALT Mode (3/4)

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock		
			When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})	When CPU is Operating on Low-speed On-chip Oscillator Clock (f_{IL})
System clock			Clock supply to the CPU is stopped.		
	Main system clock	f_{IH}	Operation disabled		
		f_{IM}			
		f_X			
		f_{EX}			
		f_{PLL}			
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped) (Operation disabled when RTC power-on-reset occurs)	Cannot operate	Operation disabled
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped) (Operation disabled when RTC power-on-reset occurs)	
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of the option byte (000C0H) and bit 4 (WUTMMCK0) of the subsystem clock supply option control register (OSMC). ● WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (However, setting the WUTMMCK0 and SELLOSC bits to 1 is prohibited while the sub clock (f_{SX}) operates.) ● WUTMMCK0 = 0, SELLOSC = 0, and WDTON=0: Stops ● WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates ● WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stops		Operation continues (cannot be stopped)
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM			Operation stopped (Operable while in the DTC is executed)		
Port (latch)			Status before HALT mode was set is retained (rewriting the port register by the DTC can change the port pin setting).		
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
Independent power supply real-time clock (RTC)			Operable (Operation stopped when RTC power-on-reset occurs)		
Frequency measurement function			Operation disabled		
High-speed on-chip oscillator clock frequency correction function					
Oscillation stop detection					
12-bit Interval timer			Operable		
8-bit Interval timer					
Sampling output timer detector					
Timer RJ					
Watchdog timer			See **CHAPTER 16 WATCHDOG TIMER**.		
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting.		
12-bit A/D converter			Operation disabled		
24-bit ΔΣ A/D converter					
Temperature sensor 2					
Serial array unit (SAU)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
IrDA			Operation disabled		
Serial interface (IICA)			Operation disabled		
Serial interface UARTMG			Operable		
LCD controller/driver			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)		
Data transfer controller (DTC)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
Event link controller (ELC)			Operable function blocks can be linked		
32-bit multiplier and multiply accumulator			Operation disable		

(**Remark** is listed on the next page.)

Table 29-1. Operating Statuses in HALT Mode (4/4)

HALT Mode Setting Item		When HALT Instruction is Executed While CPU is Operating on Subsystem Clock		
		When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})	When CPU is Operating on Low-speed On-chip Oscillator Clock (f_{IL})
AES circuit		Operable		
Power-on-reset function				
RTC power-on-reset function				
Voltage detection function	V_{DD}			
	V_{DD}, LVDVBAT, VRTC, EXLVD pin supply voltage			
External interrupt	INTP0 to INTP9, INTP12 to INTP14			
	RTCIC0 to RTCIC2			
Key interrupt function				
CRC operation function	High-speed CRC	Operation disabled		
	General-purpose CRC	In the calculation of the RAM area, operable when DTC is executed only		
Illegal-memory access detection function		Operable when DTC is executed only		
RAM parity error detection function				
RAM guard function				
SFR guard function				

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock f_{PLL}: PLL clock frequency

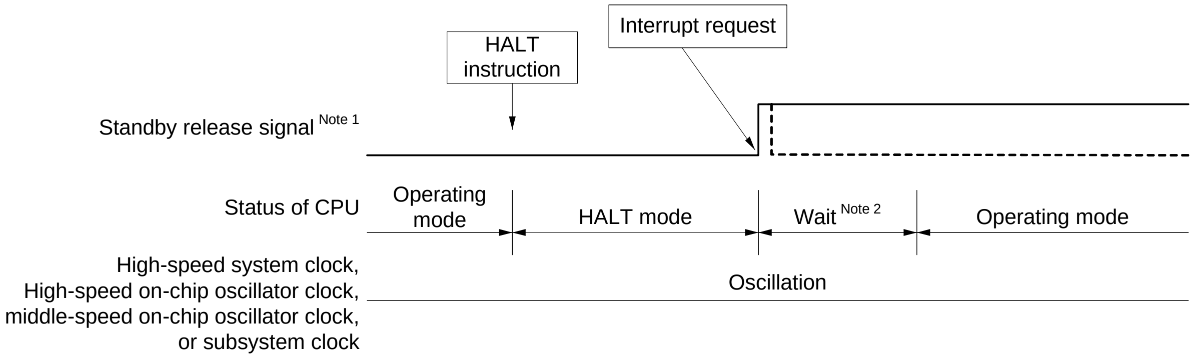
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked 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 next address instruction is executed.

Figure 29-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out

Main system clock:	15 to 16 clocks
Subsystem clock (RTCLPC = 0):	10 to 11 clocks
Subsystem clock (RTCLPC = 1):	11 to 12 clocks

- When vectored interrupt servicing is not carried out

Main system clock:	9 to 10 clocks
Subsystem clock (RTCLPC = 0):	4 to 5 clocks
Subsystem clock (RTCLPC = 1):	5 to 6 clocks

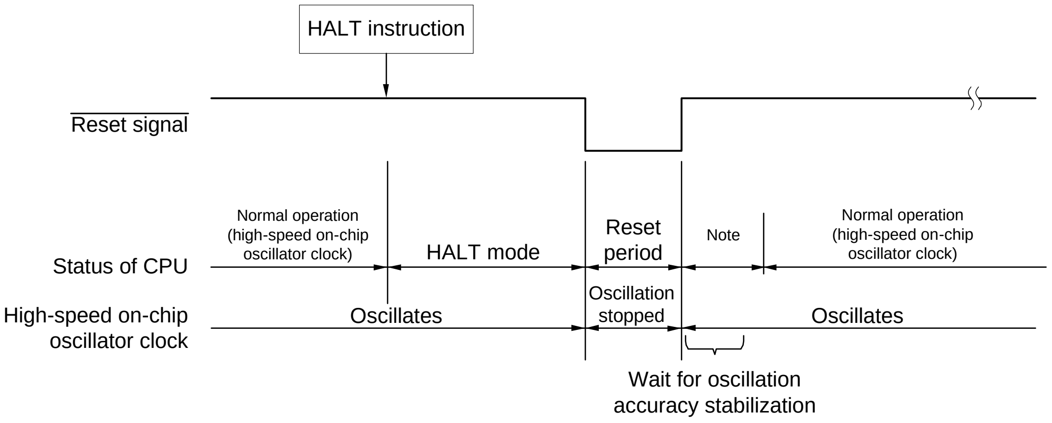
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

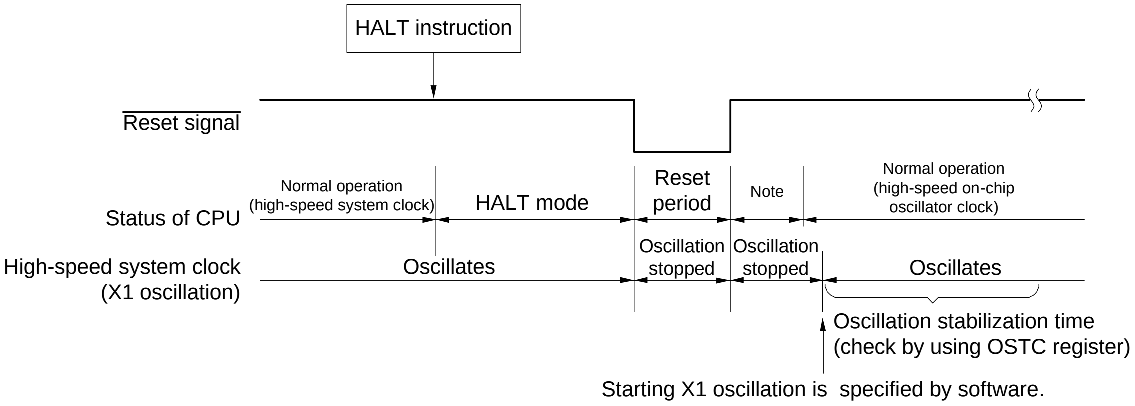
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 29-2. HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (high-speed on-chip oscillator clock)
HALT mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

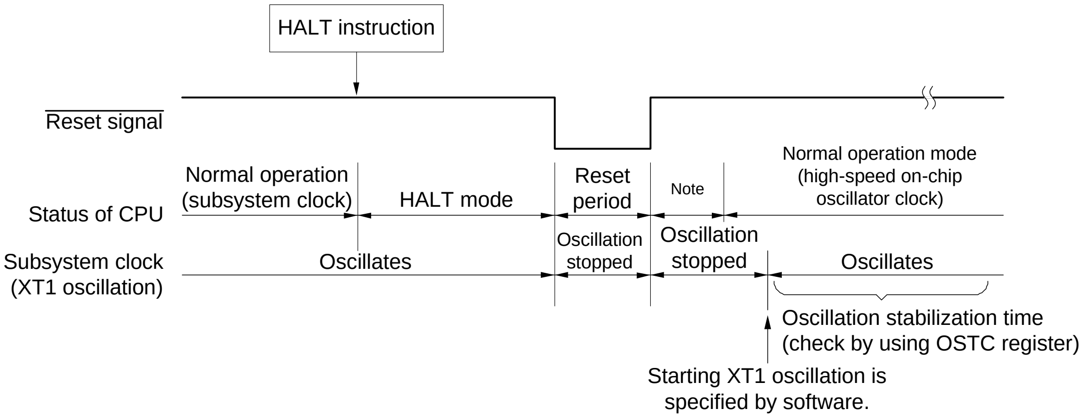
(2) When high-speed system clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
HALT mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (check by using OSTC register)
Starting X1 oscillation is specified by software.

Note For the reset processing time, see **CHAPTER 30 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 31 POWER-ON-RESET CIRCUIT**.

Figure 29-2. HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (subsystem clock)
HALT mode
Reset period
Note
Normal operation mode (high-speed on-chip oscillator clock)
Subsystem clock (XT1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (check by using OSTC register)
Starting XT1 oscillation is specified by software.

Note For the reset processing time, see **CHAPTER 30 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 31 POWER-ON-RESET CIRCUI**T.

29.3.2 STOP mode

(1) STOP mode setting and operating statuses

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 high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation.
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 statuses in the STOP mode are shown below.

Table 29-2. Operating Statuses in STOP Mode (1/2)

STOP Mode Setting Item			When STOP Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Stopped			
		f_{IM}				
		f_X				
		f_{EX}				
		f_{PLL}				
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained (Operation disabled when RTC power-on-reset occurs)			
		f_{EXS}				
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply option control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited while the sub clock (f_{SX}) operates.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop			
CPU			Operation stopped			
Code flash memory						
Data flash memory			Operation stopped (Transition to the STOP mode is not possible while rewriting the data flash memory)			
RAM			Operation stopped			
Port (latch)			Status before STOP mode was set is retained			
Timer array unit			Operation disabled			
Independent power supply real-time clock (RTC)			Operable (Operation stopped when RTC power-on-reset occurs)			
Frequency measurement function			Operation disabled			
High-speed on-chip oscillator clock frequency correction function						
Oscillation stop detection			Operable (only when f_{IL} is oscillating)			
12-bit interval timer			Operable			
8-bit interval timer						
Sampling output timer detector						
Timer RJ			● Operable in event count mode when TRJIO input with no filer is selected ● Operable when the sub clock (f_{SX}) is selected as the count source and RTCLPC in the OSMC register is 0 ● Operable when the low-speed on-chip oscillator clock frequency (f_{IL}) is selected as the count source ● Operation is disabled under any conditions other than the above			
Watchdog timer			See **CHAPTER 16 WATCHDOG TIMER**.			
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting.			
12-bit A/D converter			Operation stopped			
24-bit ΔΣ A/D converter			Operation disabled			
Temperature sensor 2						
Serial array unit (SAU)			Wakeup operation is enabled only for CSI00 and UART0 (switching to SNOOZE mode). Operation is disabled for anything other than CSI00 and UART0.			
IrDA			Operation disabled			
Serial interface (IICA)			Wakeup by address match operable			
Serial interface UARTMG			Operable			
LCD controller/driver			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)			
Data transfer controller (DTC)			Operable			
Event link controller (ELC)			Operable function blocks can be linked			
32-bit multiplier and multiply accumulator			Operation disabled			
AES circuit						

(**Remark** is listed on the next page.)

Table 29-2. Operating Statuses in STOP Mode (2/2)

Item \ STOP Mode Setting		When STOP Instruction is Executed While CPU is Operating on Main System Clock			
		When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
Power-on-reset function		Operable			
RTC power-on-reset function					
Voltage detection function	V_{DD}				
	V_{DD}, LVDVBAT, VRTC, EXLVD pin supply voltage				
External interrupt	INTP0 to INTP9, INTP12 to INTP14				
	RTCIC0 to RTCIC2				
Key interrupt function					
CRC operation function	High-speed CRC	Operation stopped			
	General-purpose CRC				
Illegal-memory access detection function					
RAM parity error detection function					
RAM guard function					
SFR guard function					

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.

Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock	f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock	f_X: X1 clock
f_{EX}: External main system clock	f_{XT}: XT1 clock
f_{EXS}: External subsystem clock	f_{PLL}: PLL clock frequency

(2) Release from STOP mode

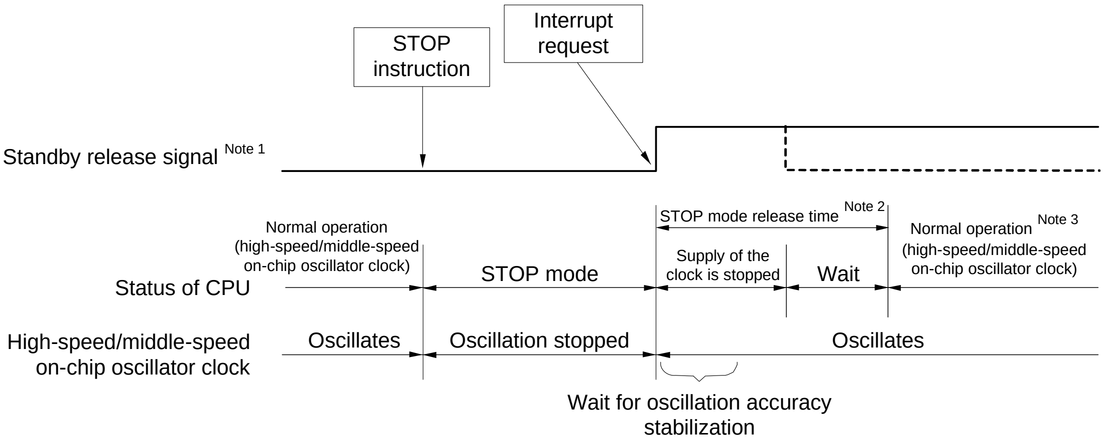
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked 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 next address instruction is executed.

Figure 29-3. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed/middle-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Interrupt request
Standby release signal Note 1
Normal operation (high-speed/middle-speed on-chip oscillator clock)
STOP mode
STOP mode release time Note 2
Supply of the clock is stopped
Wait
Normal operation Note 3 (high-speed/middle-speed on-chip oscillator clock)
Status of CPU
High-speed/middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

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

Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time

Supply of the clock is stopped:

When high-speed on-chip oscillator clock: 18 μs to 65 μs

When middle-speed on-chip oscillator clock: 22 μs to 31 μs (in HS mode)
Up to 3.4 μs (during operation at 4 MHz in LS mode)
Up to 4.2 μs (during operation at 2 MHz in LS mode)
Up to 5.9 μs (during operation at 1 MHz in LS mode)
Up to 5.9 μs (during operation at 1 MHz in LP mode)

Wait:

(common to the high-speed/middle-speed on-chip oscillator clock)

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

3. Before switching the operating clock from the CPU/peripheral hardware clock (f_{CLK}) to the high-speed on-chip oscillator clock after using the middle-speed on-chip oscillator clock for the transition from STOP mode to normal mode, use software to set up waiting for the corresponding period from the list below.

In HS mode: 24 μs
In LS mode: 10 μs
In LP mode: 7 μs

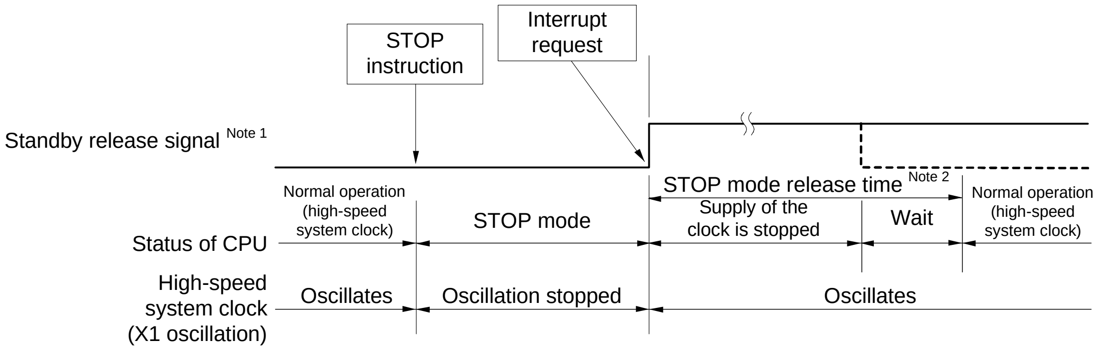
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 29-3. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock


STOP instruction
Interrupt request
Standby release signal Note 1
STOP mode release time Note 2
Normal operation (high-speed system clock)
STOP mode
Supply of the clock is stopped
Wait
Normal operation (high-speed system clock)
Status of CPU
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillates

Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time

Supply of the clock is stopped:

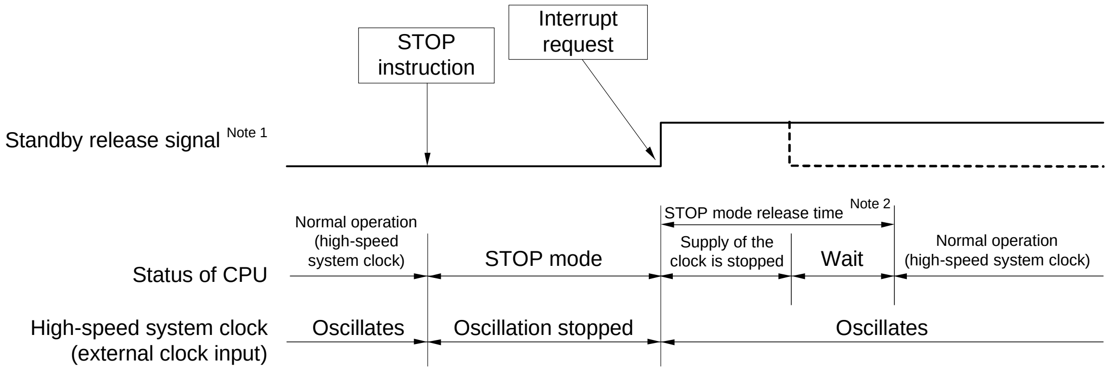
18 µs to “whichever is longer 65 µs or the oscillation stabilization time (set by OSTS)”

Wait:

● When vectored interrupt servicing is carried out: 10 to 11 clocks

● When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock


STOP instruction
Interrupt request
Standby release signal Note 1
STOP mode release time Note 2
Normal operation (high-speed system clock)
STOP mode
Supply of the clock is stopped
Wait
Normal operation (high-speed system clock)
Status of CPU
High-speed system clock (external clock input)
Oscillates
Oscillation stopped
Oscillates

Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time

Supply of the clock is stopped: 18 µs to 65 µs

Wait:

● When vectored interrupt servicing is carried out: 7 clocks

● When vectored interrupt servicing is not carried out: 1 clock

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

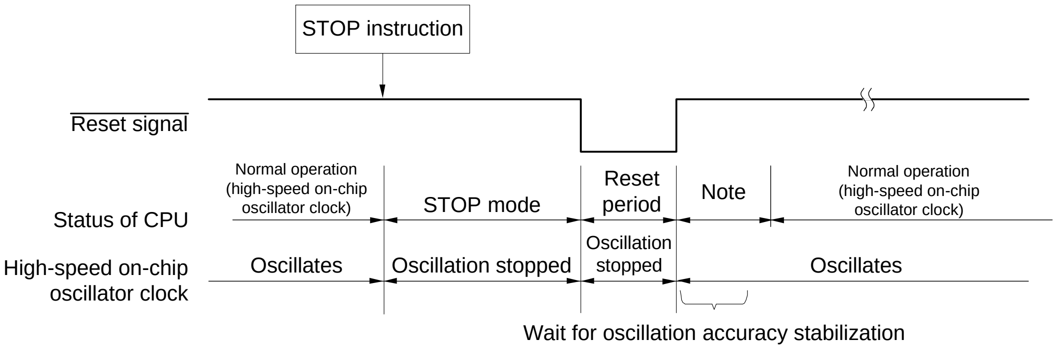
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

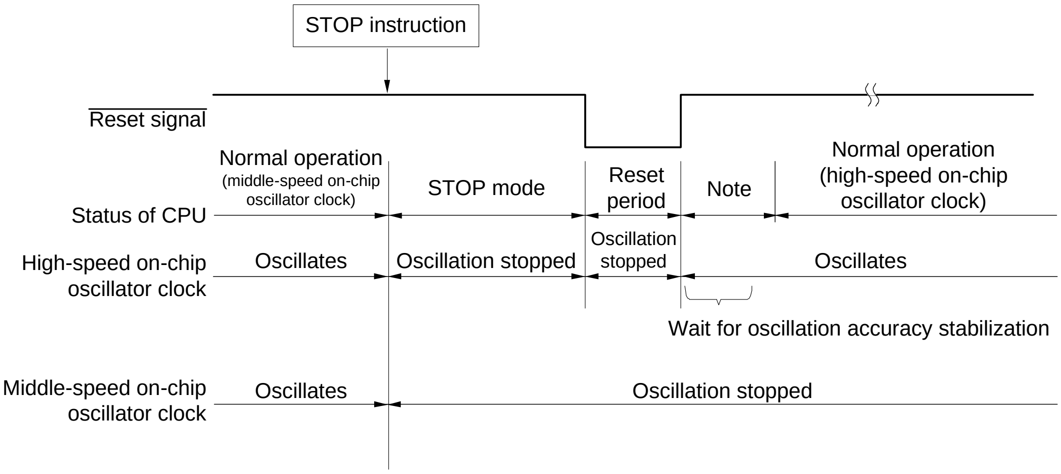
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 29-4. STOP Mode Release by Reset

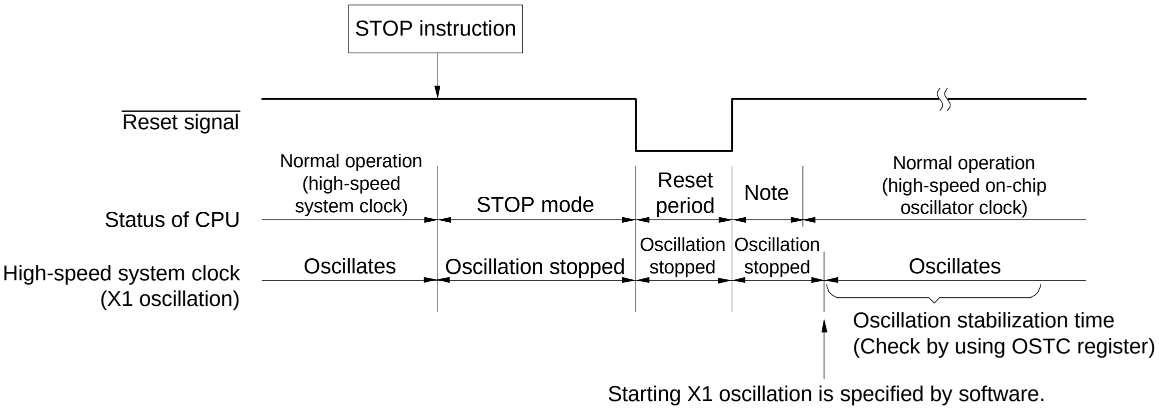
(1) When high-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (high-speed on-chip oscillator clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

(2) When middle-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (middle-speed on-chip oscillator clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization
Middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped

(3) When high-speed system clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (Check by using OSTC register)
Starting X1 oscillation is specified by software.

Note For the reset processing time, see **CHAPTER 30 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 31 POWER-ON-RESET CIRCUIT**.

29.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI0, UART0, or DTC. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock[Note].

When using CSI0 or UART0 in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see **20.3 Registers Controlling Serial Array Unit**.

When DTC transfer is used in SNOOZE mode, before switching to the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **25.3 Registers Controlling DTC**.

Note When using UART reception to transition from STOP mode to SNOOZE mode, use the high-speed on-chip oscillator.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode:

When high-speed on-chip oscillator clock: 18 μs to 65 μs

When middle-speed on-chip oscillator clock: 22 μs to 31 μs (in HS mode)
Up to 3.4 μs (during operation at 4 MHz in LS mode)
Up to 4.2 μs (during operation at 2 MHz in LS mode)
Up to 5.9 μs (during operation at 1 MHz in LS mode)
Up to 5.9 μs (during operation at 1 MHz in LP mode)

Remark 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:

When high-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 7 clocks
  LS (Low-speed main) mode: “1.3 μs to 4.5 μs” + 7 clocks
  LV (Low-voltage main) mode: “17.5 μs to 25.2 μs” + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 1 clock
  LS (Low-speed main) mode: “1.3 μs to 4.5 μs” + 1 clock
  LV (Low-voltage main) mode: “17.5 μs to 25.2 μs” + 1 clock

When middle-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode: “6.0 μs to 10.3 μs” + 7 clocks
  LS (Low-speed main) mode: “1.8 μs to 4.9 μs” + 7 clocks
  LP (Low-power main) mode: “3.8 μs to 4.9 μs” + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode: “6.0 μs to 10.3 μs” + 1 clock
  LS (Low-speed main) mode: “1.8 μs to 4.9 μs” + 1 clock
  LP (Low-power main) mode: “3.8 μs to 4.9 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

Table 29-3. Operating Statuses in SNOOZE Mode (1/2)

STOP Mode Setting Item			During STOP Mode, Reception of Data through CSI0 or UART0 or Activation of the DTC by an Interrupt	
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation started	Stopped
		f_{IM}	Stopped	Operation started
		f_X	Stopped	
		f_{EX}		
		f_{PLL}		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained (Operation disabled when RTC power-on-reset occurs)	
		f_{EXS}		

	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply option control register (OSMC)
			WUTMMCK0 = 1 or SELLOSC = 1: Oscillates
			(Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited while the sub clock (f_{SX}) operates.)
			WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop
			WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates
			WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop	

CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM			Operation stopped (Operable while in the DTC is executed)	
Port (latch)			Use of the status while in the STOP mode continues (rewriting the port register by the DTC can change the port pin setting)	
Timer array unit			Operation disabled	
Independent power supply real-time clock (RTC)			Operable (Operation stopped when RTC power-on-reset occurs)	
Frequency measurement function			Operation disabled	
High-speed on-chip oscillator clock frequency correction function				
Oscillation stop detection			Operable (only when f_{IL} is oscillating)	
12-bit interval timer			Operable	
8-bit interval timer				
Sampling output timer detector			Operable when the sub clock (f_{SX}) is selected as the count source and RTCLPC in the OSMC register is 0.	
Timer RJ			● Operable in event count mode when TRJIO input with no filer is selected ● Operable when the sub clock (f_{SX}) is selected as the count source and RTCLPC in the OSMC register is 0 ● Operable when the low-speed on-chip oscillator clock frequency (f_{IL}) is selected as the count source ● Operation is disabled under any conditions other than the above	
Watchdog timer			See **CHAPTER 16 WATCHDOG TIMER**.	
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting.	
12-bit A/D converter			Operation stopped	
24-bit ΔΣ A/D Converter			Operation disabled	
Temperature sensor 2				
Serial array unit (SAU)			Operable only CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.	
IrDA			Operation disabled	
Serial interface (IICA)				
Serial interface UARTMG			Operable	
LCD controller/driver			Operable (However, this depends on the status of the clock selected as the LCD source clock: operation is possible if the selected clock is operating, but operation will stop if the selected clock is stopped.)	
Data transfer controller (DTC)			Operable	
Event link controller (ELC)			Operable function blocks can be linked	
32-bit multiplier and multiply accumulator			Operation disabled	
AES circuit				

(**Remark** is listed on the next page.)

Table 29-3. Operating Statuses in SNOOZE Mode (2/2)

STOP Mode Setting Item		During STOP Mode, Reception of Data through CSI0 or UART0 or Activation of the DTC by an Interrupt	
		When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})
Power-on-reset function		Operable	
RTC power-on-reset function			
Voltage detection function	V_{DD}		
	V_{DD}, LVDVBAT, VRTC, EXLVD pin supply voltage		
External interrupt	INTP0 to INTP9, INTP12 to INTP14		
	RTCIC0 to RTCIC2		
Key interrupt function			
CRC operation function	High-speed CRC	Operation stopped	
	General-purpose CRC		
Illegal-memory access detection function		Operable when executing the DTC	
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.

Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock

f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock

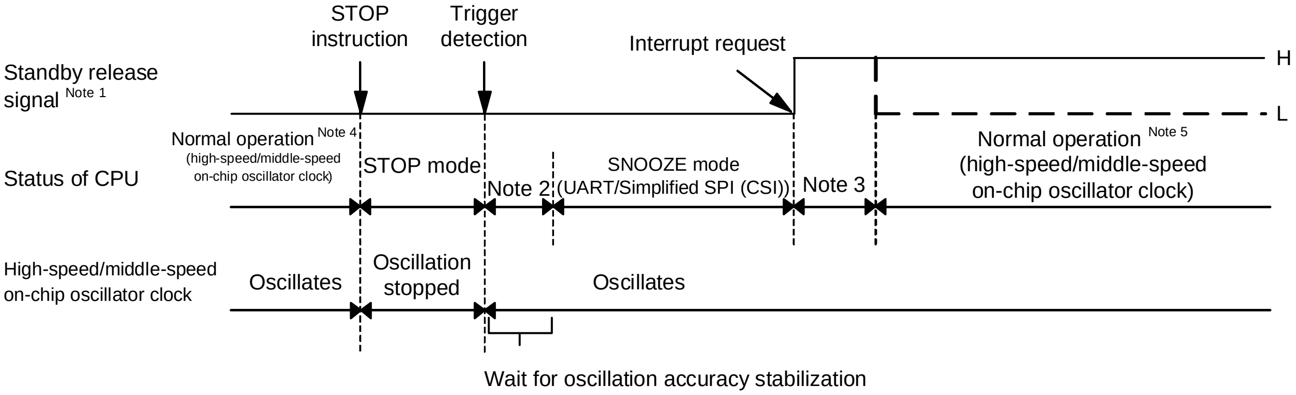
f_{EX}: External main system clock f_{XT}: XT1 clock

f_{EXS}: External subsystem clock f_{SX}: Sub clock

f_{PLL}: PLL clock frequency

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 29-5. When the Interrupt Request Signal is Generated in the SNOOZE Mode


STOP instruction
Trigger detection
Interrupt request
Standby release signal Note 1
H
L
Status of CPU
Normal operation Note 4 (high-speed/middle-speed on-chip oscillator clock)
STOP mode
Note 2
SNOOZE mode (UART/Simplified SPI (CSI))
Note 3
Normal operation Note 5 (high-speed/middle-speed on-chip oscillator clock)
High-speed/middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. Transition time from STOP mode to SNOOZE mode

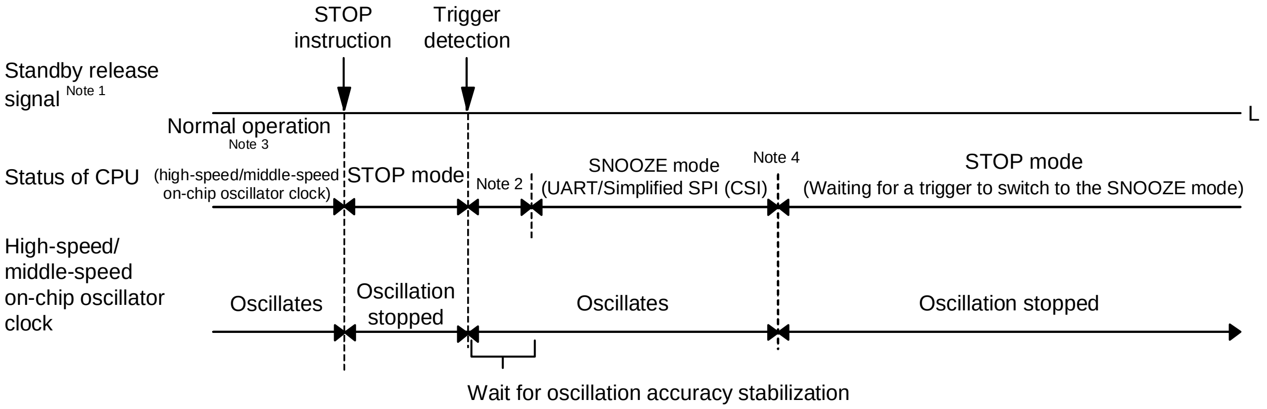
3. Transition time from SNOOZE mode to normal operation

4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 29-6. When the Interrupt Request Signal is not Generated in the SNOOZE Mode


STOP instruction
Trigger detection
Standby release signal Note 1
L
Status of CPU
Normal operation Note 3 (high-speed/middle-speed on-chip oscillator clock)
STOP mode
Note 2
SNOOZE mode (UART/Simplified SPI (CSI)
Note 4
STOP mode (Waiting for a trigger to switch to the SNOOZE mode)
High-speed/ middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stopped
Wait for oscillation accuracy stabilization

Notes 1. For details of the standby release signal, see **Figure 27-1 Basic Configuration of Interrupt Function**.

2. Transition time from STOP mode to SNOOZE mode

3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

4. If a standby release signal is generated in response to an interrupt from a module which is not set to operate in the SNOOZE mode during a transition of the chip from SNOOZE mode to STOP mode, the high-speed on-chip oscillator clock may run slowly for up to 15 μs from when the CPU starts to operate. If the clock frequency accuracy specified in the electrical characteristics is required immediately after release from standby, wait for the number of cycles at the actual CPU clock frequency that is equivalent to 15 μs.

Remark For details of the SNOOZE mode function, see **CHAPTER 20 SERIAL ARRAY UNIT**.

CHAPTER 30 RESET FUNCTION

The following eight operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit[Note 1]
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction[Note 2]
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access
(8) Reset of the RTC and XT1 oscillator by comparison of supply voltage of the RTC power-on reset (RTCPOR) circuit and detection voltage

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction[Note 2], RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in **Table 30-1**.

The RTC and XT1 oscillator are reset by the RTC power-on reset (RTCPOR).

Notes 1. If RTC power-on-reset does not occur, independent power supply RTC and XT1 oscillator circuit can be operated even during the reset period of (3) power-on-reset.

2. This reset occurs when instruction code FFH is executed.
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

Cautions 1. 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 43.4 AC Characteristics, and then input a high level to the pin.

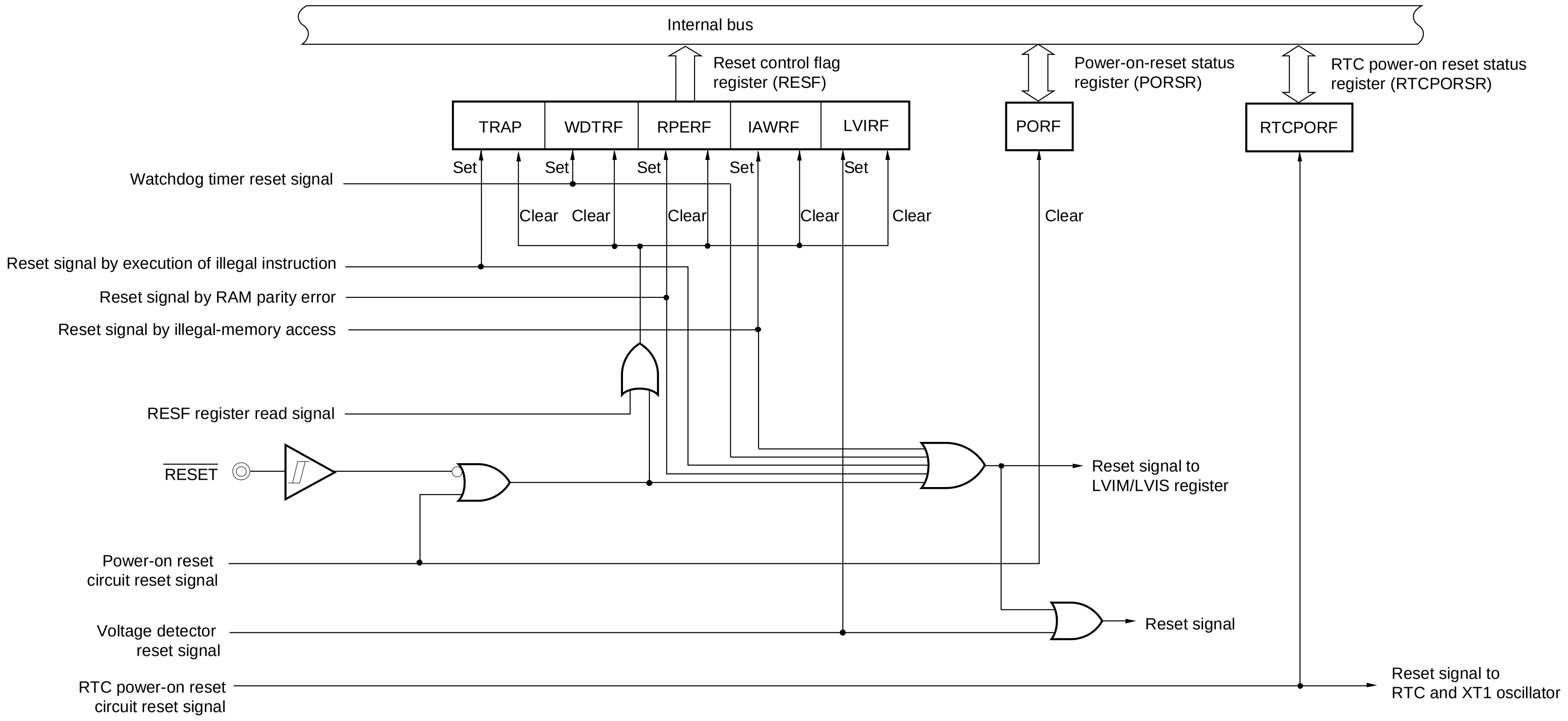
2. During reset input, the X1 clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating.

3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.

- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **P130: Low level during the reset period and after receiving a reset signal.**
- **Ports other than P40 and P130: High-impedance during the reset period or after receiving a reset signal.**

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Figure 30-1. Block Diagram of Reset Function



Caution An LVD circuit internal reset does not reset the LVD circuit.

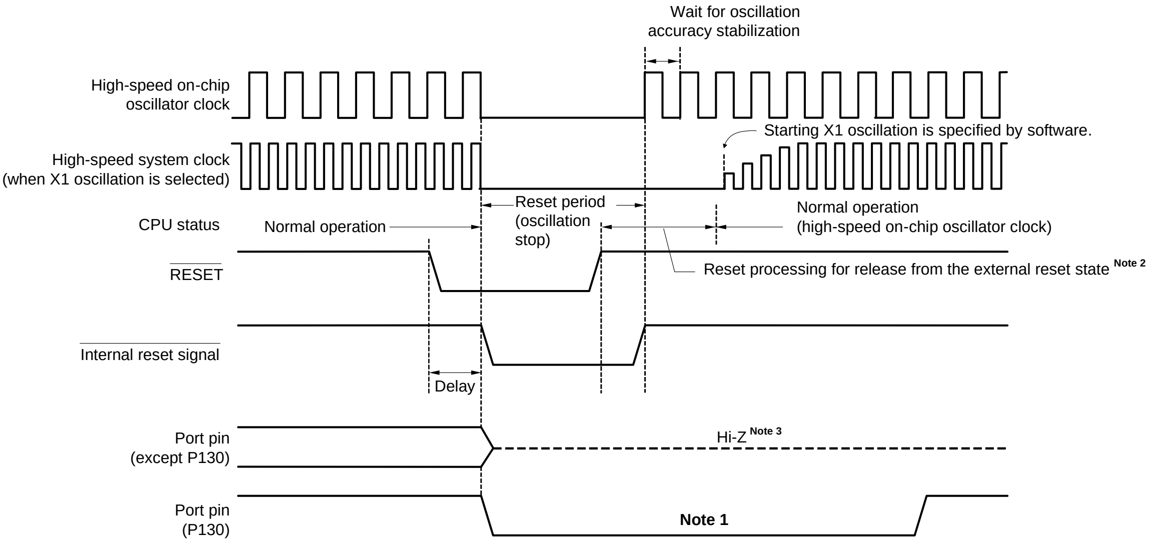
Remarks 1. LVIM: Voltage detection register

2. LVIS: Voltage detection level register

30.1 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.

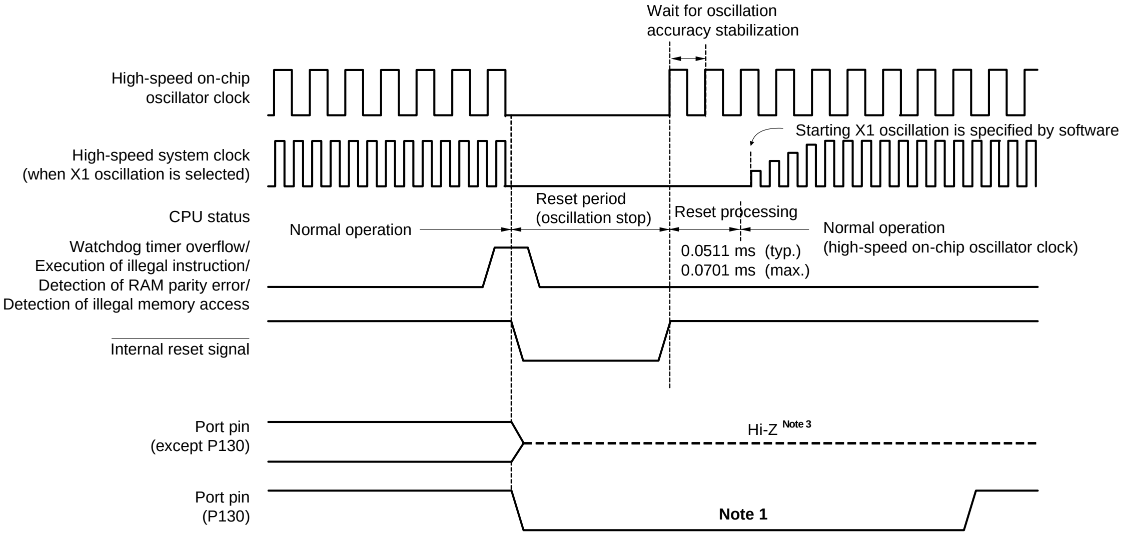
Figure 30-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Release from the reset state is automatic 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.

(**Notes** and **Caution** are listed on the next page.)

Figure 30-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory


Wait for oscillation
accuracy stabilization
High-speed on-chip oscillator clock
High-speed system clock (when X1 oscillation is selected)
Starting X1 oscillation is specified by software
CPU status
Normal operation
Reset period (oscillation stop)
Reset processing
Normal operation (high-speed on-chip oscillator clock)
0.0511 ms (typ.)
0.0701 ms (max.)
Watchdog timer overflow/ Execution of illegal instruction/ Detection of RAM parity error/ Detection of illegal memory access
Internal reset signal
Port pin (except P130)
Hi-Z Note 3
Port pin (P130)
Note 1

Notes 1. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

2. Reset times (times for release from the external reset state)

After the first release of the POR:	0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
	0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.
After the second release of the POR:	0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
	0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

3. P40 becomes the following state.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).

Reset by POR and LVD circuit supply voltage detection is automatically released when internal $V_{DD} \geq V_{POR}$ or internal $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

For details, see **CHAPTER 31 POWER-ON-RESET CIRCUIT** or **CHAPTER 32 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

30.2 States of Operation During Reset Periods

Table 30-1 shows the states of operation during reset periods. **Table 30-2** shows the states of the hardware after receiving a reset signal.

Table 30-1. Operation Statuses During Reset Period (1/2)

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_{IM}	Operation stopped.
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
		f_{PLL}	Operation stopped.
	Subsystem clock	f_{XT}	Operation possible when RTC power-on-reset does not occur
		f_{EXS}	Operation possible when RTC power-on-reset does not occur
	f_{IL}		Operation stopped
CPU			
Code flash memory			
Data flash memory			
RAM			
Port (latch)			High impedance[Note]
Timer array unit			Operation stopped
Realtime clock with independent power supply			Operation possible when RTC power-on-reset does not occur
Frequency measurement circuit			Operation stopped
High-speed on-chip oscillator clock frequency correction function			
Oscillation stop detection			
12-bit interval timer			
8-bit interval timer			
Watchdog timer			
Clock output/buzzer output			
Sampling output timer detector			
Timer RJ			
12-bit A/D converter			
24-bit ΔΣ A/D converter			
Temperature sensor 2			
Serial array unit (SAU)			
IrDA			
Serial interface (IICA)			
Serial interface UARTMG			
LCD controller/driver			
Data transfer controller (DTC)			
Event link controller (ELC)			
Power-on-reset function			Detection operation possible
RTC power-on-reset function			

(**Note** and **Remark** are listed on the next page.)

Table 30-1. Operation Statuses During Reset Period (2/2)

Item		During Reset Period
Voltage detection function	V_{DD}	Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
	V_{DD}, LVDVBAT, VRTC, EXLVD pin supply voltage	Operation stopped
External interrupt	INTP0 to INTP9, INTP12 to INTP14	
	RTCIC0 to RTCIC2	
Key interrupt function		
CRC operation function	High-speed CRC	
	General-purpose CRC	
32-bit multiplier and multiply accumulator		
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection function		
AES circuit		
Detection of tampering		

Note P40 and P130 become the following state.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the internal pull-up resistor).
- P130: Low level during the reset period

Remark

f_{IH}:	High-speed on-chip oscillator clock
f_{IM}:	Middle-speed on-chip oscillator clock
f_{X}:	X1 oscillation 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
f_{PLL}:	PLL clock frequency

Table 30-2. Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment[Note]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark For the state of the special function register (SFR) after receiving a reset signal, see **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

30.3 Register for Confirming Reset Source

30.3.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 30-4. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined[Note 1] 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	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

RPERF	Internal reset request by RAM parity
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

IAWRF	Internal reset request by illegal-memory access
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	No internal reset request has been generated, or the RESF register has been cleared.
1	An internal reset request has been generated.

Notes 1. The value after reset varies depending on the reset source. See **Table 30-3**.

2. This reset occurs when instruction code FFH is executed.
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area. Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 34.3.3 RAM parity error detection function.

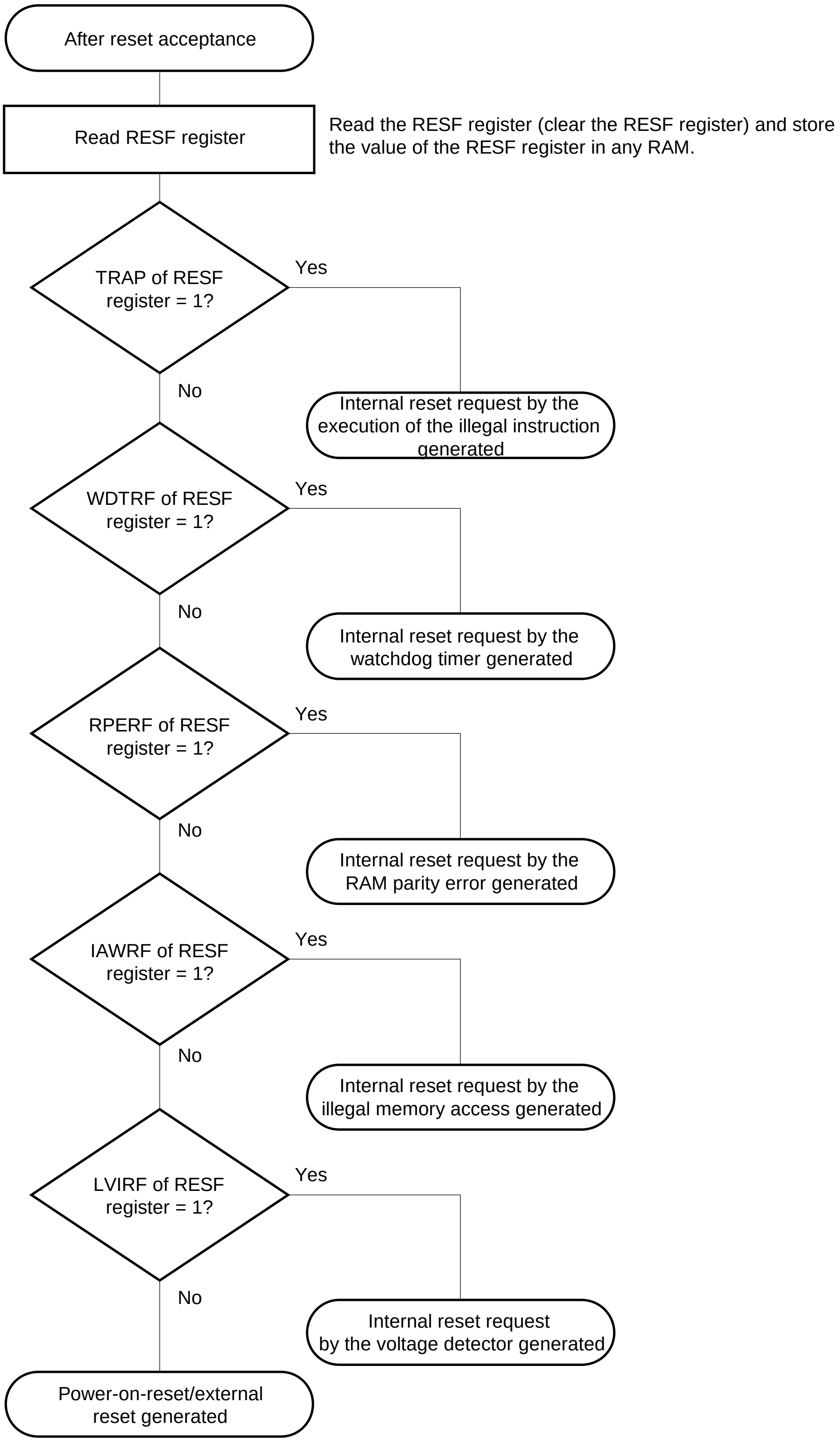
The status of the RESF register when a reset request is generated is shown in **Table 30-3**.

Table 30-3. RESF Register Status 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 LVD
TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF bit			Held	Set (1)			
RPERF bit				Held	Set (1)		
IAWRF bit					Held	Set (1)	
LVIRF bit						Held	Set (1)

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction.
Figure 30-5 shows the procedure for checking a reset source.

Figure 30-5. Example of Procedure for Checking Reset Source


After reset acceptance
Read RESF register
Read the RESF register (clear the RESF register) and store the value of the RESF register in any RAM.
TRAP of RESF register = 1?
Yes
No
Internal reset request by the execution of the illegal instruction generated
WDTRF of RESF register = 1?
Yes
No
Internal reset request by the watchdog timer generated
RPERF of RESF register = 1?
Yes
No
Internal reset request by the RAM parity error generated
IAWRF of RESF register = 1?
Yes
No
Internal reset request by the illegal memory access generated
LVIRF of RESF register = 1?
Yes
No
Internal reset request by the voltage detector generated
Power-on-reset/external reset generated

30.3.2 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 is valid, and writing “0” is ignored.

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 an 8-bit memory manipulation instruction.

Power-on reset signal generation clears this register to 00H.

Cautions 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another factor occurs.

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 30-6. Format of Power-on-Reset Status Register (PORSR)

Address: F00F9H After reset: 00H 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.

30.3.3 RTC power-on-reset status register (RTCPORSR)

The RTCPORSR register is used to check the occurrence of an RTC Power-on reset.

Writing 1 to bit 0 (RTCPORF) of the RTCPORSR register enables this function. Writing 0 disables this function.

Write 1 to the RTCPORF bit in advance to enable checking of the occurrence of an RTC power-on reset.

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

The RTCPORSR register runs on the VRTC power-supply. Immediately after the VRTC power-supply is turned on, use detection of the voltage on the VRTC pin (see **32.3.5 VRTC pin voltage detection control register (LVDVRTC)**) to confirm that the supply of the power to the VRTC pin has started.

Generation of the RTC power-on reset signal clears this register to 00H.

Cautions 1. The RTCPORSR register is reset only by an RTC power-on reset; it retains the value when a reset caused by another source occurs.

2. The RTCPORSR register is readable and writable while the VRTCEN bit is “1”.

Figure 30-7. Format of RTC Power-on-Reset Status Register (RTCPORSR)

Address: F0380H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCPORSR	0	0	0	0	0	0	0	RTCPORF

RTCPORF	Checking occurrence of RTC power-on reset
0	A value 1 has not been written, or an RTC power-on reset has occurred.
1	No RTC power-on reset has occurred.

30.3.4 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 of the reset state of the corresponding on-chip peripheral module.

Figure 30-8. Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IRDARES	ADCRES	IICA0RES	SAU1RES	SAU0RES	SAU2RES	TAU0RES

PRR0n	Control resetting of the on-chip peripheral modules
0	Releases the on-chip peripheral modules from the reset state.
1	The on-chip peripheral modules are in the reset state.

Remark n = 0 to 6

Target modules for each of the bits are listed below.

Table 30-4. Target Modules for Each Bit in PRR0

Bit	Bit Name	Target Module
6	IRDARES	IrDA
5	ADCRES	A/D converter/temperature sensor 2
4	IICA0RES	Serial interface IICA
3	SAU1RES	Serial array unit (unit 1)
2	SAU0RES	Serial array unit (unit 0)
1	SAU2RES	Serial array unit (unit 2)
0	TAU0RES	Timer array unit

30.3.5 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 of the reset state of the corresponding on-chip peripheral module.

Figure 30-9. Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	<2>	<1>	<0>
PRR1	0	0	SMOTD1RES	SMOTD0RES	0	TRJ1RES	TRJ0RES	DSADRES

PRR1n	Control resetting of the on-chip peripheral modules
0	Releases the on-chip peripheral modules from the reset state.
1	The on-chip peripheral modules are in the reset state.

Remark n = 0 to 2, 4, 5

Target modules for each of the bits are listed below.

Table 30-5. Target Modules for Each Bit in PRR1

Bit	Bit Name	Target Module
5	SMOTD1RES	Sampling output timer detector 1
4	SMOTD0RES	Sampling output timer detector 0
2	TRJ1RES	Timer RJ1
1	TRJ0RES	Timer RJ0
0	DSADRES	24-bit ΔΣ A/D converter

30.3.6 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 of the reset state of the corresponding on-chip peripheral module.

Figure 30-10. Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	0
PRR2	TMKARES	OSDCRES	UARTMG1RES	UARTMG0RES	0	MACRES	0	0

PRR2n	Control resetting of the on-chip peripheral modules
0	Releases the on-chip peripheral modules from the reset state.
1	The on-chip peripheral modules are in the reset state.

Remark n = 2, 4 to 7

Target modules for each of the bits are listed below.

Table 30-6. Target Modules for Each Bit in PRR2

Bit	Bit Name	Target Module
7	TMKARES	12-bit interval timer
6	OSDCRES	Oscillation stop detection circuit
5	UARTMG1RES	Serial interface UARTMG1
4	UARTMG0RES	Serial interface UARTMG0
2	MACRES	32-bit multiplier and accumulator

CHAPTER 31 POWER-ON-RESET CIRCUIT

31.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). However, be sure to maintain the reset state until the power supply voltage reaches the operating voltage range specified in **43.4 AC Characteristics**, by using the voltage detector or external reset pin.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detector or externally input reset signal, before the operation voltage falls below the range defined in **43.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) and power-on-reset status register (PORSR) are cleared to 00H.

Remarks 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 **CHAPTER 30 RESET FUNCTION**.

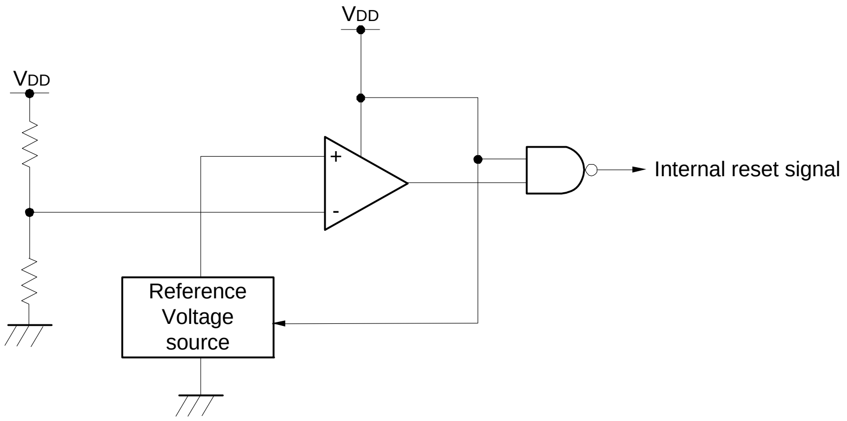
2. Whether an internal reset has been generated by the power-on reset circuit can be checked by using the power-on-reset status register (PORSR). For details of the PORSR register, see **CHAPTER 30 RESET FUNCTION**.

3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **43.6.5 POR circuit characteristics**.

31.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in **Figure 31-1**.

Figure 31-1. Block Diagram of Power-on-reset Circuit

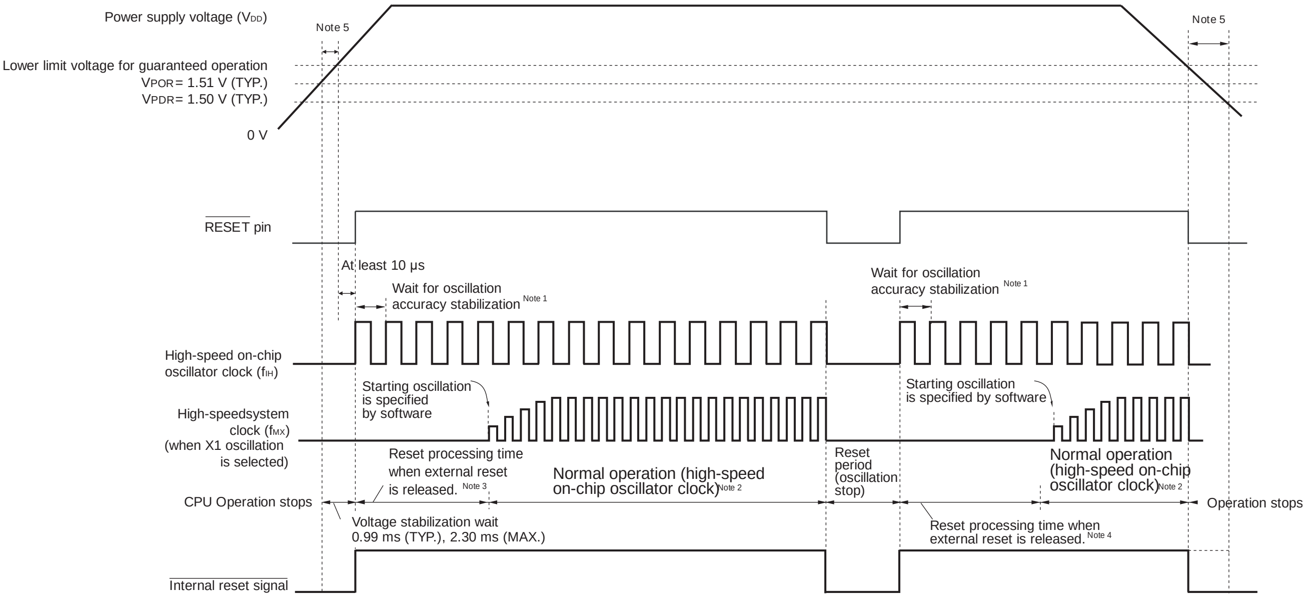


31.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 31-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used


Power supply voltage (VDD)
Note 5
Note 5
Lower limit voltage for guaranteed operation
VPOR = 1.51 V (TYP.)
VPDR = 1.50 V (TYP.)
0 V
RESET pin
At least 10 μs
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speedsystem clock (fMX) (when X1 oscillation is selected)
Reset processing time when external reset is released. Note 3
Normal operation (high-speed on-chip oscillator clock) Note 2
Reset period (oscillation stop)
Normal operation (high-speed on-chip oscillator clock) Note 2
CPU Operation stops
Operation stops
Voltage stabilization wait 0.99 ms (TYP.), 2.30 ms (MAX.)
Reset processing time when external reset is released. Note 4
Internal reset signal

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock, or subsystem clock, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator 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, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator clock, please switch from to check the oscillation stabilization time by using for example the timer function.

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} (1.51 V, typ.) is reached.

With the LVD circuit in use: 0.672 ms (typ.), 0.832 ms (max.)

With the LVD circuit not in use: 0.399 ms (typ.), 0.519 ms (max.)

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: 0.531 ms (typ.), 0.675 ms (max.)

With the LVD circuit not in use: 0.259 ms (typ.), 0.362 ms (max.)

5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **43.4 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 detector or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

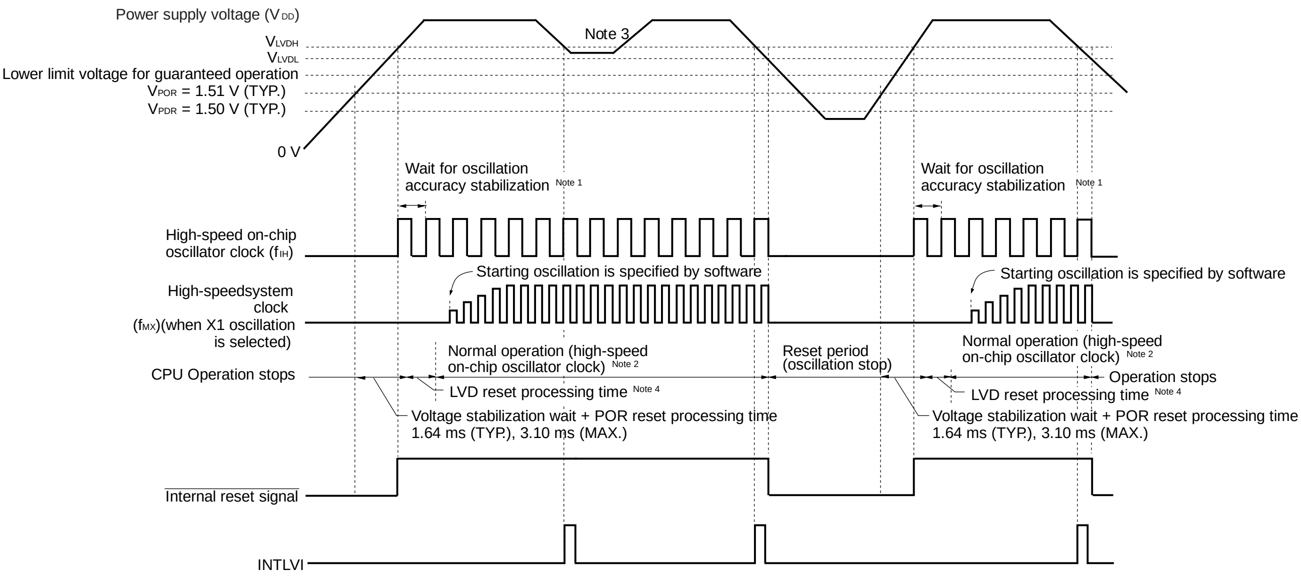
Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 32 VOLTAGE DETECTOR.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 31-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD interrupt & reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 0)


Power supply voltage (V_{DD})
V_{LVDH}
V_{LVDL}
Lower limit voltage for guaranteed operation
V_{POR} = 1.51 V (TYP.)
V_{PDR} = 1.50 V (TYP.)
0 V
Note 3
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (f_{IH})
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speedsystem clock (f_{MX})(when X1 oscillation is selected)
Normal operation (high-speed on-chip oscillator clock) Note 2
Reset period (oscillation stop)
Normal operation (high-speed on-chip oscillator clock) Note 2
CPU Operation stops
Operation stops
LVD reset processing time Note 4
LVD reset processing time Note 4
Voltage stabilization wait + POR reset processing time 1.64 ms (TYP.), 3.10 ms (MAX.)
Voltage stabilization wait + POR reset processing time 1.64 ms (TYP.), 3.10 ms (MAX.)
Internal reset signal
INTLVI

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock, or subsystem clock, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator 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, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator clock, please switch from to check the oscillation stabilization time by using for example the timer function.

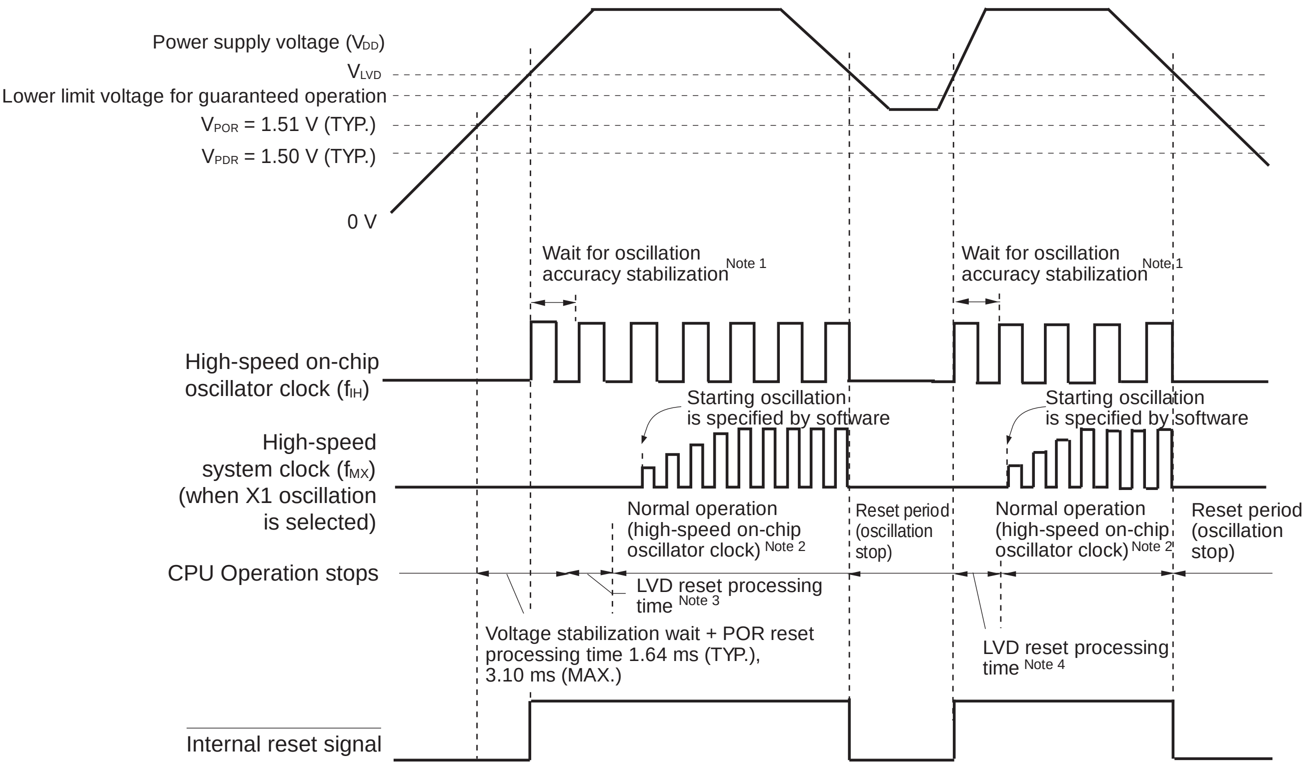
3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 32-20 Setting Procedure for Operating Voltage Check and Reset**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 31-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1 = 1, LVIMDS0 = 1)


Power supply voltage (VDD)
VLVD
Lower limit voltage for guaranteed operation
VPOR = 1.51 V (TYP.)
VPDR = 1.50 V (TYP.)
0 V
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Normal operation (high-speed on-chip oscillator clock) Note 2
Reset period (oscillation stop)
Normal operation (high-speed on-chip oscillator clock) Note 2
Reset period (oscillation stop)
CPU Operation stops
LVD reset processing time Note 3
Voltage stabilization wait + POR reset processing time 1.64 ms (TYP.), 3.10 ms (MAX.)
LVD reset processing time Note 4
Internal reset signal

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock, or subsystem clock, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator 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, or the middle-speed on-chip oscillator clock, or the low-speed on-chip oscillator clock, please switch from to check the oscillation stabilization time by using for example the timer function.

3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

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 (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (typ.), 0.0701 ms (max.)

Remarks 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in **Note 3** of **Figure 31-2 (3)**.

CHAPTER 32 VOLTAGE DETECTOR

32.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The detection voltages can be reset using the LVIS register. The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected as one of 14 levels (For details, see **32.3.2 Voltage detection level register (LVIS)** and **CHAPTER 37 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **43.4 AC Characteristics**. This is done by utilizing the voltage detector 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 detector 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 400C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. This level is also used for generating resets. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets. The detection voltages can be reset using the LVIS register.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating interrupts/reset release. The detection voltages can be reset using the LVIS register.

The reset and internal interrupt signals are generated in each mode as follows.

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Immediately after a reset occurs, the internal reset state of LVD remains until $V_{DD} \geq V_{LVD}$. The internal reset of LVD is cleared when $V_{DD} \geq V_{LVD}$ is detected. After the internal reset of LVD is released, an interrupt request signal (INTLVI) is generated when $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 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 **CHAPTER 30 RESET FUNCTION.**

The RL78/I1C (512 KB) products have voltage detection function for each power supply pin.

While voltage detection function is operating, whether the supply voltage of each pin is more than the detection level can be checked by interruption or reading the voltage detection flag.

- The LVD circuit compares the V_{DD} pin voltage (V_{DD}) with the detection voltage (V_{LVDVDD}), and generates a one-shot interrupt request signal (INTLVDVDD) by detecting $V_{DD} > V_{LVDVDD}$ or $V_{DD} < V_{LVDVDD}$.

- The LVD circuit compares the LVDVBAT pin voltage (LVDVBAT) with the detection voltage ($V_{LVDVBAT}$), and generates a one-shot interrupt request signal (INTLVDVBAT) by detecting LVDVBAT > $V_{LVDVBAT}$ or LVDVBAT < $V_{LVDVBAT}$.

- The LVD circuit compares the VRTC pin voltage (VRTC) with the detection voltage ($V_{LVDVRTC}$), and generates a one-shot interrupt request signal (INTLVDVRTC) by detecting VRTC > $V_{LVDVRTC}$ or VRTC < $V_{LVDVRTC}$.

- The LVD circuit compares the EXLVD pin voltage (EXLVD) with the detection voltage ($V_{LVDEXLVD}$), and generates a one-shot interrupt request signal (INTLVDEXLVD) by detecting EXLVD > $V_{LVDEXLVD}$ or EXLVD < $V_{LVDEXLVD}$.

32.2 Configuration of Voltage Detector

The block diagrams of the voltage detector (LVD) are shown in **Figure 32-1** to **Figure 32-5**.

Figure 32-1. Block Diagram of Voltage Detector (LVD)

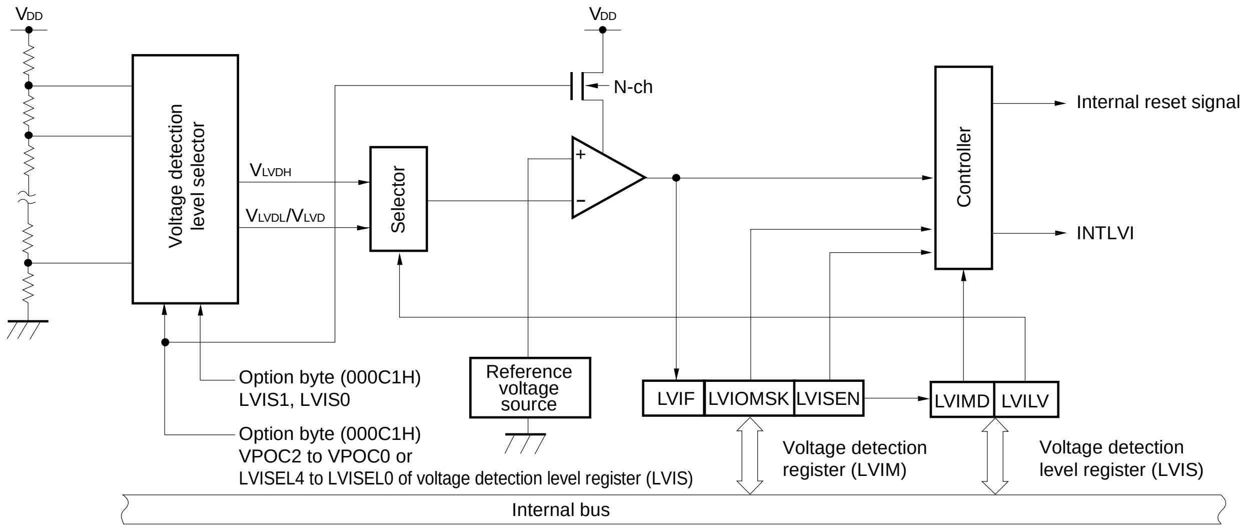


Figure 32-2. Block Diagram of V_{DD} Pin Voltage Detector

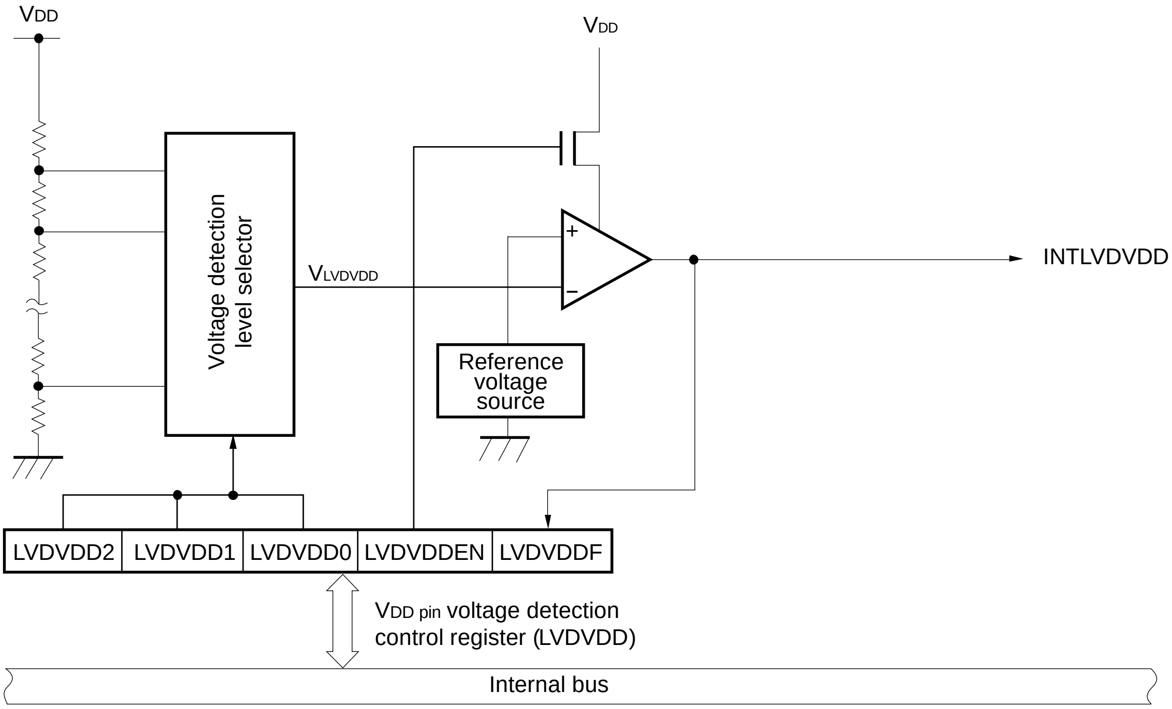


Figure 32-3. Block Diagram of LVDVBAT Pin Voltage Detector

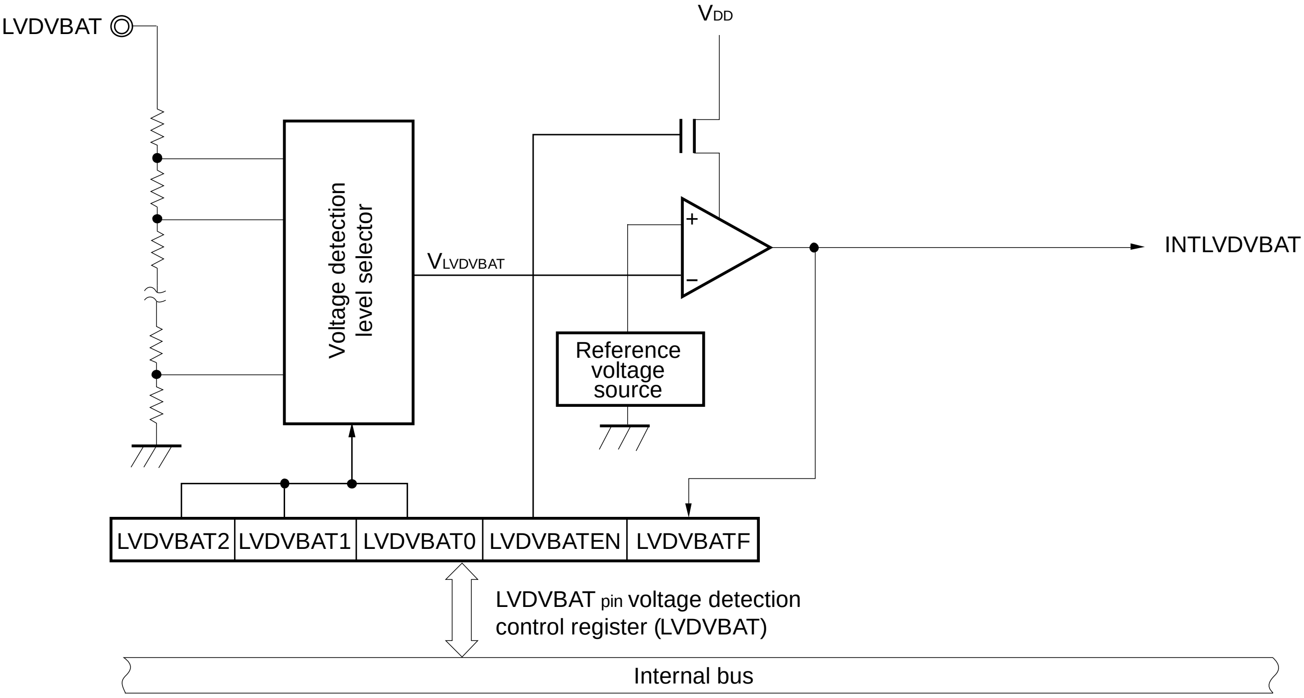


Figure 32-4. Block Diagram of VRTC Pin Voltage Detector

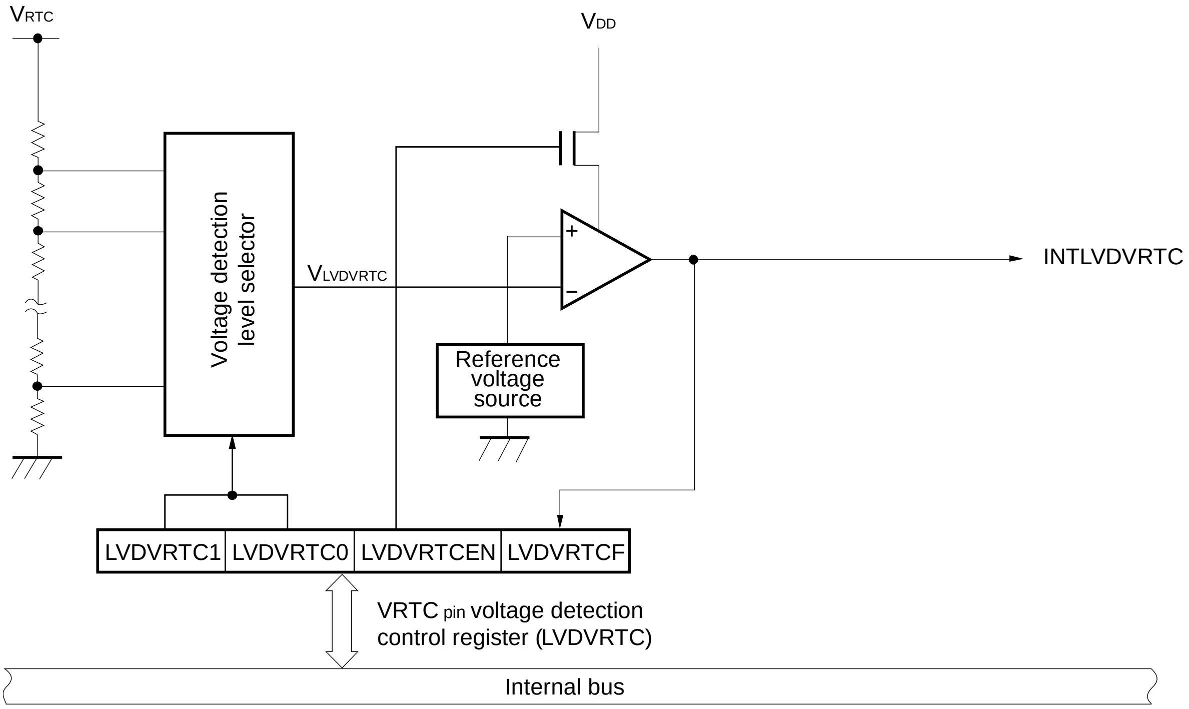
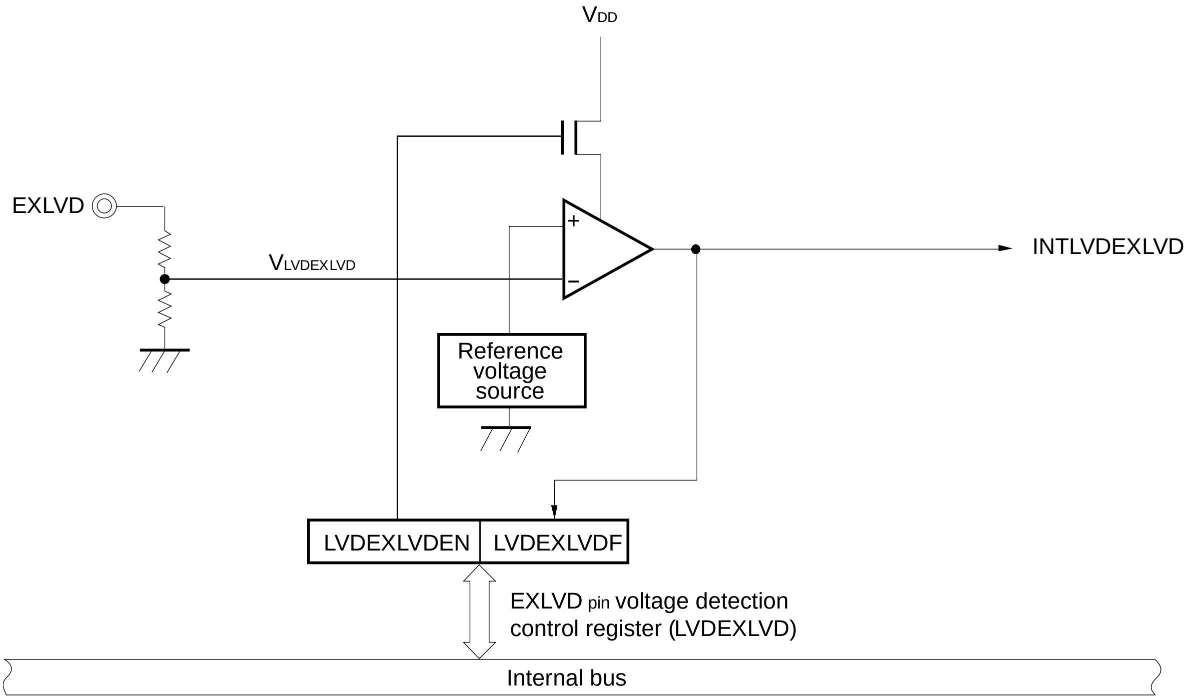


Figure 32-5. Block Diagram of EXLVD Pin Voltage Detector



32.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)
- Voltage detection control register for V_{DD} pin (LVDVDD)
- Voltage detection control register for LVDVBAT pin (LVDVBAT)
- Voltage detection control register for VRTC pin (LVDVRTC)
- Voltage detection control register for EXLVD pin (LVDEXLVD)

32.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

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

Reset signal generation clears this register to 00H.

Figure 32-6. Format of Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN	0	0	0	0	0	LVIOMSK	LVIF

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid [Note 3]

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

2. Bits 0 and 1 are read-only.

3. The LVIOMSK bit is automatically set to 1 for the following periods and generation of an LVD reset or interrupt is masked.

- Period when LVISEN = 1
  In either of the following cases, generation of an LVD reset or interrupt is masked only in interrupt & reset mode.
- Wait time until the LVD detection voltage stabilizes after an LVD interrupt is generated
- Wait time until the LVD detection voltage stabilizes after the value of the LVILV bit is changed

32.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level. The minimum supply voltage (LVD detection voltage) and LVD detection level settings that are set by the user option byte can be changed by software.

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

Reset signal generation input sets this register to Note 1.

Caution Do not change the detection voltage in interrupt & reset mode.

Figure 32-7. Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH   After reset: **Note 1**   R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD[Note 2]	0	LVISEL4[Note 6]	LVISEL3	LVISEL2	LVISEL1	LVISEL0	LVILV[Note 2]

LVIMD[Note 2]	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVISEL4[Note 6]	LVISEL3	LVISEL2	Minimum operating voltage (typical falling value)[Note 5]
0	0	0	1.63 V
0	0	1	1.84 V
0	1	0	2.45 V
0	1	1	2.75 V
1	1	1	1.53 V (LVD OFF)
Other than above			Setting prohibited

LVISEL1	LVISEL0	LVD detection level setting[Note 5]
0	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 1.2 V[Note 3]
0	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.2 V[Note 3]
1	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.1 V[Note 3]
1	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2[Note 4]

LVILV[Note 2]	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Notes 1. The reset value changes depending on the setting of the option byte.
After a reset is released, the values of VPOC2 to VPOC0 and LVIS1 and LVIS0 in the user option byte are reflected in LVISEL4 to LVISEL2, LVISEL1, and LVISEL0, respectively.
The reset values of LVIMD and LVIVL are set as follows.
When LVIMDS1, LVIMDS0 in the option byte = 1, 0: LVIMD = 0, LVILV = 0
When LVIMDS1, LVIMDS0 in the option byte = 1, 1: LVIMD = 1, LVILV = 1
When LVIMDS1, LVIMDS0 in the option byte = 0, 1: LVIMD = 0, LVILV = 1

2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

3. Indicates an approximate detection value. For details on the actual detection voltage, refer to the LVD section in Electrical Specifications.

4. Cannot be selected when LVIMDS1 and LVIMDS0 = 1 and 0.

5. When changing LVISEL4 to LVISEL0 to use two or more LVD detection voltages, the setting value that indicates the highest voltage value among the LVD detection voltages to be used should be set in the VPOC2 to VPOC0 bits and LVIS1 and LVIS0 bits before using the voltages.

6. Rewriting LVISEL4 is prohibited. Keep the initial value unchanged.

Cautions 1. When rewriting the LVIMD and LVILV bits, use the procedure shown in Figure 32-20.

2. Specify the LVD operation mode and initial detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 32-8 shows the format of the user option byte (000C1H/400C1H). For details about the option byte, see CHAPTER 37 OPTION BYTE.

Figure 32-8. Format of User Option Byte (000C1H/400C1H) (1/2)

Address: 000C1H/400C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Note When bank swapping is to be used, set the same value at 400C1H as that at 000C1H, because switching between 000C1H and 400C1H will proceed.

Remarks 1. For details on the LVD circuit, see **32.2 Configuration of Voltage Detector**.

2. The detection voltage is a TYP. value. For details, see **43.6.6 LVD circuit characteristics**.

(**Cautions** are listed on the next page.)

Figure 32-8. Format of User Option Byte (000C1H/400C1H) (2/2)

Address: 000C1H/400C1H$^{\text{Note}}$

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

● LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

● LVD off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note When bank swapping is to be used, set the same value at 400C1H as that at 000C1H, because switching between 000C1H and 400C1H will proceed.

Cautions 1. Set bit 4 to 1.

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 43.4 AC Characteristics. This is done by utilizing the voltage detector 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 detector 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 400C2H).

Remarks 1. ×: Don’t care

2. For details on the LVD circuit, see **32.2 Configuration of Voltage Detector**.

3. The detection voltage is a TYP. value. For details, see **43.6.6 LVD circuit characteristics**.

32.3.3 V_{DD} pin voltage detection control register (LVDVDD)

This register is used to enable/disable V_{DD} pin voltage detection and select the detection voltage.

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

Reset signal generation input sets this register to 00H.

Figure 32-9. Format of Voltage Detection Control Register for V_{DD} Pin (LVDVDD)

Address: F0332H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	3	2	1	0
LVDVDD	LVDVDDEN	LVDVDDF	0	0	0	LVDVDD2	LVDVDD1	LVDVDD0

LVDVDDEN	V_{DD} pin voltage detection enable/disable
0	Disables detection[Note 3]
1	Enables detection

LVDVDDF[Note 2]	V_{DD} pin voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVDVDD}), or detection is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVDVDD})

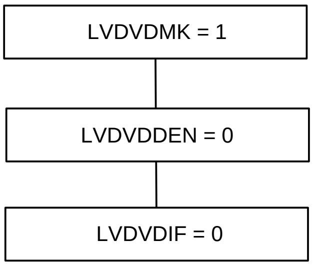
LVDVDD2	LVDVDD1	LVDVDD0	Detection voltage (V_{LVDVDD})	
			Rising edge	Falling edge
0	0	0	2.53 V	2.46 V
0	0	1	2.74 V	2.67 V
0	1	0	2.94 V	2.87 V
0	1	1	3.15 V	3.08 V
1	0	0	3.46 V	3.39 V
1	0	1	3.77 V	3.70 V
Other than above			Setting prohibited	

Notes 1. Bit 6 is read only.

2. When the LVDVDDEN bit is set to 1 while voltage of V_{DD} pin < detection voltage (V_{LVDVDD}), the LVDVDDF bit is undefined until the stabilization time (300 μs) elapses.

3. To disable INTLVDVDD generation, do the following steps.

Figure 32-10. Setting Procedure to Disable V_{DD} Pin Voltage Detection Function



Sets the interrupt mask flag to 1 to disable interrupt acknowledgement.

Sets the LVDVDDEN bit to 0 to stop voltage detection function operation.

Clears the interrupt request flag.

32.3.4 LVDVBAT pin voltage detection control register (LVDVBAT)

This register is used to enable/disable LVDVBAT pin voltage detection and select the detection voltage.

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

Reset signal generation input sets this register to 00H.

Figure 32-11. Format of Voltage Detection Control Register for LVDVBAT Pin (LVDVBAT)

Address: F0333H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	3	2	1	0
LVDVBAT	LVDVBATEN	LVDVBATF	0	0	0	LVDVBAT2	LVDVBAT1	LVDVBAT0

LVDVBATEN	LVDVBAT pin voltage detection enable/disable
0	Disables detection[Note 3]
1	Enables detection

LVDVBATF[Note 2]	LVDVBAT pin voltage detection flag
0	Supply voltage (LVDVBAT) ≥ detection voltage ($V_{LVDVBAT}$), or detection is off
1	Supply voltage (LVDVBAT) < detection voltage ($V_{LVDVBAT}$)

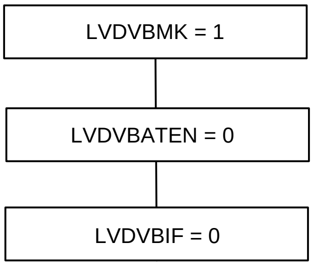
LVDVBAT2	LVDVBAT1	LVDVBAT0	Detection voltage ($V_{LVDVBAT}$)	
			Rising edge	Falling edge
0	0	0	2.23 V	2.17 V
0	0	1	2.43 V	2.37 V
0	1	0	2.63 V	2.57 V
0	1	1	2.73 V	2.67 V
1	0	0	2.83 V	2.77 V
1	0	1	2.93 V	2.87 V
1	1	0	3.13 V	3.07 V
Other than above			Setting prohibited	

Notes 1. Bit 6 is read only.

2. When the LVDVBATEN bit is set to 1 while voltage of LVDVBAT pin < detection voltage ($V_{LVDVBAT}$), the LVDVBATF bit is undefined until the stabilization time (500 μs) elapses.

3. To disable INTLVDBAT generation, do the following steps.

Figure 32-12. Setting Procedure to Disable LVDVBAT Pin Voltage Detection Function



Sets the interrupt mask flag to 1 to disable interrupt acknowledgement.

Sets the LVDVBATEN bit to 0 to stop voltage detection function operation.

Clears the interrupt request flag.

32.3.5 VRTC pin voltage detection control register (LVDVRTC)

This register is used to enable/disable VRTC pin voltage detection and select the detection voltage.

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

Reset signal generation input sets this register to 00H.

Figure 32-13. Format of Voltage Detection Control Register for VRTC Pin (LVDVRTC)

Address: F0334H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	3	2	1	0
LVDVRTC	LVDVRTCEN	LVDVRTCF	0	0	0	0	LVDVRTC1	LVDVRTC0

LVDVRTCEN	VRTC pin voltage detection enable/disable
0	Disables detection[Note 3]
1	Enables detection

LVDVRTCF[Note 2]	VRTC pin voltage detection flag
0	Supply voltage (VRTC) ≥ detection voltage ($V_{LVDVRTC}$), or detection is off
1	Supply voltage (VRTC) < detection voltage ($V_{LVDVRTC}$)

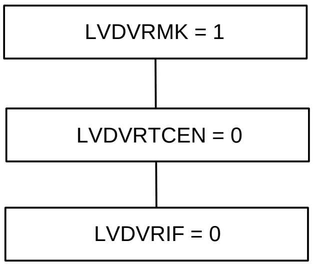
LVDVRTC1	LVDVRTC0	Detection voltage ($V_{LVDVRTC}$)	
		Rising edge	Falling edge
0	0	2.22 V	2.16 V
0	1	2.43 V	2.37 V
1	0	2.63 V	2.57 V
1	1	2.84 V	2.78 V

Notes 1. Bit 6 is read only.

2. When the LVDVRTCEN bit is set to 1 while voltage of VRTC pin < detection voltage ($V_{LVDVRTC}$), the LVDVRTCF bit is undefined until the stabilization time (300 µs) elapses.

3. To disable INTLVDVRTC generation, do the following steps.

Figure 32-14. Setting Procedure to Disable VRTC Pin Voltage Detection Function



Sets the interrupt mask flag to 1 to disable interrupt acknowledgement.

Sets the LVDVRTCEN bit to 0 to stop voltage detection function operation.

Clears the interrupt request flag.

32.3.6 EXLVD pin voltage detection control register (LVDEXLVD)

This register is used to enable/disable EXLVD pin voltage detection and select the detection voltage.

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

Reset signal generation input sets this register to 00H.

Figure 32-15. Format of Voltage Detection Control for EXLVD Pin Register (LVDEXLVD)

Address: F0335H After reset: 00H R/W[Note 1]

Symbol	<7>	<6>	5	4	3	2	1	0
LVDEXLVD	LVDEXLVDEN	LVDEXLVDF	0	0	0	0	0	0

LVDEXLVDEN	EXLVD pin voltage detection enable/disable
0	Disables detection[Note 3]
1	Enables detection

LVDEXLVDF[Note 2]	EXLVD pin voltage detection flag
0	Supply voltage (EXLVD) ≥ detection voltage ($V_{LVDEXLVD}$), or detection is off
1	Supply voltage (EXLVD) < detection voltage ($V_{LVDEXLVD}$)

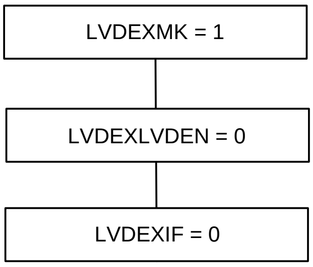
Detection voltage ($V_{LVDEXLVD}$)	
Rising edge	Falling edge
1.33 V (fixed)	1.28 V (fixed)

Notes 1. Bit 6 is read only.

2. When the LVDEXLVDEN bit is set to 1 while voltage of EXLVD pin < detection voltage ($V_{LVDEXLVD}$), the LVDEXLVDF bit is undefined until the stabilization time (300 μs) elapses.

3. To disable INTLVDEXLVD generation, do the following steps.

Figure 32-16. Setting Procedure to Disable EXLVD Pin Voltage Detection Function



Sets the interrupt mask flag to 1 to disable interrupt acknowledgement.

Sets the LVDEXLVDEN bit to 0 to stop voltage detection function operation.

Clears the interrupt request flag.

32.4 Operation of Voltage Detector

32.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  See **32.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

● Operation in LVD reset mode

In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

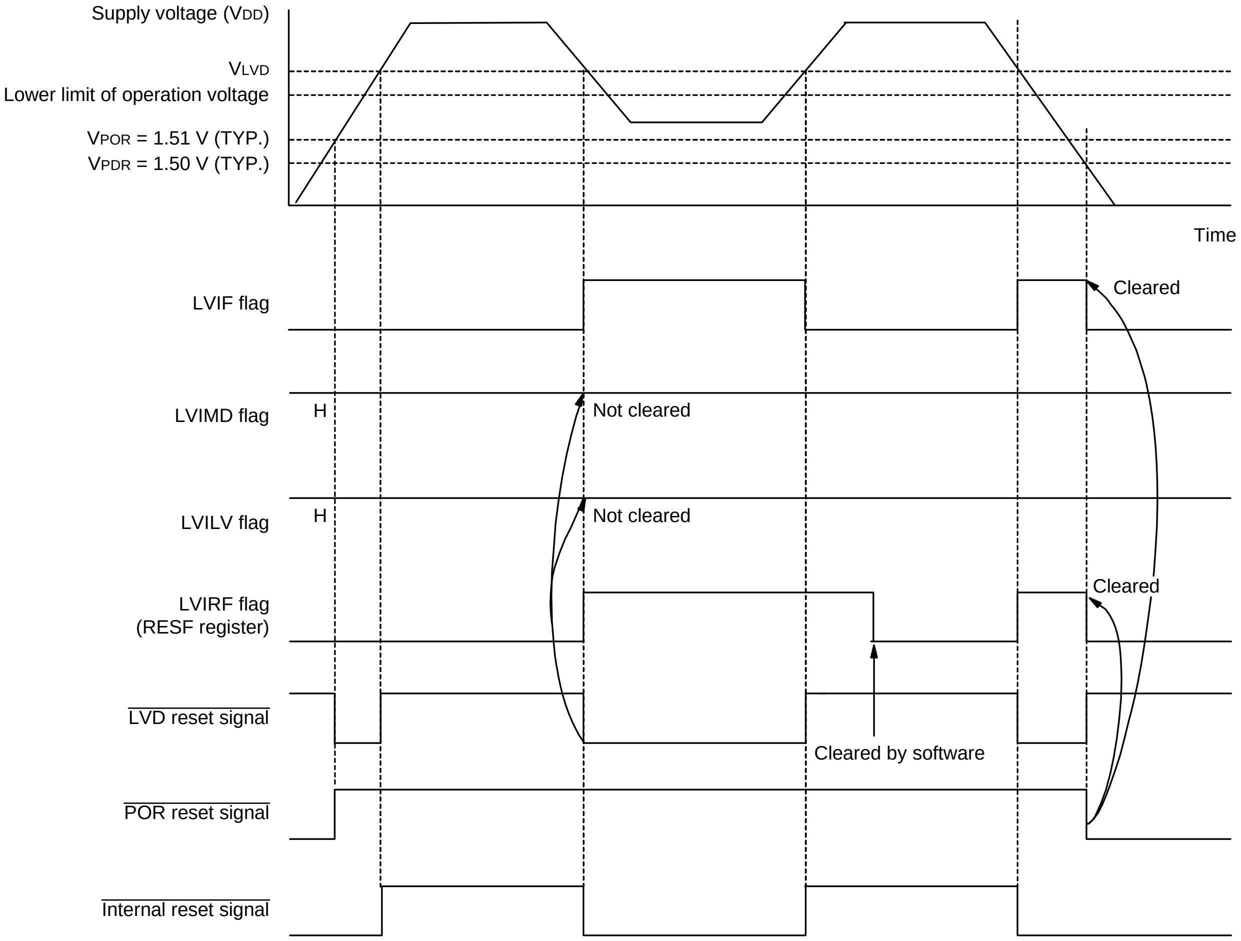
At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

The reset release voltage when an LVD reset is generated is the detection voltage set by the option byte or detection voltage set by the LVIS register, whichever is higher. The state of an internal reset by the LVD is retained until the supply voltage exceeds the voltage detection level.

The reset release voltage used for resets other than an LVD reset is the same voltage detection level set by the option byte.

Figure 32-17 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 32-17. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

32.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **32.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

● Operation in LVD interrupt mode

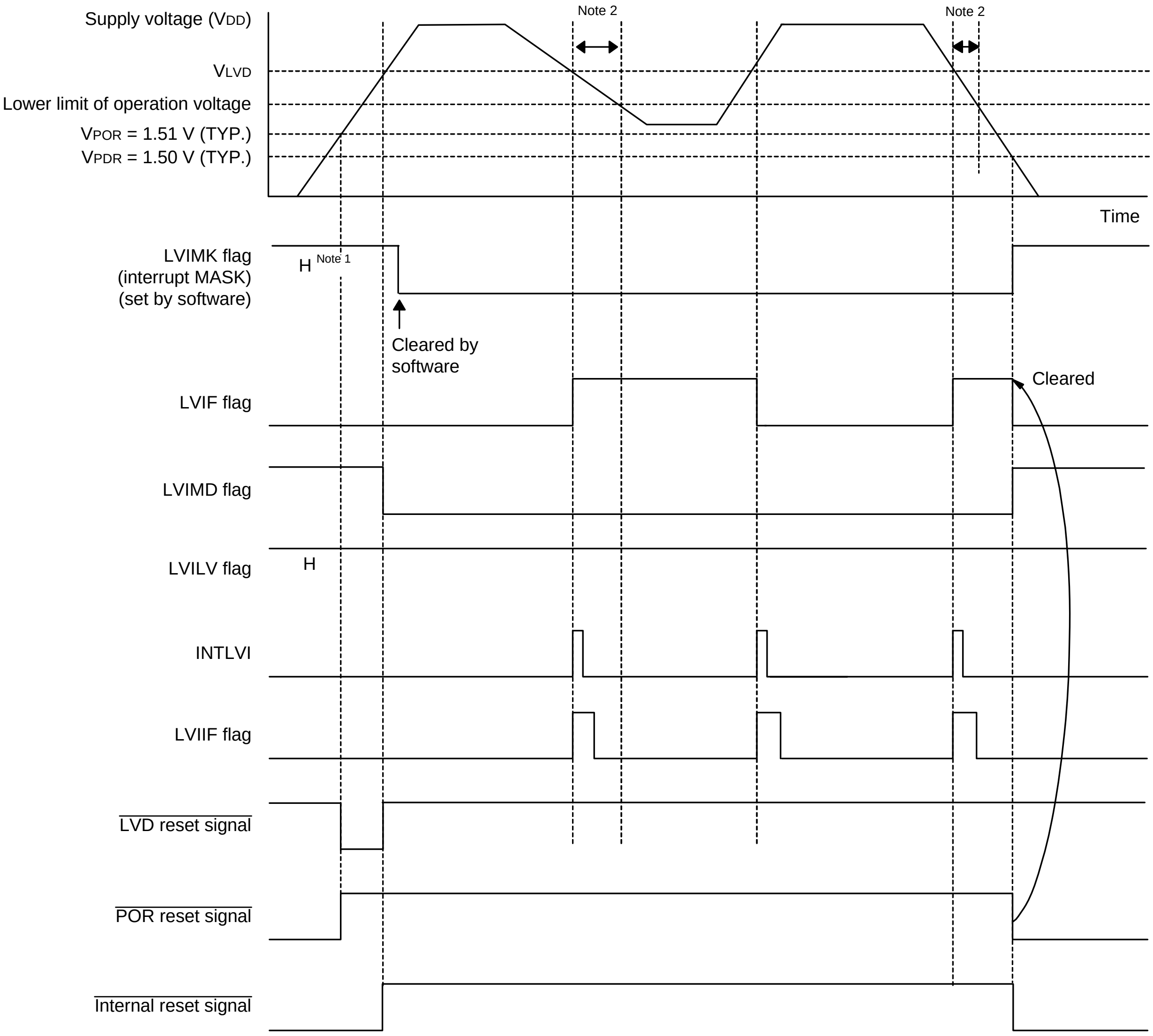
In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) immediately after a reset occurs. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

After the internal reset of LVD is released, an interrupt request signal by LVD (INTLVD) is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). 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 **43.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

After the LVISEN bit is set to 1 (LVD is masked) by changing the detection level, if the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}) when LVISEN is set to 0, an interrupt request signal by the LVD (INTLVI) is generated.

Figure 32-18 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 32-18. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

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 **43.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

32.4.3 When used as interrupt and reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H. Do not manipulate the detection voltage using the LVIS register.

The operation is started in the following initial setting state when the interrupt & reset mode is set.

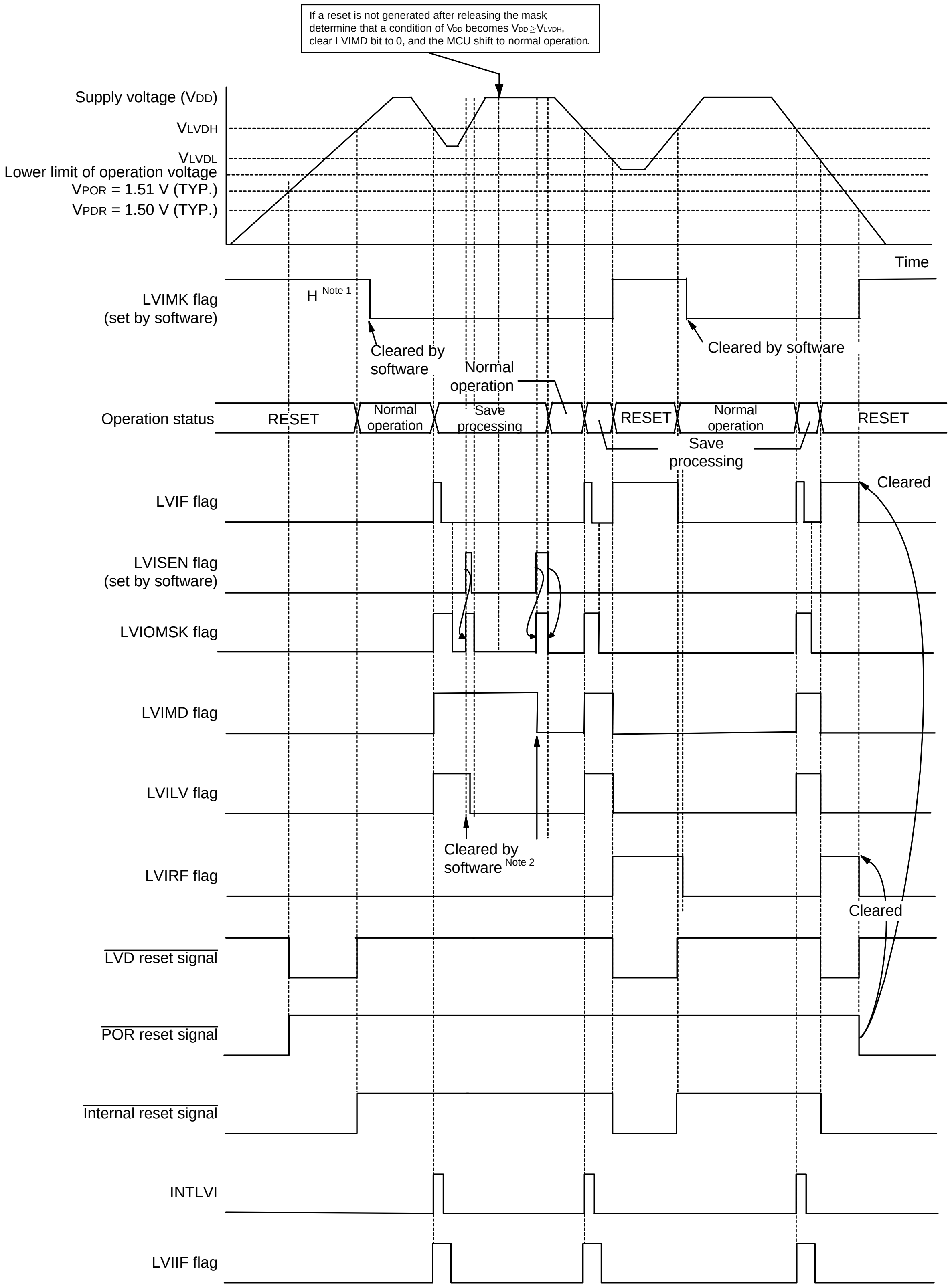
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **32.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

● Operation in LVD interrupt & reset mode

In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}). An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}). To use the LVD reset & interrupt mode, perform the processing according to **Figure 32-20 Setting Procedure for Operating Voltage Check and Reset**.

Figure 32-19 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 32-19. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



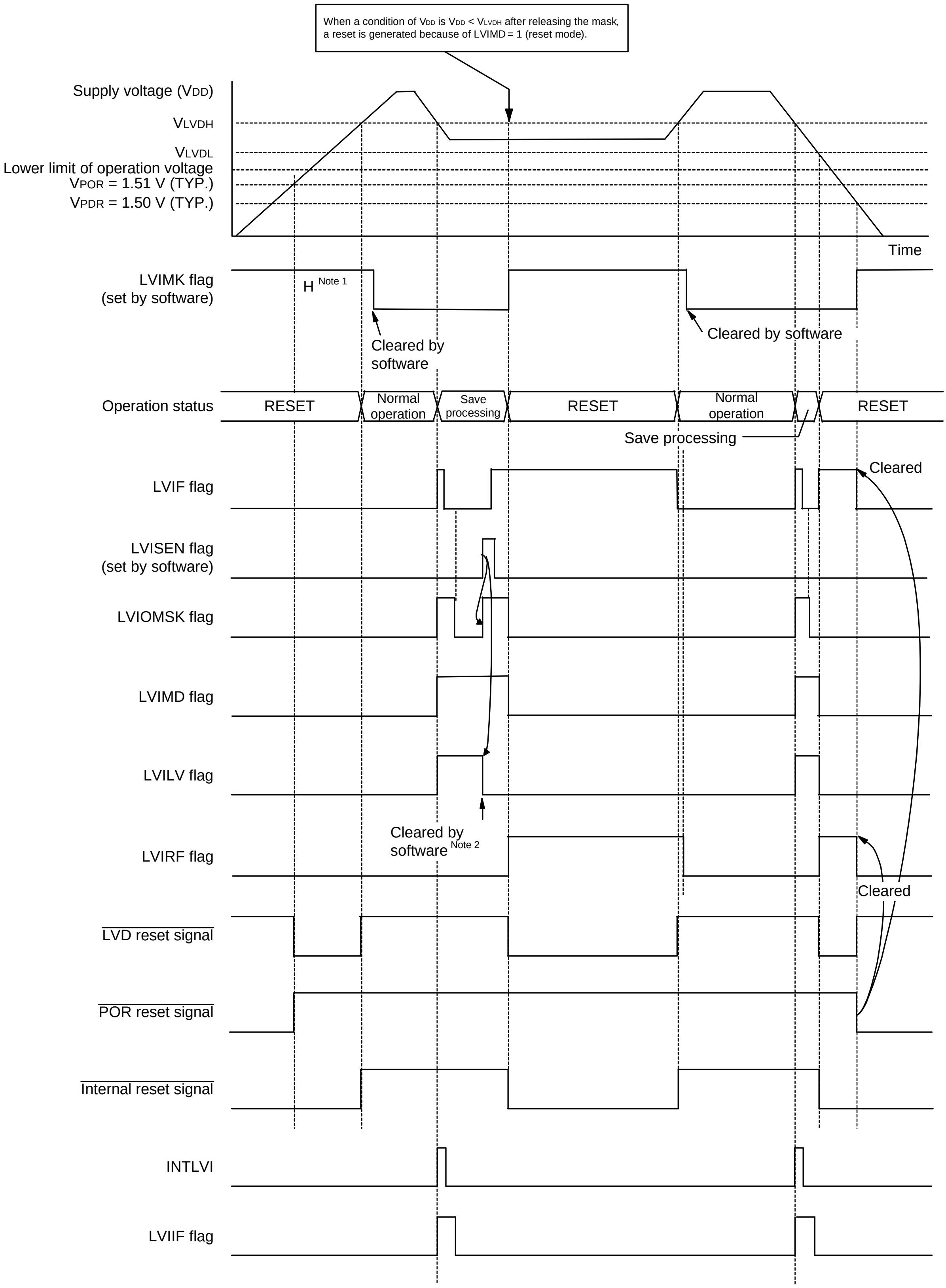
(**Notes** and **Remark** are listed on the next page.)

Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 32-20 Setting Procedure for Operating Voltage Check and Reset** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{POR}: POR power supply fall detection voltage

Figure 32-19. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



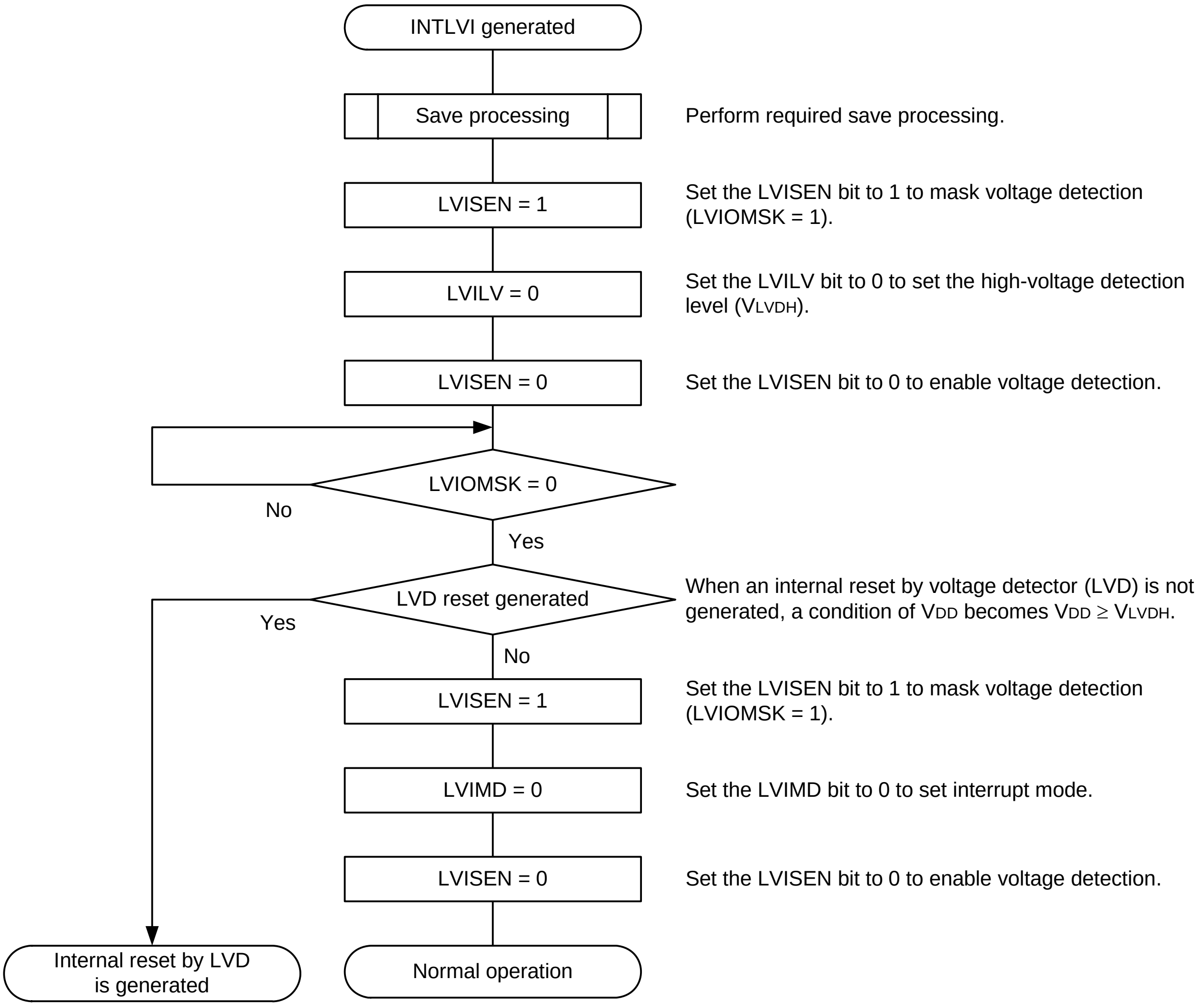
(**Notes** and **Remark** are listed on the next page.)

Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 32-20 Setting Procedure for Operating Voltage Check and Reset** in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 32-20. Setting Procedure for Operating Voltage Check and Reset

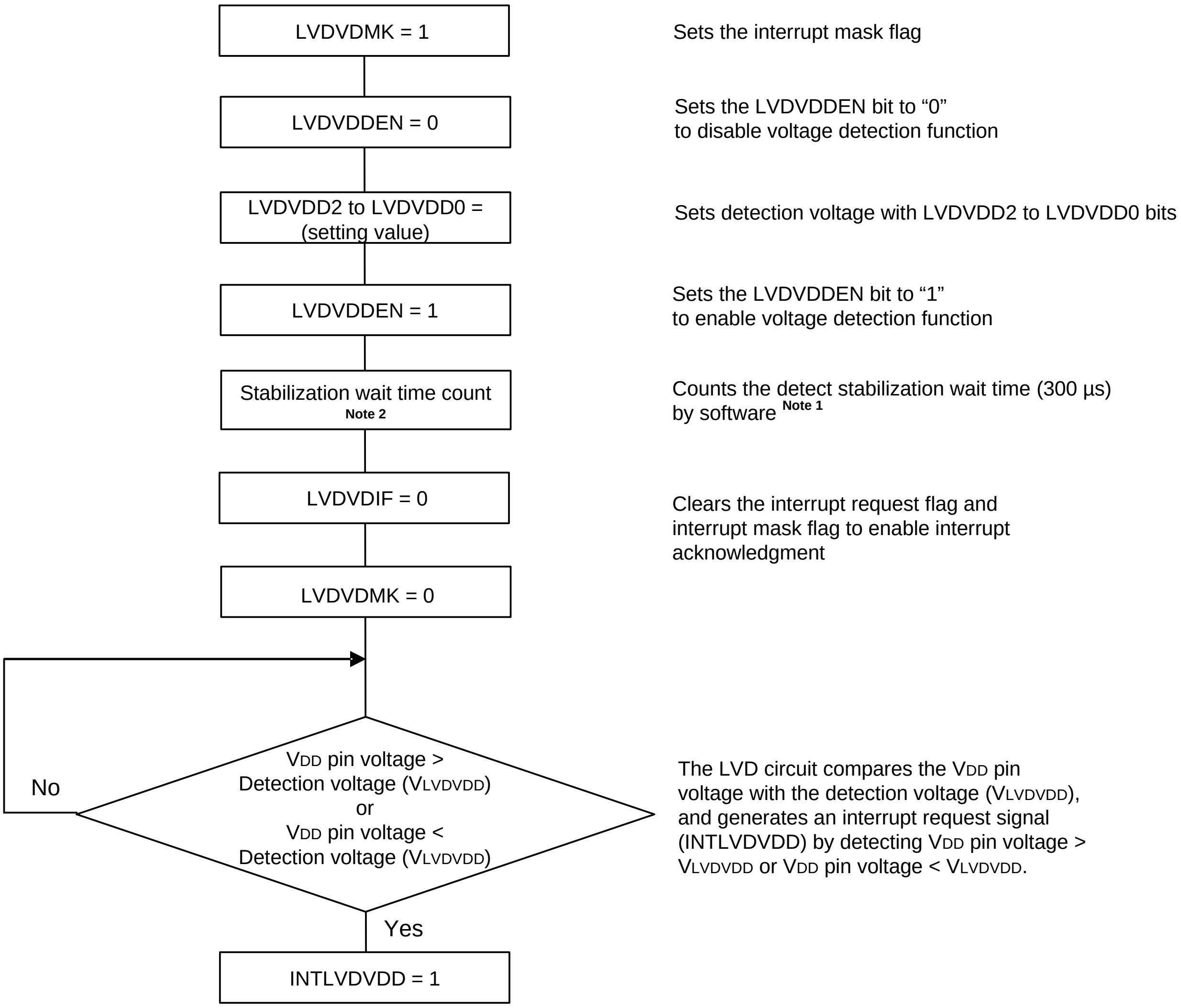

INTLVI generated
Save processing
Perform required save processing.
LVISEN = 1
Set the LVISEN bit to 1 to mask voltage detection (LVIOMSK = 1).
LVILV = 0
Set the LVILV bit to 0 to set the high-voltage detection level (V_{LVDH}).
LVISEN = 0
Set the LVISEN bit to 0 to enable voltage detection.
LVIOMSK = 0
No
Yes
LVD reset generated
When an internal reset by voltage detector (LVD) is not generated, a condition of V_{DD} becomes $V_{DD} \geq V_{LVDH}$.
Yes
No
LVISEN = 1
Set the LVISEN bit to 1 to mask voltage detection (LVIOMSK = 1).
LVIMD = 0
Set the LVIMD bit to 0 to set interrupt mode.
LVISEN = 0
Set the LVISEN bit to 0 to enable voltage detection.
Internal reset by LVD is generated
Normal operation

32.4.4 Each power supply pin voltage detection setting procedure

(1) V_{DD}

The setting procedure of V_{DD} pin voltage detection is shown below.

Figure 32-21. Setting Procedure of V_{DD} Pin Voltage Detection


LVDVDMK = 1
Sets the interrupt mask flag
LVDVDDEN = 0
Sets the LVDVDDEN bit to "0"
to disable voltage detection function
LVDVDD2 to LVDVDD0 =
(setting value)
Sets detection voltage with LVDVDD2 to LVDVDD0 bits
LVDVDDEN = 1
Sets the LVDVDDEN bit to "1"
to enable voltage detection function
Stabilization wait time count Note 2
Counts the detect stabilization wait time (300 μs)
by software Note 1
LVDVDIF = 0
Clears the interrupt request flag and
interrupt mask flag to enable interrupt
acknowledgment
LVDVDMK = 0
VDD pin voltage >
Detection voltage (VLVDVDD)
or
VDD pin voltage <
Detection voltage (VLVDVDD)
No
Yes
The LVD circuit compares the VDD pin
voltage with the detection voltage (VLVDVDD),
and generates an interrupt request signal
(INTLVDVDD) by detecting VDD pin voltage >
VLVDVDD or VDD pin voltage < VLVDVDD.
INTLVDVDD = 1

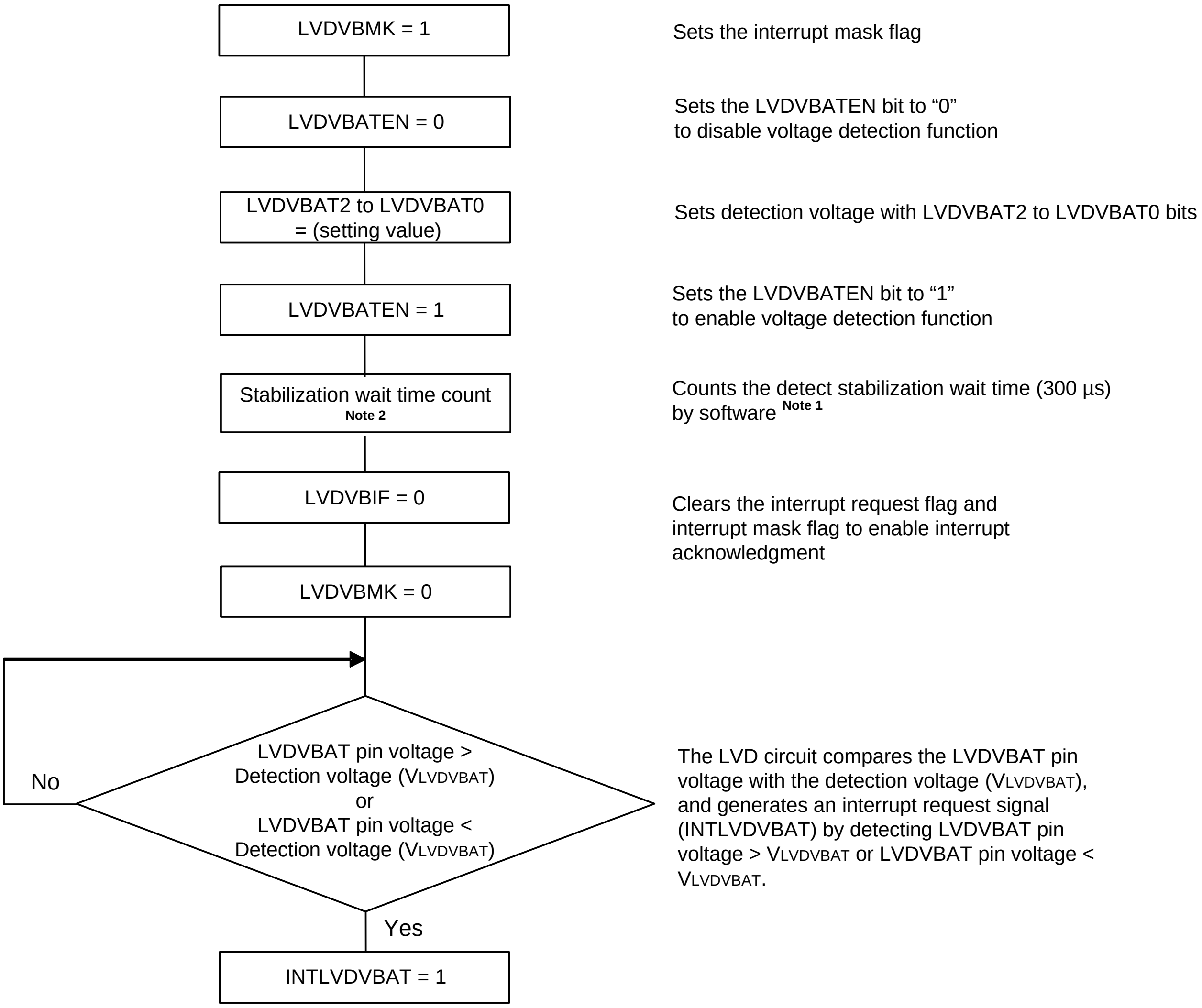
Notes 1. Be sure to set the detection voltage level when pin voltage < detection voltage.

2. If other process operating secures the stabilization wait time after setting the LVDVDDEN bit to 1 to enable the voltage detect function operation, the count process isn't required.

(2) LVDVBAT

The setting procedure of LVDVBAT pin voltage detection is shown below.

Figure 32-22. Setting Procedure of LVDVBAT Pin Voltage Detection


LVDVBMK = 1
Sets the interrupt mask flag
LVDVBATEN = 0
Sets the LVDVBATEN bit to "0" to disable voltage detection function
LVDVBAT2 to LVDVBAT0 = (setting value)
Sets detection voltage with LVDVBAT2 to LVDVBAT0 bits
LVDVBATEN = 1
Sets the LVDVBATEN bit to "1" to enable voltage detection function
Stabilization wait time count Note 2
Counts the detect stabilization wait time (300 µs) by software Note 1
LVDVBIF = 0
Clears the interrupt request flag and interrupt mask flag to enable interrupt acknowledgment
LVDVBMK = 0
LVDVBAT pin voltage > Detection voltage (VLVDVBAT) or LVDVBAT pin voltage < Detection voltage (VLVDVBAT)
No
Yes
The LVD circuit compares the LVDVBAT pin voltage with the detection voltage (VLVDVBAT), and generates an interrupt request signal (INTLVDVBAT) by detecting LVDVBAT pin voltage > VLVDVBAT or LVDVBAT pin voltage < VLVDVBAT.
INTLVDVBAT = 1

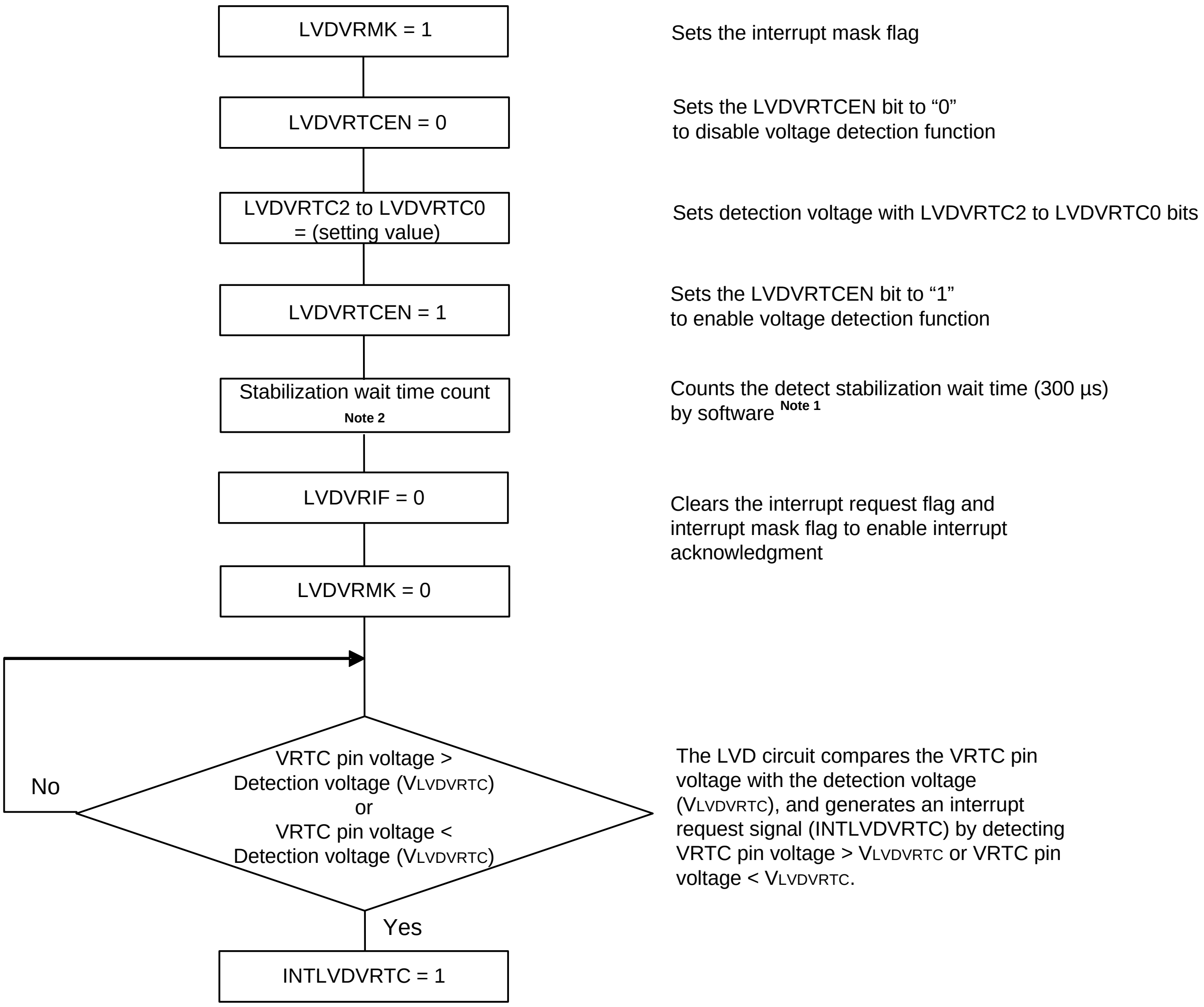
Notes 1. Be sure to set the detection voltage level when pin voltage < detection voltage.

2. If other process operating secures the stabilization wait time after setting the LVDVBATEN bit to 1 to enable the voltage detect function operation, the count process isn't required.

(3) VRTC

The setting procedure of VRTC pin voltage detection is shown below.

Figure 32-23. Setting Procedure of VRTC Pin Voltage Detection


LVDVRMK = 1
Sets the interrupt mask flag
LVDVRTCEN = 0
Sets the LVDVRTCEN bit to “0” to disable voltage detection function
LVDVRTC2 to LVDVRTC0 = (setting value)
Sets detection voltage with LVDVRTC2 to LVDVRTC0 bits
LVDVRTCEN = 1
Sets the LVDVRTCEN bit to “1” to enable voltage detection function
Stabilization wait time count Note 2
Counts the detect stabilization wait time (300 μs) by software Note 1
LVDVRIF = 0
Clears the interrupt request flag and interrupt mask flag to enable interrupt acknowledgment
LVDVRMK = 0
VRTC pin voltage > Detection voltage (VLVDVRTC) or VRTC pin voltage < Detection voltage (VLVDVRTC)
No
Yes
The LVD circuit compares the VRTC pin voltage with the detection voltage (VLVDVRTC), and generates an interrupt request signal (INTLVDVRTC) by detecting VRTC pin voltage > VLVDVRTC or VRTC pin voltage < VLVDVRTC.
INTLVDVRTC = 1

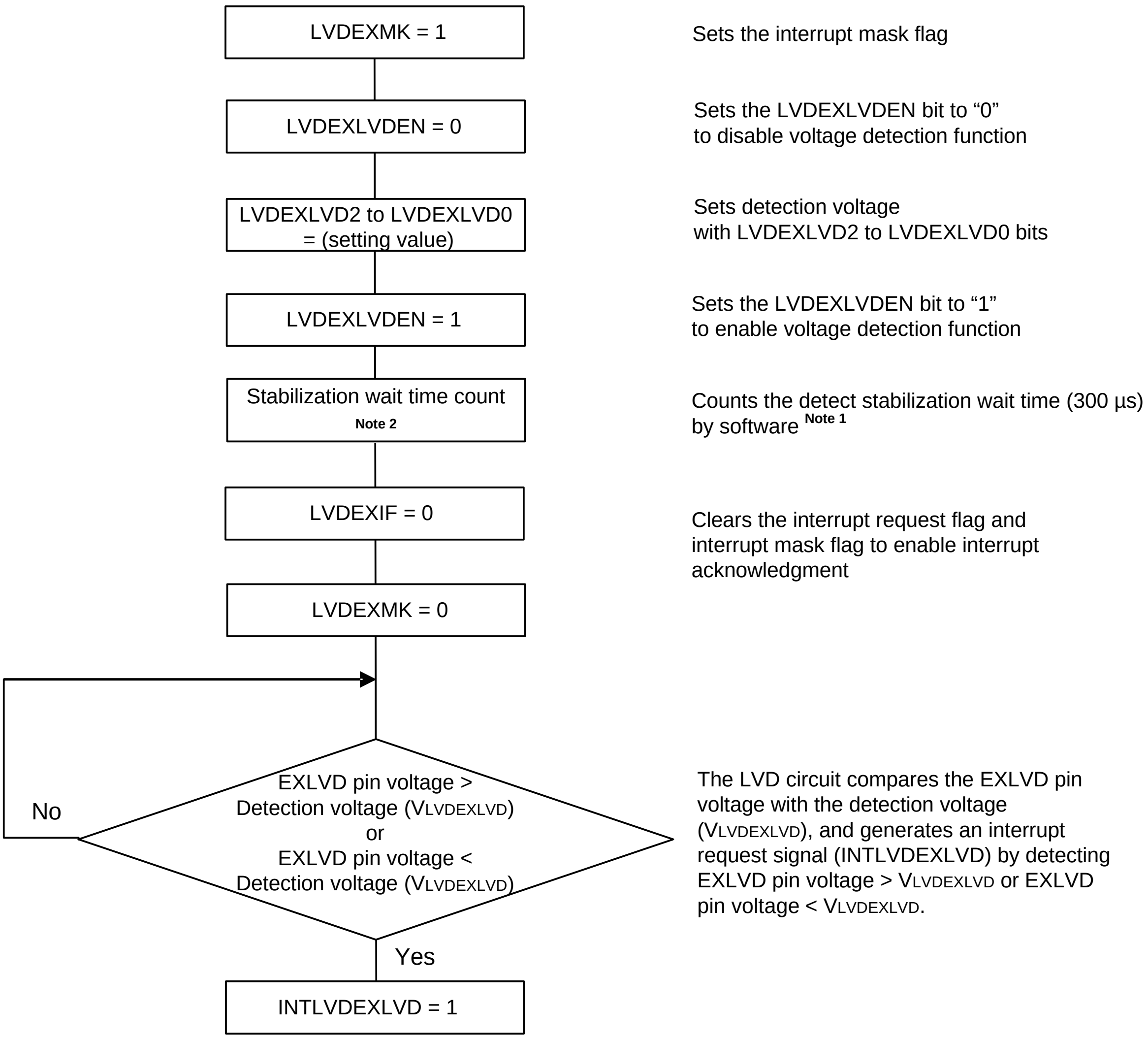
Notes 1. Be sure to set the detection voltage level when pin voltage < detection voltage.

2. If other process operating secures the stabilization wait time after setting the LVDVRTCEN bit to 1 to enable the voltage detect function operation, the count process isn't required.

(4) EXLVD

The setting procedure of EXLVD pin voltage detection is shown below.

Figure 32-24. Setting Procedure of EXLVD Pin Voltage Detection


LVDEXMK = 1
LVDEXLVDEN = 0
LVDEXLVD2 to LVDEXLVD0 = (setting value)
LVDEXLVDEN = 1
Stabilization wait time count Note 2
LVDEXIF = 0
LVDEXMK = 0
EXLVD pin voltage > Detection voltage (VLVDEXLVD) or EXLVD pin voltage < Detection voltage (VLVDEXLVD)
No
Yes
INTLVDEXLVD = 1

Sets the interrupt mask flag

Sets the LVDEXLVDEN bit to “0” to disable voltage detection function

Sets detection voltage with LVDEXLVD2 to LVDEXLVD0 bits

Sets the LVDEXLVDEN bit to “1” to enable voltage detection function

Counts the detect stabilization wait time (300 μs) by software Note 1

Clears the interrupt request flag and interrupt mask flag to enable interrupt acknowledgment

The LVD circuit compares the EXLVD pin voltage with the detection voltage (VLVDEXLVD), and generates an interrupt request signal (INTLVDEXLVD) by detecting EXLVD pin voltage > VLVDEXLVD or EXLVD pin voltage < VLVDEXLVD.

Notes 1. Be sure to set the detection voltage level when pin voltage < detection voltage.

2. If other process operating secures the stabilization wait time after setting the LVDEXLVDEN bit to 1 to enable the voltage detect function operation, the count process isn't required.

32.5 Changing of LVD Detection Voltage Setting

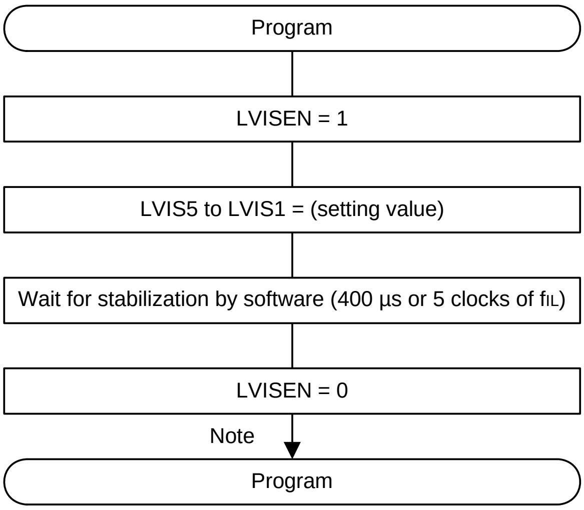
To change the LVD detection voltage by software, use the following procedure.

The LVD detection voltage can be changed in interrupt mode and reset mode.

In interrupt & reset mode, the value of the LVD detection voltage cannot be changed. Keep the initial value (set value in the option byte) unchanged.

To use two or more LVD detection voltages by changing LVISEL4 to LVISEL0 in the LVIS register by software, the highest voltage value of the used LVD detection voltages must be specified in the VPOC2 to VPOC0, LVIS1, and LVIS0 bits in the option byte (000C1H).

Figure 32-25. Changing of LVD Detection Voltage Setting

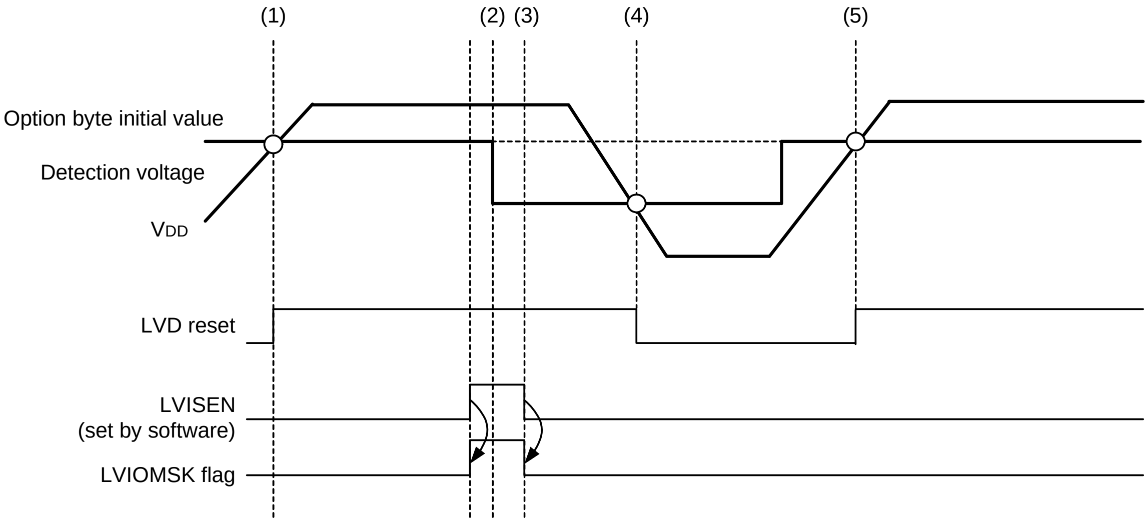

Program
LVISEN = 1
LVIS5 to LVIS1 = (setting value)
Wait for stabilization by software (400 μs or 5 clocks of f_{IL})
LVISEN = 0
Note
Program

Note After LVISEN is set to 0, LVD is detected if $V_{LVD} > V_{DD}$, and a reset/interrupt is generated.

32.5.1 Changing of LVD detection voltage setting in LVD reset mode

Figure 32-26 shows an example of timing for changing LVD detection voltage setting in LVD reset mode.

Figure 32-26. Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) The value of the LVIS register is changed.

(3) Waiting for stabilization by software is completed (400 μs or five f_{IL} clock cycles after (2))

(4) At LVD detection (falling), the detection voltage set by the LVIS register

(5) At the LVD reset release (rising), the detection voltage set by the option byte

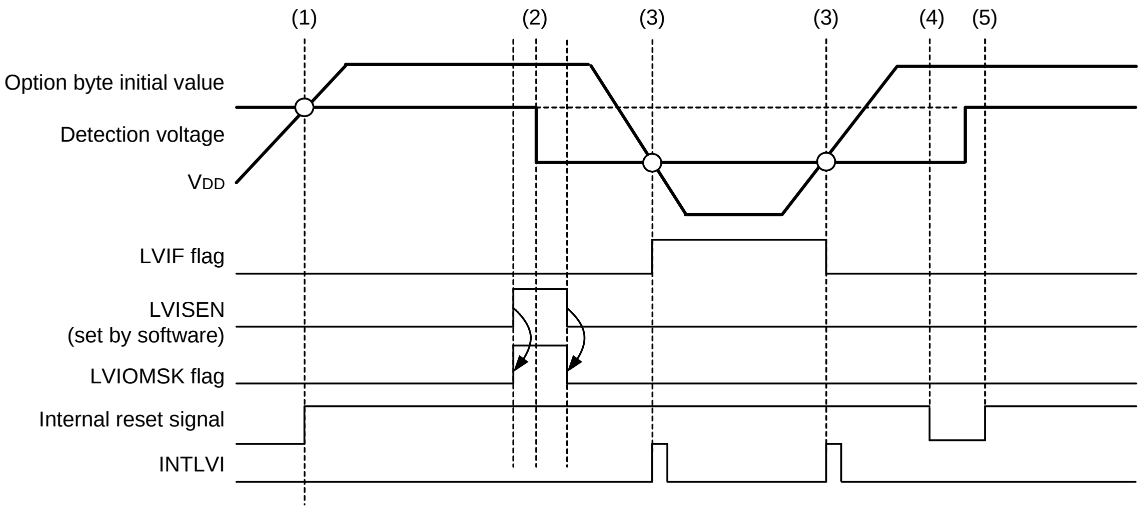
When changing the LVD detection voltage setting, note the following.

Caution The value of the reset release voltage in LVD reset mode is set to the set value in the option byte.

32.5.2 Changing of LVD detection voltage setting in LVD interrupt mode

Figure 32-27 shows an example of timing for changing LVD detection voltage setting in LVD interrupt mode.

Figure 32-27. Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) The value of the LVIS register is changed.

(3) At LVD detection (falling and rising), the detection voltage set by the LVIS register

(4) An internal reset is generated.

(5) The voltage value is changed to the set value in the option byte again when the internal reset is released.

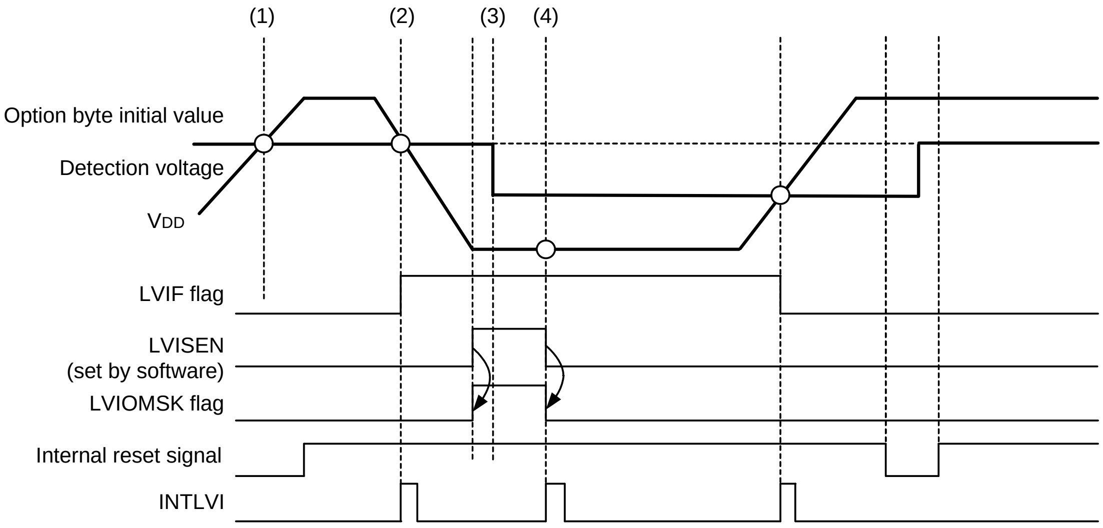
When changing the LVD detection voltage setting, note the following.

Cautions 1. Immediately after all resets are generated, the LVD internal reset retains its reset state until V_{DD} ≥ V_{LVD} (set value in the option byte). The LVD internal reset is released when V_{DD} ≥ V_{LVD} is detected (set value in the option byte).

After that, an interrupt request signal (INTLVI) is generated when V_{DD} < V_{LVD} or V_{DD} ≥ V_{LVD} is detected.

2. If the LVD set voltage is changed by setting LVISEL4 to LVISEL0 in the LVIS register while V_{DD} < V_{LVD}, an LVD interrupt is generated when the masking is released (LVISEN = 0). See Figure 32-28.

Figure 32-28. Example of Timing for Changing LVD Detection Voltage Using LVIS When $V_{DD} < V_{LVD}$



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) At LVD detection (falling), the detection voltage set by the option byte

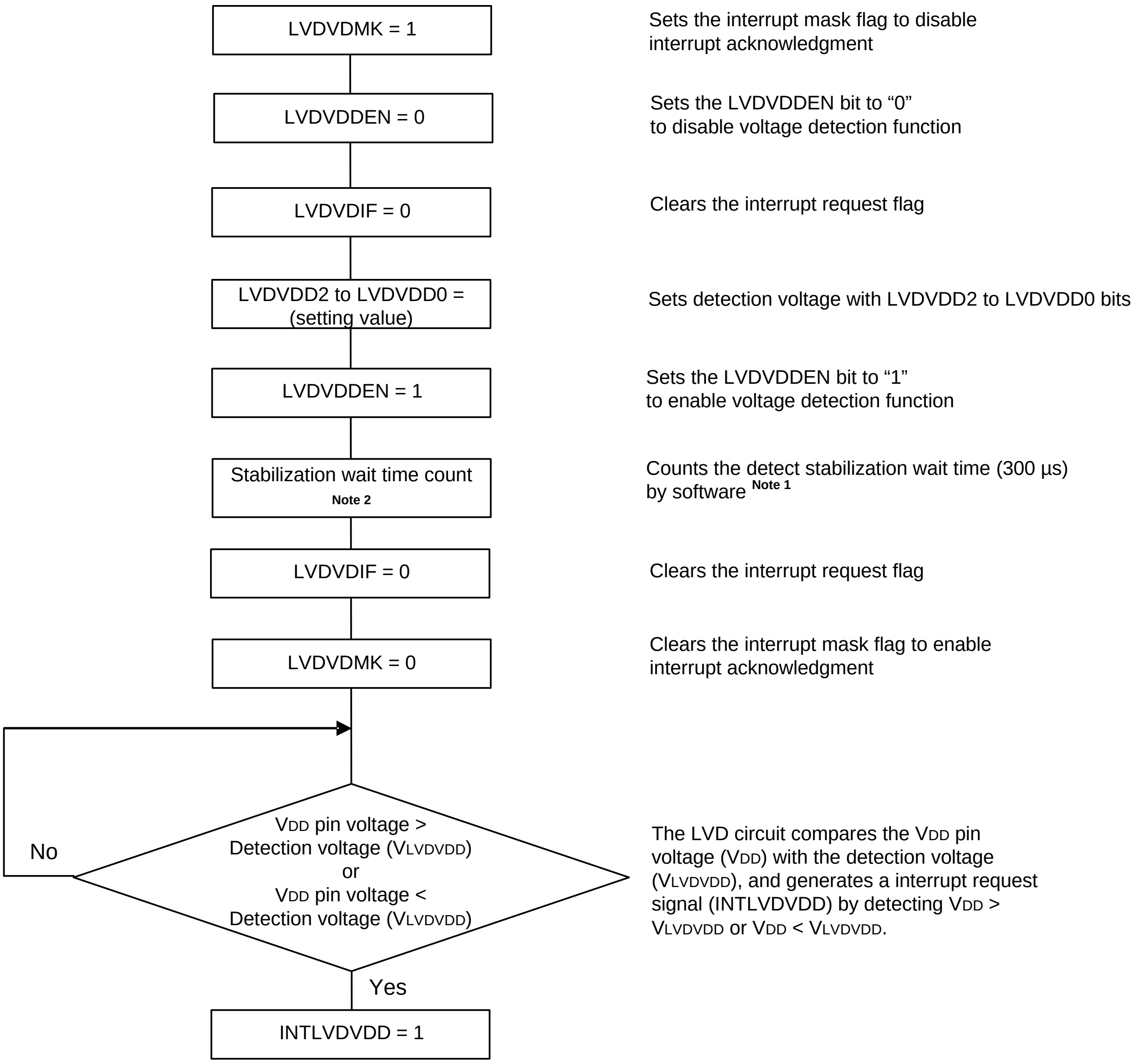
(3) The value of the LVIS register is changed.

(4) If $V_{DD} < V_{LVD}$ at the same time the masking is released, an interrupt is generated.

32.5.3 Changing of each power supply pin LVD detection voltage setting

To change the LVD detection voltage during voltage detection, use the following procedure.

Figure 32-29. Changing of LVD Detection Voltage Setting (V_{DD} pin)


LVDVDMK = 1
Sets the interrupt mask flag to disable interrupt acknowledgment
LVDVDDEN = 0
Sets the LVDVDDEN bit to “0” to disable voltage detection function
LVDVDIF = 0
Clears the interrupt request flag
LVDVDD2 to LVDVDD0 = (setting value)
Sets detection voltage with LVDVDD2 to LVDVDD0 bits
LVDVDDEN = 1
Sets the LVDVDDEN bit to “1” to enable voltage detection function
Stabilization wait time count Note 2
Counts the detect stabilization wait time (300 μs) by software Note 1
LVDVDIF = 0
Clears the interrupt request flag
LVDVDMK = 0
Clears the interrupt mask flag to enable interrupt acknowledgment
VDD pin voltage > Detection voltage (VLVDVDD) or VDD pin voltage < Detection voltage (VLVDVDD)
No
Yes
The LVD circuit compares the VDD pin voltage (VDD) with the detection voltage (VLVDVDD), and generates a interrupt request signal (INTLVDVDD) by detecting VDD > VLVDVDD or VDD < VLVDVDD.
INTLVDVDD = 1

Notes 1. Be sure to set the detection voltage level when pin voltage < detection voltage.

2. If other process operating secures the stabilization wait time after setting the LVDVDDEN bit to 1 to enable the voltage detect function operation, the count process isn't required.

32.6 Cautions for Voltage Detector

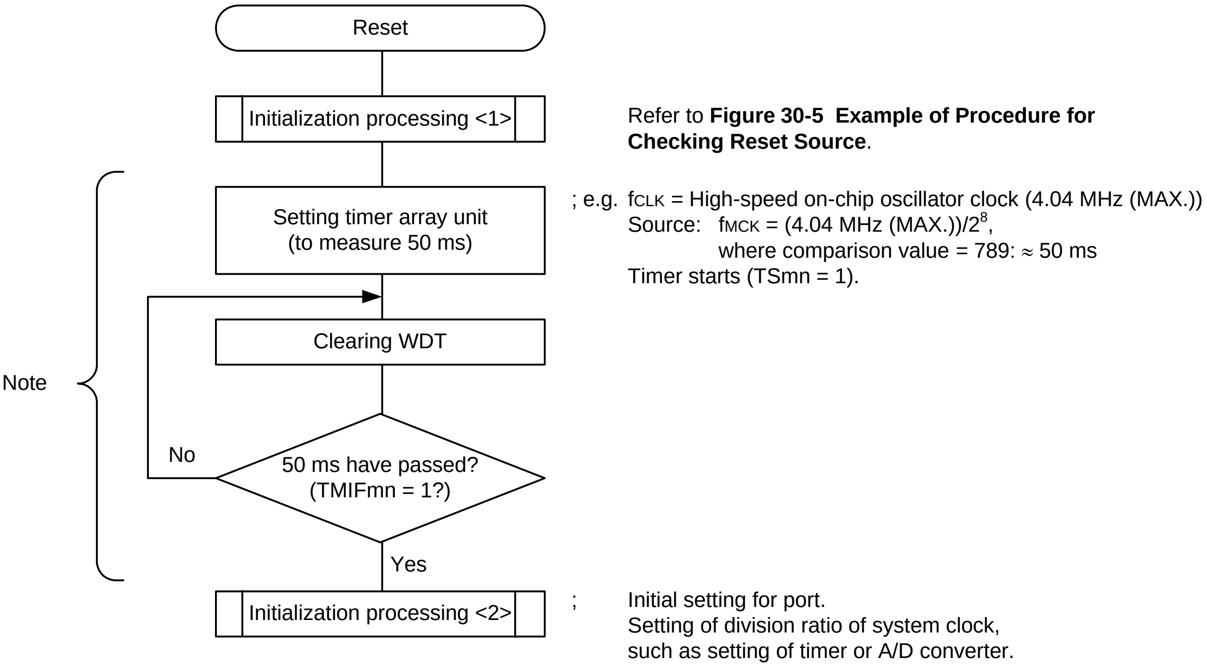
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. 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 32-30. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage


Reset
Initialization processing <1>
Refer to **Figure 30-5 Example of Procedure for Checking Reset Source**.
Setting timer array unit (to measure 50 ms)
; e.g. f_{CLK} = High-speed on-chip oscillator clock (4.04 MHz (MAX.))
Source: f_{MCK} = (4.04 MHz (MAX.))/2^8,
where comparison value = 789: ≈ 50 ms
Timer starts (TSmn = 1).
Clearing WDT
Note
No
50 ms have passed? (TMIFmn = 1?)
Yes
Initialization processing <2>
; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note If reset is generated again during this period, initialization processing <2> is not started.

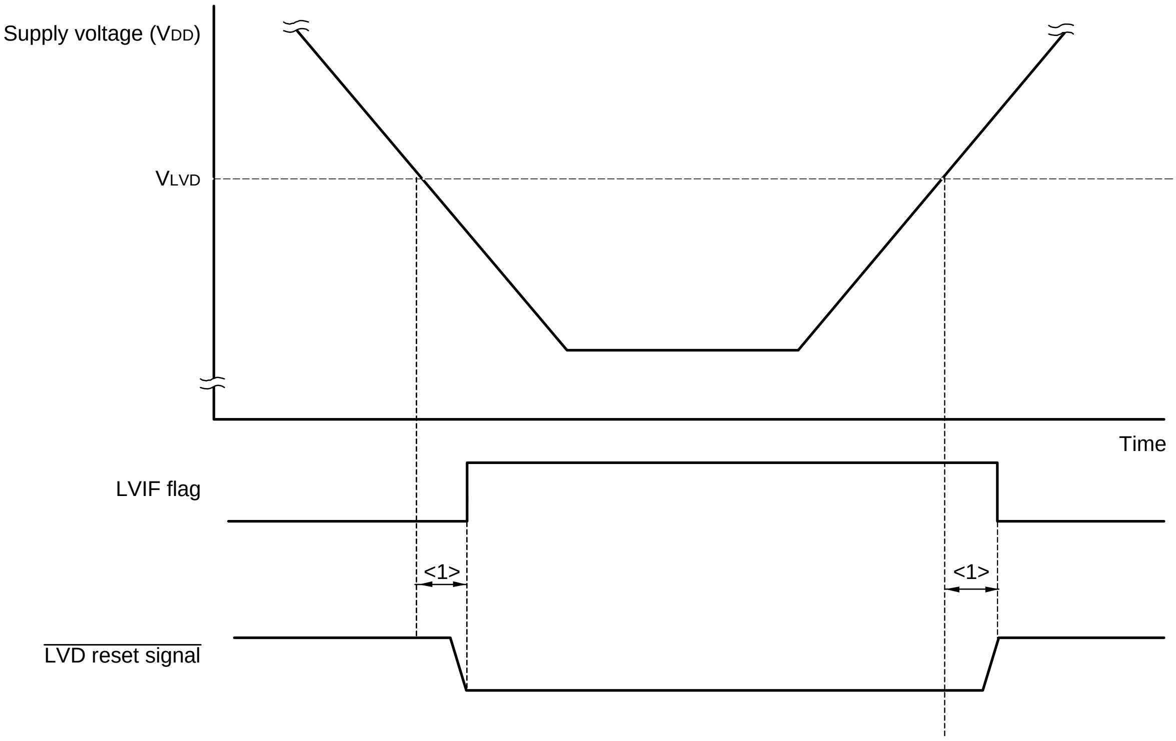
Remark m = 0
n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 32-31**).

Figure 32-31. Delay from the Time LVD Reset Source Is Generated until the Time LVD Reset has Been Generated or Released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD 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 **43.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is 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 **43.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

(5) When voltage supply to the V_{DD} pin is shut off

When voltage supply to the V_{DD} pin is shut off, the voltage detection function of the VRTC pin cannot be used, even if voltage supply to the VRTC pin is not shut off.

CHAPTER 33 OSCILLATION STOP DETECTOR

33.1 Functions of Oscillation Stop Detector

The oscillation stop detection circuit monitors the sub clock (f_{SX}) operating status with a low-speed on-chip oscillator clock (f_{IL}). If it detects that operation is stopped longer than a predefined interval, it assumes that an XT1 oscillator circuit error has occurred and outputs an oscillation stop interrupt signal.

When the system is reset, operation of the oscillation stop detector must be enabled by software after the reset period ends.

Operation of the oscillation stop detector is stopped by software. Or, oscillation stop detection operation is stopped by reset from the $\overline{\text{RESET}}$ pin or internal reset due to execution of an invalid instruction[Note]. Furthermore, after a reset, enable oscillation stop detection operation with software.

Note Occurs when instruction code for FFH is executed.
Reset due to invalid instruction does not occur during emulation with in-circuit emulator or on-chip debug emulator.

The period used by the oscillation stop detector to judge that oscillation is 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 (f_{IL}) cycle × ((value of OSDCCMP11 to OSDCCMP0) + 1)

- OSDCCMP11 to OSDCCMP0 = 003H: 232 µs (MIN.), 267 µs (TYP.), 314 µs (MAX.)
- OSDCCMP11 to OSDCCMP0 = FFFH: 237 ms (MIN.), 273 ms (TYP.), 322 ms (MAX.)

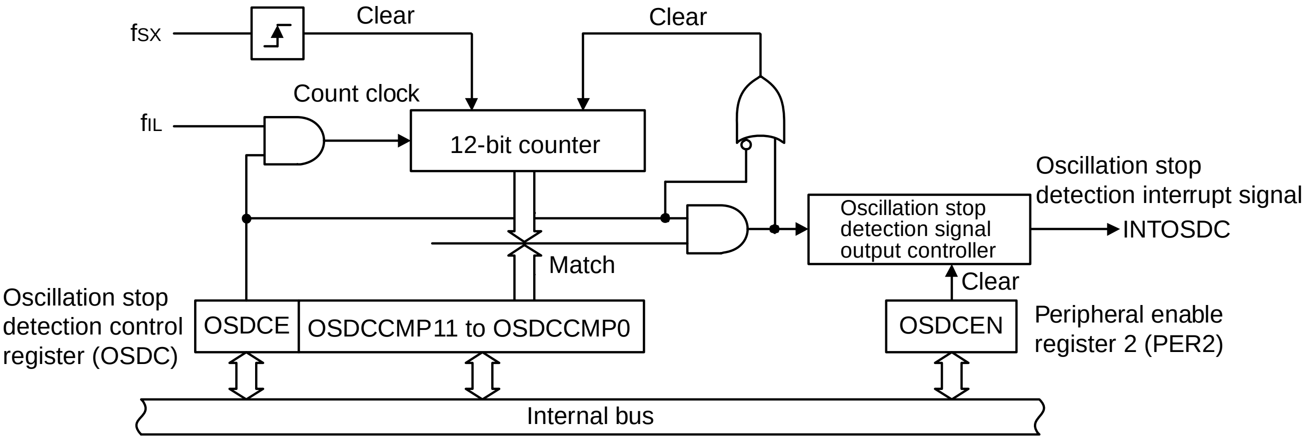
33.2 Configuration of Oscillation Stop Detector

The oscillation stop detector includes the following hardware.

Table 33-1. Configuration of Oscillation Stop Detector

Item	Configuration
Control registers	Peripheral enable register 2 (PER2) Peripheral reset control register 2 (PRR2) Subsystem clock supply option control register (OSMC) Oscillation stop detection control register (OSDC)

Figure 33-1. Block Diagram of Oscillation Stop Detector


fSX
Clear
Clear
Count clock
fIL
12-bit counter
Oscillation stop detection interrupt signal
INTOSDC
Oscillation stop detection signal output controller
Match
Clear
Oscillation stop detection control register (OSDC)
OSDCE
OSDCCMP11 to OSDCCMP0
OSDCEN
Peripheral enable register 2 (PER2)
Internal bus

33.3 Registers Used by Oscillation Stop Detector

33.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the oscillation stop detector is to be used, be sure to set bit 6 (OSDCEN) to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 33-2. Format of Peripheral Enable Register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

OSDCEN	Control of oscillation stop detection circuit input clock supply
0	Stops input clock supply. ● SFR used by the oscillation stop detection circuit cannot be written. The read value is 00H. However, the SFR is not initialized. Note
1	Enables input clock supply. ● SFR used by the oscillation stop detection circuit can be read and written.

Note To initialize the oscillation stop detection circuit and the SFR used by the oscillation stop detection circuit, use bit 6 (OSDCRES) of PRR2.

Cautions 1. When using the oscillation stop detector, be sure to set the OSDCEN bit to 1. If OSDCEN = 0, writing to a control register of the oscillation stop detector is ignored, and, even if the register is read, only the default value is read.

2. Be sure to set bits 3 and 1 to "0".

33.3.2 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.

To place the oscillation stop detection circuit in the reset state, be sure to set bit 6 (OSDCRES) to 1.

The PRR0 register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 33-3. Format of Peripheral Reset Control Register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	0
PRR2	TMKARES	OSDCRES	UARTMG1RES	UARTMG0RES	0	MACRES	0	0

OSDCRES	Control resetting of the oscillation stop detection circuit
0	The oscillation stop detection circuit is released from the reset state.
1	The oscillation stop detection circuit is in the reset state.

33.3.3 Subsystem clock supply option control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the independent power supply RTC, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 is stopped in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock.

In addition, the OSMC register can be used to select the operating clock of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1.

The low-speed on-chip oscillator clock cannot be selected as the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1. When the serial interface UARTMG0 or UARTMG1, sampling output timer detector 0, or sampling output timer detector 1 is to be used, select the sub clock (f_{SX}) as the operating clock by setting the WUTMMCK0 bit to 0.

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

Reset signal generation clears this register to 00H.

Figure 33-4. Format of Subsystem clock supply option control register (OSMC)

Address: F00F3H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0

RTCLPC [Note4]	Setting in STOP mode or HALT mode while sub clock (f_{SX}) is selected as CPU clock
0	Enables supply of sub clock (f_{SX}) to peripheral functions (See **Tables 29-1** to **29-3** for peripheral functions whose operations are enabled.)
1	Stops supply of sub clock (f_{SX}) to peripheral functions other than the independent power supply RTC, 12-bit interval timer,8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, oscillation stop detection circuit, and frequency measurement circuit.

WUTMMCK0	Selection of the operating clock for the 12-bit interval timer, 8-bit interval timer, LCD controller/driver, frequency measurement circuit, and timers RJ0 and RJ1	Selection of the count operation/stop trigger clock for the frequency measurement circuit	Selection of the output clock for the clock output/buzzer output controller	Selection of the operating clock for the serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, and sampling output timer detector 1
0	Sub clock (f_{SX})	Sub clock (f_{SX}) selected.	Sub clock (f_{SX})	Sub clock (f_{SX})
1	Low-speed on-chip oscillator clock (f_{IL}) [Notes 2, 3, 6, 7]	Low-speed on-chip oscillator clock (f_{IL}) selected [Note 6]	Clock output is prohibited. [Note 5]	Setting prohibited

Notes 1. Be sure to set bits 0 to 3, 5, and 6 to 0.

2. Do not set the WUTMMCK0 bit to 1 while the sub clock (f_{SX}) is oscillating.

3. Switching between the sub clock (f_{SX}) and the low-speed on-chip oscillator clock (fIL) can be enabled by the WUTMMCK0 bit only when all of the 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller, LCD controller/driver, frequency measurement circuit, serial interfaces UARTMG0 and UARTMG1, sampling output timer detector 0, sampling output timer detector 1, and timers RJ0 and RJ1 are stopped.

4. When the sub clock (f_{SX}) 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.

5. When the WUTMMCK0 bit is set to 1, clock output from the PCLBUZn pin is prohibited.

6. When the WUTMMCK0 bit is set to 1, the low-speed on-chip oscillator clock (f_{IL}) oscillates.

7. When the WUTMMCK0 bit is set to 1, internal voltage boosting cannot be used for the LCD drive voltage generator of the LCD controller/driver.

33.3.4 Oscillation stop detection control register (OSDC)

This 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.

Reset signal generation clears this register to 0FFFFH.

Figure 33-5. Format of Oscillation Stop Detection Control Register (OSDC)

Address: F02D0H After reset: 0FFFFH R/W

Symbol	15	14	13	12	11	10	9	8
OSDC	OSDCE	0	0	0	OSDCCMP 11	OSDCCMP 10	OSDCCMP 9	OSDCCMP 8

Symbol	7	6	5	4	3	2	1	0
OSDC	OSDCCMP 7	OSDCCMP 6	OSDCCMP 5	OSDCCMP 4	OSDCCMP 3	OSDCCMP 2	OSDCCMP 1	OSDCCMP 0

OSDCE	Control of oscillation stop detector operation
0	Stop operation of the oscillation stop detector.
1	Start 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. It is judged that oscillation has stopped when oscillation has been stopped for (A-2) to (A+1) clock cycles, where A refers to the time specified by these bits. Oscillation stop judgment time = Low-speed on-chip oscillator clock (f_{IL}) cycle × ((value of OSDCCMP11 to OSDCCMP0) + 1)

Cautions 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.

2. The oscillation stop detector stops oscillation stop detection by setting the OSDCE bit to 0 by software or by reset from the $\overline{\text{RESET}}$ pin or internal reset due to execution of an invalid instruction[Note].

Furthermore, since the oscillation of XT1 oscillator clock is also stopped with an internal reset, after a reset, enable oscillation stop detection operation after resuming oscillation of the XT1 oscillation clock with software.

3. Be sure to set bits 14 to 12 to "0".

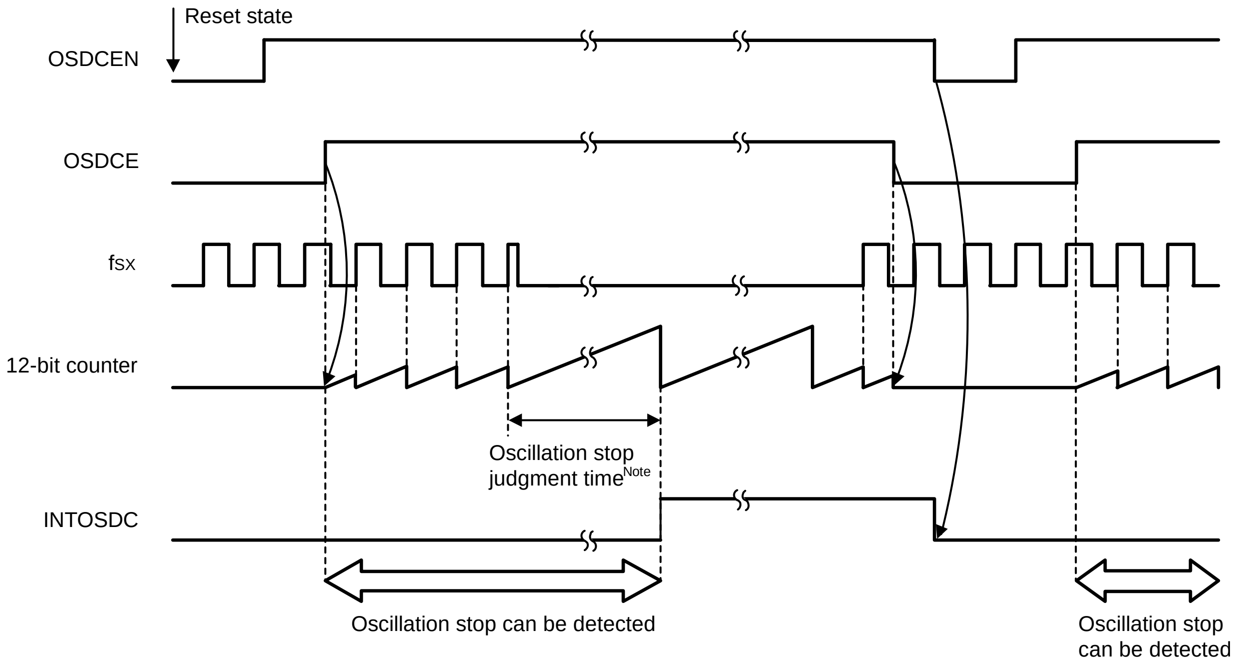
Note Occurs when instruction code for FFH is executed.
Reset due to invalid instruction does not occur during emulation with in-circuit emulator or on-chip debug emulator.

33.4 Operation of Oscillation Stop Detector

33.4.1 How the oscillation stop detector operates

1. The sub clock (fSX) starts operating after the external reset ends.
2. A value is written to the oscillation stop detection control register (OSDC) and the oscillation stop detector starts operating.
3. While the oscillation stop detector is operating, if the sub clock (fSX) stops oscillating continuously 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 33-6. Timing of Oscillation Stop Detection by Oscillation Stop Detector



Note It is judged that oscillation has stopped when oscillation has been stopped for (A-2) to (A+1) clock cycles, where A refers to the time specified by these bits.

33.5 Cautions on Using the Oscillation Stop Detector

The oscillation stop detector should be used in conjunction with the watchdog timer.
Oscillation stop detection can be used under either of the following conditions:

- When 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
- When bit 4 (WUTMMCK0) of the OSMC register is set to 1

CHAPTER 34 SAFETY FUNCTIONS

34.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/I1C (512 KB) to comply with the IEC60730 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the RL78/I1C (512 KB) that 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 CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when the RAM is read as data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

(7) A/D test function

This is used to make the 12-bit A/D converter run a self-check by A/D converting one from among the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply.

(8) Digital output signal level detection function for I/O pins

When the I/O pins are output mode, the output level of the pin can be read.

Remark For usage examples of the safety functions complying with the IEC60730 safety standards, refer to **RL78 MCU series IEC60730/60335 self test library application note (R01AN1062, R01AN1296)**.

34.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
● Flash memory CRC control register (CRC0CTL) ● Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
● CRC input register (CRCIN) ● CRC data register (CRCD)	CRC operation function (general-purpose CRC)
● RAM parity error control register (RPECTL)	RAM parity error detection function
● Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
● Timer input select register 0 (TIS0)	Frequency detection function
● A/D self-diagnosis data register (ADRD)	A/D test function
● Port mode select register (PMS)	Digital output signal level detection function for I/O ports

The content of each register is described in **34.3 Operation of Safety Functions**.

34.3 Operation of Safety Functions

34.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/I1C (512 KB) 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, 341 µs@24 MHz with 32 KB 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 CRC, because the general CRC operates in LSB first order.

34.3.1.1 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.

Reset signal generation clears this register to 00H.

Figure 34-1. Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	0	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	00000H to 03FFBH (16 K - 4 bytes)
0	0	0	0	1	00000H to 07FFBH (32 K - 4 bytes)
0	0	0	1	0	00000H to 0BFFBH (48 K - 4 bytes)
0	0	0	1	1	00000H to 0FFFBH (64 K - 4 bytes)
0	0	1	0	0	00000H to 13FFBH (80 K - 4 bytes)
0	0	1	0	1	00000H to 17FFBH (96 K - 4 bytes)
0	0	1	1	0	00000H to 1BFFBH (112 K - 4 bytes)
0	0	1	1	1	00000H to 1FFFBH (128 K - 4 bytes)
0	1	0	0	0	00000H to 23FFBH (144 K - 4 bytes)
0	1	0	0	1	00000H to 27FFBH (160 K - 4 bytes)
0	1	0	1	0	00000H to 2BFFBH (176 K - 4 bytes)
0	1	0	1	1	00000H to 2FFFBH (192 K - 4 bytes)
0	1	1	0	0	00000H to 33FFBH (208 K - 4 bytes)
0	1	1	0	1	00000H to 37FFBH (224 K - 4 bytes)
0	1	1	1	0	00000H to 3BFFBH (240 K - 4 bytes)
0	1	1	1	1	00000H to 3FFFBH (256 K - 4 bytes)
1	0	0	0	0	00000H to 43FFBH (272 K - 4 bytes)
1	0	0	0	1	00000H to 47FFBH (288 K - 4 bytes)
1	0	0	1	0	00000H to 4BFFBH (304 K - 4 bytes)
1	0	0	1	1	00000H to 4FFFBH (320 K - 4 bytes)
1	0	1	0	0	00000H to 53FFBH (336 K - 4 bytes)
1	0	1	0	1	00000H to 57FFBH (352 K - 4 bytes)
1	0	1	1	0	00000H to 5BFFBH (368 K - 4 bytes)
1	0	1	1	1	00000H to 5FFFBH (384 K - 4 bytes)
1	1	0	0	0	00000H to 63FFBH (400 K - 4 bytes)
1	1	0	0	1	00000H to 67FFBH (416 K - 4 bytes)
1	1	0	1	0	00000H to 6BFFBH (432 K - 4 bytes)
1	1	0	1	1	00000H to 6FFFBH (448 K - 4 bytes)
1	1	1	0	0	00000H to 73FFBH (464 K - 4 bytes)
1	1	1	0	1	00000H to 77FFBH (480 K - 4 bytes)
1	1	1	1	0	00000H to 7BFFBH (496 K - 4 bytes)
1	1	1	1	1	00000H to 7FFFBH (512 K - 4 bytes)

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.

34.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 34-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

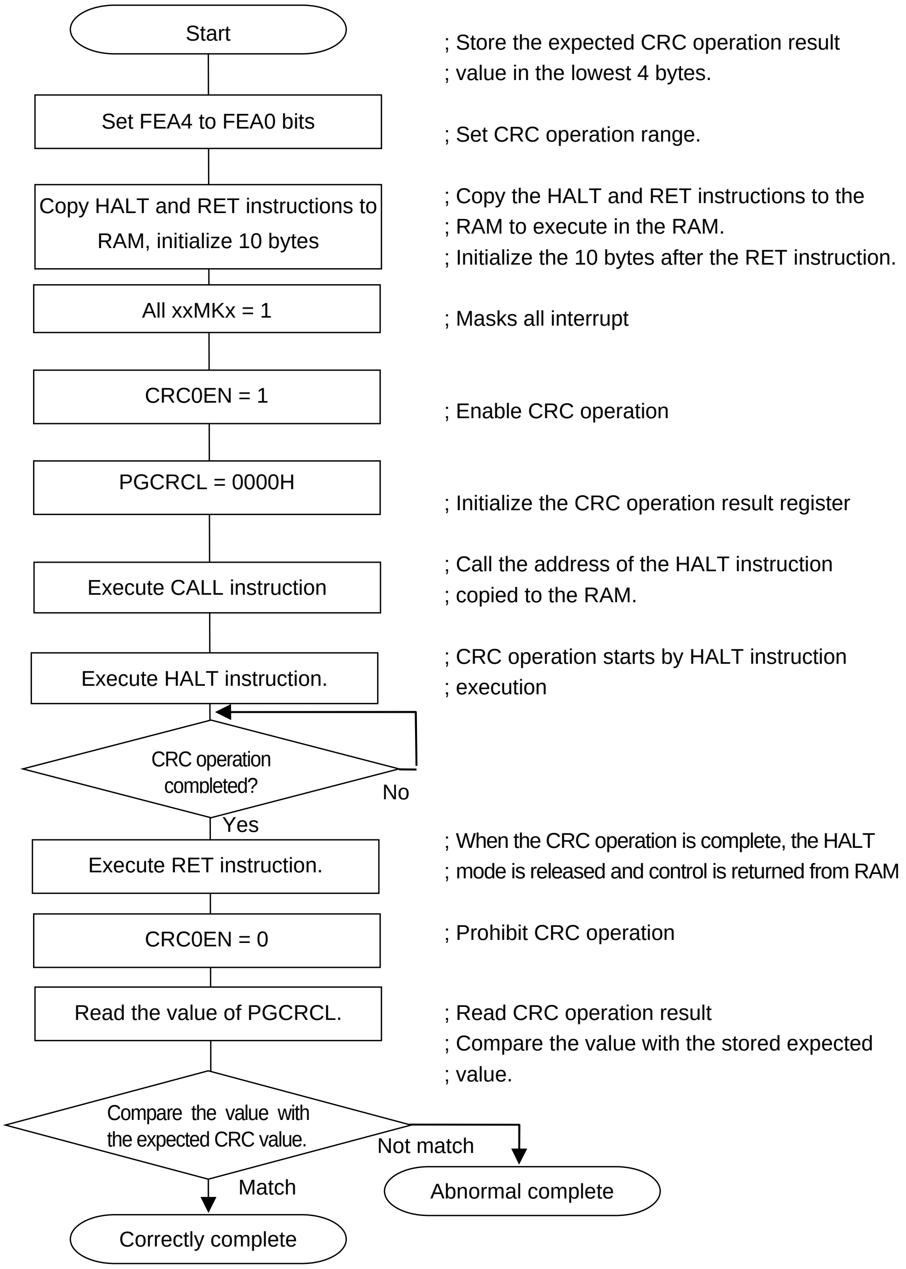
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 34-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 34-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)


Start
; Store the expected CRC operation result
; value in the lowest 4 bytes.
Set FEA4 to FEA0 bits
; Set CRC operation range.
Copy HALT and RET instructions to RAM, initialize 10 bytes
; Copy the HALT and RET instructions to the
; RAM to execute in the RAM.
; Initialize the 10 bytes after the RET instruction.
All xxMKx = 1
; Masks all interrupt
CRC0EN = 1
; Enable CRC operation
PGCRCL = 0000H
; Initialize the CRC operation result register
Execute CALL instruction
; Call the address of the HALT instruction
; copied to the RAM.
Execute HALT instruction.
; CRC operation starts by HALT instruction
; execution
CRC operation completed?
No
Yes
Execute RET instruction.
; When the CRC operation is complete, the HALT
; mode is released and control is returned from RAM
CRC0EN = 0
; Prohibit CRC operation
Read the value of PGCRCL.
; Read CRC operation result
; Compare the value with the stored expected
; value.
Compare the value with the expected CRC value.
Not match
Abnormal complete
Match
Correctly complete

Cautions 1. The CRC operation is executed only on the code flash.

2. Store the expected CRC operation value in the area below the operation range in the code flash.

3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

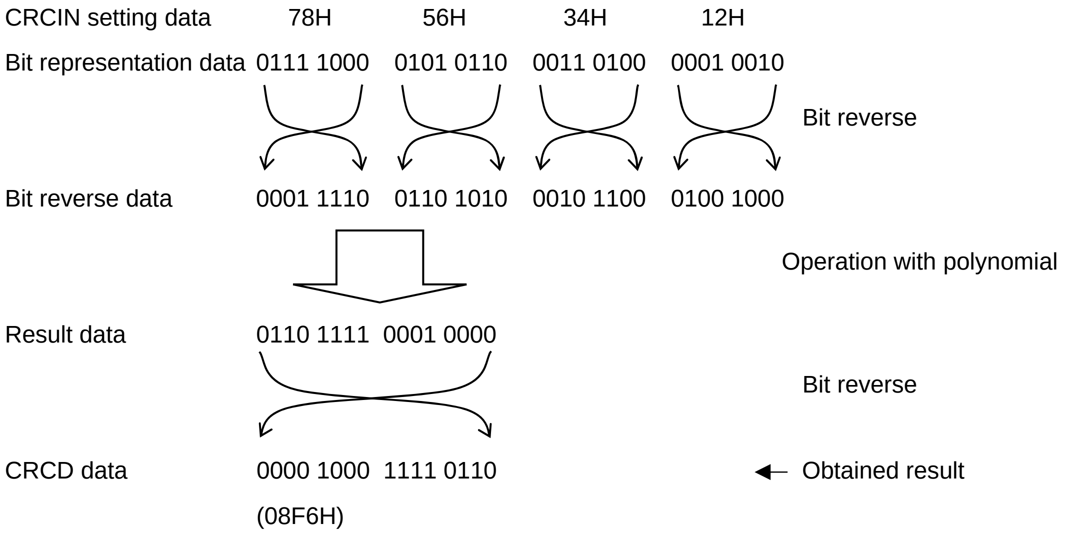
The expected CRC value can be calculated by using the Integrated Development Environment CS+. See **Integrated Development Environment CS+ user's manual** for details.

34.3.2 CRC operation function (general-purpose CRC)

<R> In the RL78/I1C (512 KB), a general CRC operation can be executed as a peripheral function while the CPU is operating. The general 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). CRC calculation function in the HALT mode can be used only during the DTC transmission.

The general CRC operation can be executed in the main system clock operation mode as well as 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.

34.3.2.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 34-4. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

34.3.2.2 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC.

The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.
The CRCD register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 34-5. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

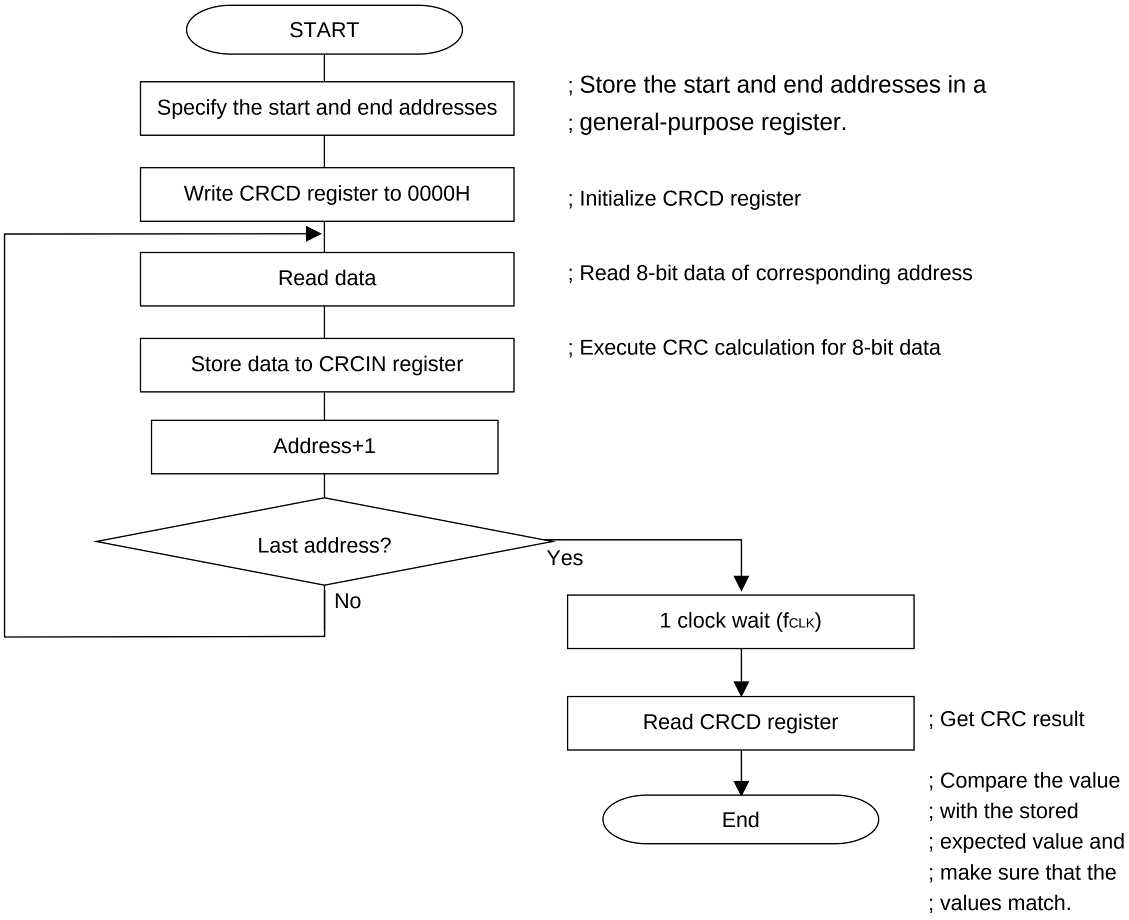
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 34-6. CRC Operation Function (General-purpose CRC)


START
Specify the start and end addresses
; Store the start and end addresses in a
; general-purpose register.
Write CRCD register to 0000H
; Initialize CRCD register
Read data
; Read 8-bit data of corresponding address
Store data to CRCIN register
; Execute CRC calculation for 8-bit data
Address+1
Last address?
Yes
No
1 clock wait (fCLK)
Read CRCD register
; Get CRC result
End
; Compare the value
; with the stored
; expected value and
; make sure that the
; values match.

34.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RAM of the RL78/I1C (512 KB). By using this RAM parity error detection function, 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.

34.3.3.1 RAM parity error control register (RPECTL)

This register is used to control 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.
Reset signal generation clears this register to 00H.

Figure 34-7. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H 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	Enable parity error resets.
1	Disable 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.

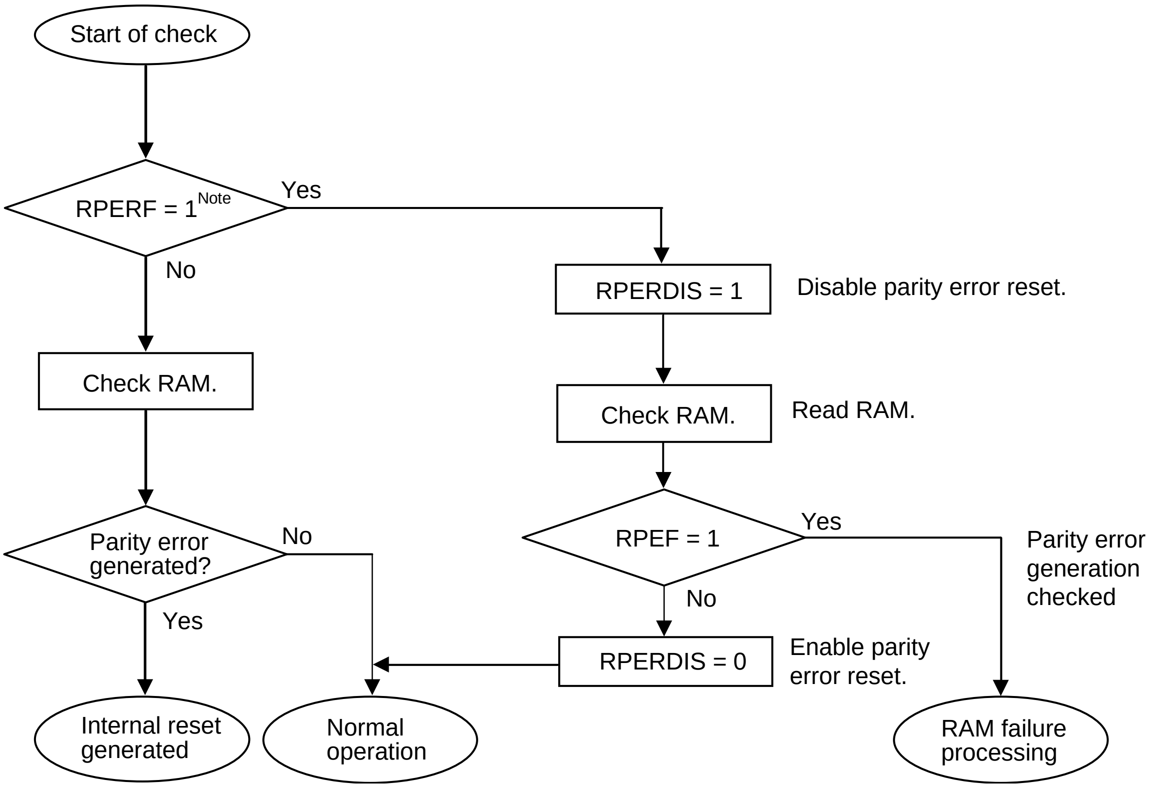
Remarks 1. The parity error reset is enabled by default (RPERDIS = 0).

2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.

3. The RPEF flag in the RPECTL register is set (1) when the RAM parity error occurs and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

4. The general registers are not included for RAM parity error detection.

Figure 34-8. Flowchart of RAM Parity Check



Note To check internal reset status using a RAM parity error, see **CHAPTER 30 RESET FUNCTION**.

34.3.4 RAM guard function

<R> This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

34.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 34-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes of space starting at the start address in the RAM
1	0	The 256 bytes of space starting at the start address in the RAM
1	1	The 512 bytes of space starting at the start address in the RAM

34.3.5 SFR guard function

<R> This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

34.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 34-10. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, PIORx, PFSEGxx, ISCLCD[Note]

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, CLKDCTL, LVIM, LVIS, RPECTL, PRRx, PMMC, MOCODIV, FMCKS, DSCCTL, MCKC

Note Pxx (Port register) is not guarded.

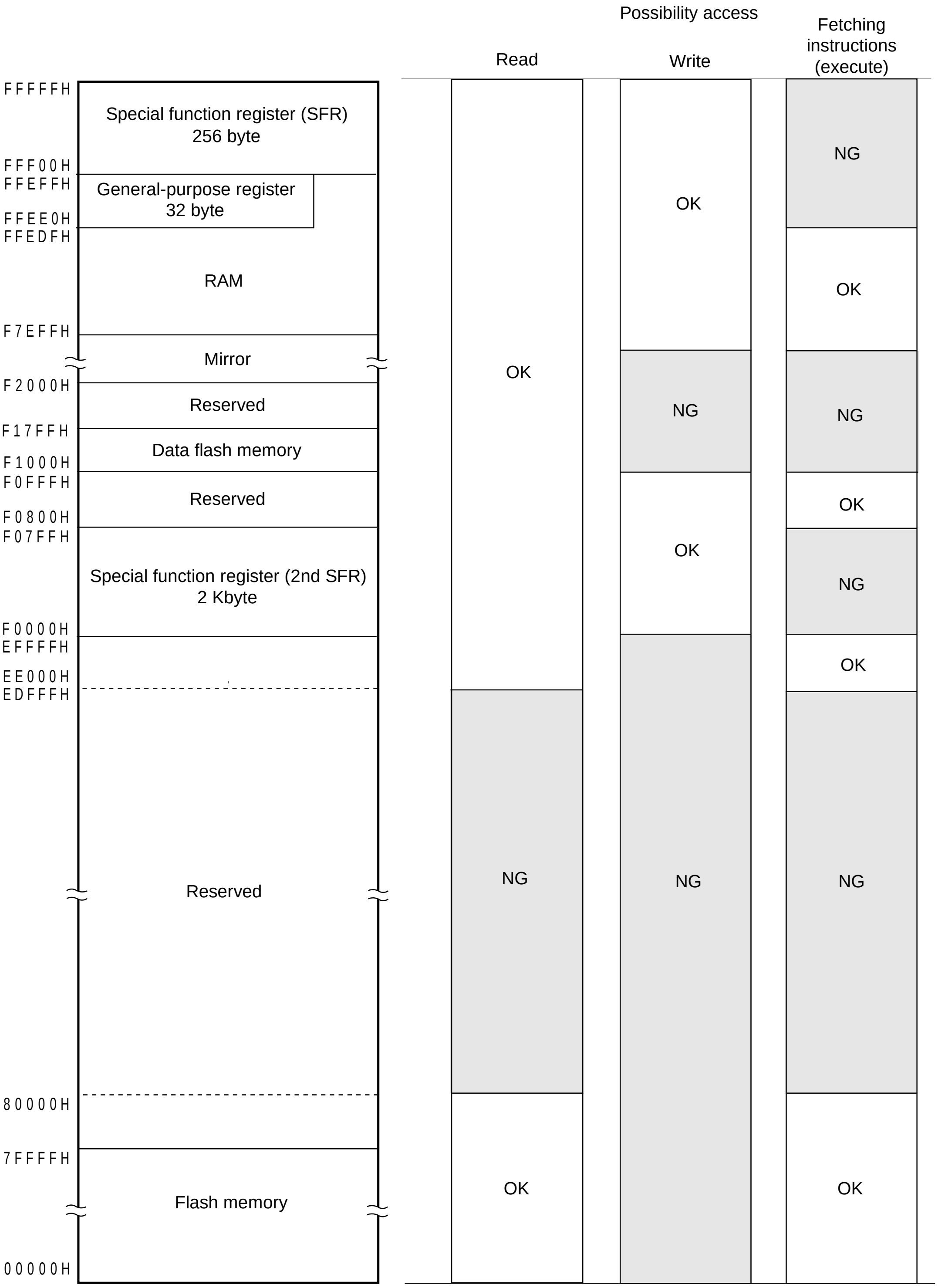
34.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in **Figure 34-11**.

Figure 34-11. Invalid Access Detection Area



34.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

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

Reset signal generation clears this register to 00H.

Figure 34-12. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 (watchdog timer operation enable) for the option byte (000C0H), the invalid memory access function is enabled even IAWEN = 0.

34.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring 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 stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

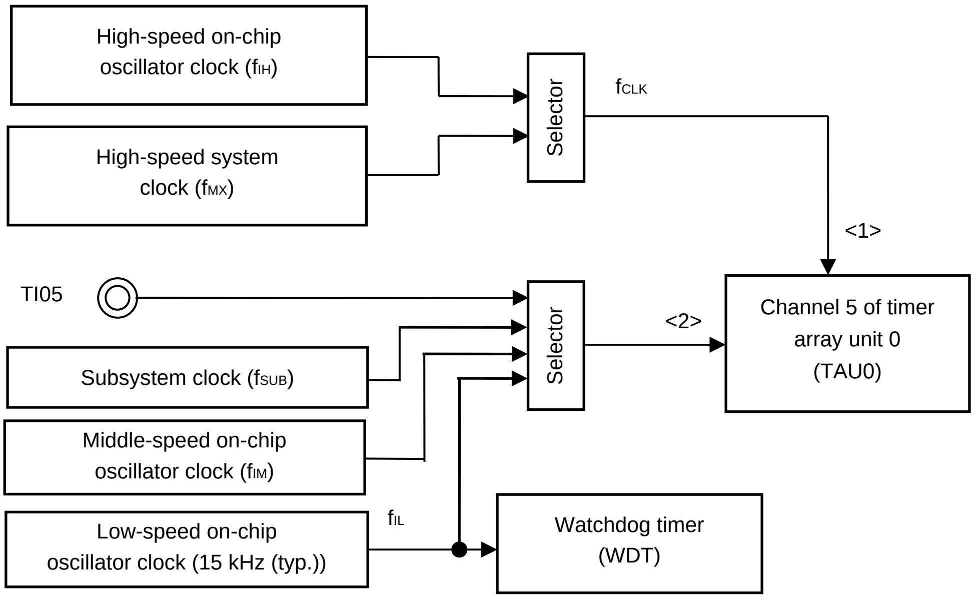
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 5 of the timer array unit

- Timer input to channel 5 (TI05)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Middle-speed on-chip oscillator clock (f_{IM})
- Subsystem clock (f_{SUB})

Figure 34-13. Configuration of Frequency Detection Function


High-speed on-chip oscillator clock (fIH)
High-speed system clock (fMX)
Selector
fCLK
<1>
TI05
<2>
Channel 5 of timer array unit 0 (TAU0)
Subsystem clock (fSUB)
Selector
Middle-speed on-chip oscillator clock (fIM)
fIL
Low-speed on-chip oscillator clock (15 kHz (typ.))
Watchdog timer (WDT)

If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute input pulse interval measurement, see **8.8.3 Operation as input pulse interval measurement**.

34.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channels 0, 1, 5, 6, and 7 of the timer array unit 0 (TAU0).

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 34-14. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI05)
0	1	1	Middle-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 above			Setting prohibited

34.3.8 A/D test function

34.3.8.1 A/D self-diagnosis data register (ADRD)

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see **17.2.9 A/D control extended register (ADCER)**.

Figure 34-15. Format of A/D Self-diagnosis Data Register (ADRD)

Address: F061Eh After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADRD																

The data formats for each given condition are shown below.

- Flush-right format

  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14. Bits 13 and 12 are read as 0.
- Flush-left format

  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0. Bits 3 and 2 are read as 0.

Table 34-1. Self-diagnosis Status Description[Note]

Bits 15 and 14 for Flush-right Format Setting Bits 1 and 0 for Flush-left Format Setting	Self-diagnosis Status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of reference power supply × 1/2 has been executed.
11b	Self-diagnosis using the voltage of reference power supply has been executed.

Note For details of self-diagnosis, see **17.2.9 A/D control extended register (ADCER)**.

34.3.9 Digital output signal level detection function for I/O ports

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O pins, the digital output level of the pin can be read when the pin is set to output mode.

34.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the port is output mode in which PMm bit of port mode register (PMm) is 0.

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

Reset signal generation clears these registers to 00H.

Figure 34-16. Format of Port Mode Select Register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when port is output mode (PMmn = 0)
0	Pmn register value is read.
1	Digital output level of the pin is read.

Cautions 1. While the PMS0 bit of the PMS register is “1”, do not change the value of the Px register by using a read-modify instruction. To change the value of the Px register, use an 8-bit manipulation instruction.

2. PMS control cannot be used for the dedicated LCD pins and the input-only pins (P121 to P124 and P137).

3. PMS control cannot be used for alternate-function pins being used as segment output pins. (“L” is always read when this register is read.)

4. PMS control cannot be used for P61 and P60 when IICA0EN (bit 4 of the PER0 register) is 0.

Remark m = 0 to 9, 12, 15
n = 0 to 7

CHAPTER 35 AES FUNCTIONS

35. 1 AES Functions

The RL78/I1C (512 KB) microcontrollers are provided with the AES function based on AES-GCM Standards, which is used in the smart meter market, for enhanced security.

Feature overview

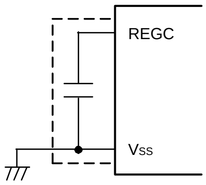
- Cipher modes of operation: GCM/ECB/CBC
- Encryption key length: 128/192/256 bits
- Number of interrupt sources: 2
- Interrupt source name: INTAES, INTAESF

For functional details, ask our sales representative.

CHAPTER 36 REGULATOR

36.1 Regulator Overview

The RL78/I1C (512 KB) contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 µF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see **Table 36-1**.

Table 36-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LV (low-voltage main) mode	1.8 V	—
LP (Low-power main) mode		
LS (low-speed main) mode		
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{SUB})
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{SUB}) has been set
	2.1 V	Other than above (include during OCD mode)Note

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

36.2 Register Controlling Regulator

The following register is used to control the regulator.

- Regulator mode control register (PMMC)

36.2.1 Regulator mode control register (PMMC)

The PMMC register is an 8-bit register used to control the mode of the on-chip regulator.

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

Reset generation sets this register to 00H.

Figure 36-1. Format of Regulator Mode Control Register (PMMC)

Address: F00F8H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	1	0
PMMC	0	MCSEL	0	0	0	0	0	0

MCSEL	Control of regulator mode
0	Normal setting
1	Low-power consumption setting

Cautions 1. Do not change the flash operation mode select register (FLMODE) when MCSEL is 1.

2. Do not set MCSEL to 1 in HS (high-speed main) mode and LV (low-voltage main) mode.

3. In LS (low-speed main) mode, transitions to the STOP mode are prohibited while MCSEL is 1.

CHAPTER 37 OPTION BYTE

37.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/I1C (512 KB) form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes. For bits for which no function is assigned, do not change their initial values.

When bank swapping is to be used during self-programming, set the same values at 400C0H to 400C3H as those at 000C0H to 000C3H, because switching between 000C0H to 000C3H and 400C0H to 400C3H will proceed.

Caution The option bytes should always be set regardless of whether each function is used.

37.1.1 User option byte (000C0H to 000C2H/400C0H to 400C2H)

(1) 000C0H/400C0H

O Operation of watchdog timer

- ● Enabling or disabling of counter operation
- ● Enabling or disabling of counter operation in the HALT or STOP mode

O Setting of overflow time of watchdog timer

O Setting of window open period of watchdog timer

O Setting of interval interrupt of watchdog timer

- ● Whether or not to use the interval interrupt is selectable.

Caution When bank swapping is to be used, set the same value at 400C0H as that at 000C0H, because switching between 000C0H and 400C0H will proceed.

(2) 000C1H/400C1H

O Setting of LVD operation mode

- ● Interrupt & reset mode.
- ● Reset mode.
- ● Interrupt mode.
- ● LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)

O Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Cautions 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 43.4 AC Characteristics. This is done by utilizing the voltage detector 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 detector 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 400C2H).

2. When bank swapping is to be used, set the same value at 400C1H as that at 000C1H, because switching between 000C1H and 400C1H will proceed.

(3) 000C2H/400C2H

O Setting of flash operation mode

It should be set, depending on the main system clock frequency (f_{MAIN}) and power supply (V_{DD}) to be used.

- LV (low-voltage main) mode
- LS (low speed main) mode
- HS (high speed main) mode

O Setting of the frequency of the high-speed on-chip oscillator

- Select from 1 MHz to 32 MHz

Caution When bank swapping is to be used, set the same value at 400C2H as that at 000C2H, because switching between 000C2H and 400C2H will proceed.

37.1.2 On-chip debug option byte (000C3H/400C3H)

O Control of on-chip debug operation

- On-chip debug operation is disabled or enabled.

O 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 bank swapping is to be used, set the same value at 400C3H as that at 000C3H, because switching between 000C3H and 400C3H will proceed.

37.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 37-1. Format of User Option Byte (000C0H/400C0H)

Address: 000C0H/400C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time + 1/2 f_{IL} is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	75%[Note 3]
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 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

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

Notes 1. When bank swapping is to be used, set the same value at 400C0H as that at 000C0H, because switching between 000C0H and 400C0H will proceed.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

3. When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 37-2. Format of User Option Byte (000C1H/400C1H) (1/2)

Address: 000C1H/400C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Note When bank swapping is to be used, set the same value at 400C1H as that at 000C1H, because switching between 000C1H and 400C1H will proceed.

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

Figure 37-2. Format of User Option Byte (000C1H/400C1H) (2/2)

Address: 000C1H/400C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

- LVD off setting (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
–		Setting of values other than above is prohibited.						

Note When bank swapping is to be used, set the same value at 400C1H as that at 000C1H, because switching between 000C1H and 400C1H will proceed.

Cautions 1. Be sure to set bit 4 to "1".

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 43.4 AC Characteristics. This is done by utilizing the voltage detector 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 detector 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 400C2H).

Remarks 1. ×: don't care

2. For LVD setting, see **32.1 Functions of Voltage Detector**.

3. The detection voltage is a typical value. For details, see **43.6.6 LVD circuit characteristics.**

Figure 37-3. Format of Option Byte (000C2H/400C2H)

Address: 000C2H/400C2H[Note 1]

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating frequency range (f_{MAIN})	Operating voltage range (V_{DD})
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high speed main) mode	1 to 6 MHz	2.1 to 5.5 V
			1 to 16 MHz	2.4 to 5.5 V
			1 to 32 MHz [Note 2]	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock (f_{IH})	
			FRQSEL3 = 0	FRQSEL3 = 1[Note 3]
0	0	0	24 MHz[Note 2]	32 MHz
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	1.5 MHz	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

Notes 1. When bank swapping is to be used, set the same value at 400C2H as that at 000C2H, because switching between 000C2H and 400C2H will proceed.

2. When the PLL clock (32 MHz) is selected as the main system clock, select 24 MHz for the high-speed on-chip oscillator clock.

3. This setting is prohibited when the high-speed on-chip oscillator clock (f_{HOCO}/2) is selected as the operating clock for the 24-bit ΔΣ A/D converter.

Cautions 1. Be sure to set bits 5 and 4 to 10B.

2. The ranges of operation frequency and operation voltage vary depending on the flash operation mode. For details, see 43.4 AC Characteristics.

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/400C3H)

Address: 000C3H/400C3H[Note]

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 debug operation.
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 erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note When bank swapping is to be used, set the same value at 400C3H as that at 000C3H, because switching between 000C3H and 400C3H will proceed.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set bits 6 to 1 to 000010B.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

37.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the link option, in addition to describing to the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT    CSEG          OPT_BYTE
       DB            36H       ; Does not use interval interrupt of watchdog timer,
                               ; Enables watchdog timer operation,
                               ; Window open period of watchdog timer is 50%,
                               ; Overflow time of watchdog timer is 2^9/fIL,
                               ; Stops watchdog timer operation during HALT/STOP mode
       DB            5AH       ; Select 2.45 V for VLVDL
                               ; Select rising edge 2.61 V, falling edge 2.55 V for VLVDH
                               ; Select the interrupt & reset mode as the LVD operation mode
       DB            A3H       ; Select the LS (low speed main) mode as the flash operation mode
                               ; and 3 MHz as the frequency of the high-speed on-chip oscillator
       DB            85H       ; Enables on-chip debug operation, does not erase flash memory
                               ; data when security ID authorization fails
```

When bank swapping is to be used during self-programming, switching between 000C0H to 000C3H and 400C0H to 400C3H will proceed.

Define the same values at 400C0H to 400C3H as those at 000C0H to 000C3H, as in the example below.

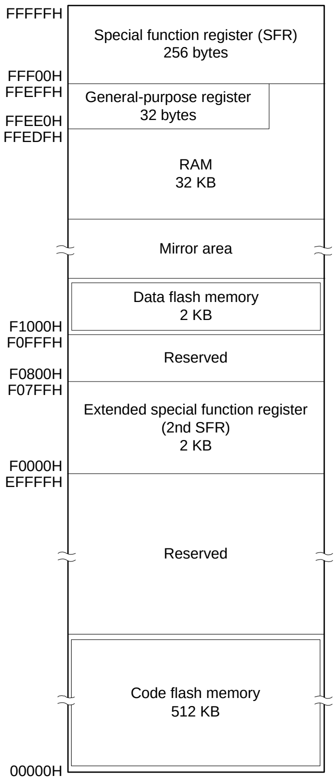
```
OPT2   CSEG    AT    400C0H
       DB            36H       ; Does not use interval interrupt of watchdog timer,
                               ; Enables watchdog timer operation,
                               ; Window open period of watchdog timer is 50%,
                               ; Overflow time of watchdog timer is 2^10/fIL,
                               ; Stops watchdog timer operation during HALT/STOP mode
       DB            5AH       ; Select 2.45 V for VLVDL
                               ; Select rising edge 2.61 V, falling edge 2.55 V for VLVDH
                               ; Select the interrupt & reset mode as the LVD operation mode
       DB            A3H       ; Select the LS (low speed main) mode as the flash operation mode
                               ; and 3 MHz as the frequency of the high-speed on-chip oscillator
       DB            85H       ; Enables on-chip debug operation, does not erase flash memory
                               ; data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 400C0H to 400C3H in order to use bank swapping, use the relocation attribute AT to specify an absolute address.

CHAPTER 38 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.

The RL78/I1C (512 KB) has a bank programming function. Bank programming enables updating of the user program while it is running. See **38.6.2**.



The following methods for programming the flash memory are available.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **38.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (that Incorporates UART) (see **38.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-Programming (see **38.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

Caution When rewriting the flash memory, stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (f_{OCO}).
Do not change the flash operation mode register (FLMODE). Rewrite the flash memory when the MCSEL bit in the regulator mode control register (PMMC) is 0.

Bank programming also enables rewriting of the code flash memory. Specifically, when the bank programming mode is selected, you can rewrite the user program in the bank for rewriting by using the flash self-programming library while the original user program is running from the startup bank. For details on the bank programming function, see **38.6.2 Bank programming function**.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **38.8 Data Flash**.

38.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP6
- E1, E2, 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. The connectors that connect the dedicated flash memory programmer must be mounted 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 38-1. Wiring Between RL78/I1C (512 KB) and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.	
Signal Name		I/O	Pin Function		80-pin	100-pin
PG-FP6	E1, E2, E2 Lite On-chip Debugging Emulator				LFQFP (12 × 12)	LFQFP (14 × 14)
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	3	9
SI/RxD	—	I/O	Transmit/ receive signal			
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	7	13
/RESET	—	Output				
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	16	22
GND		—	Ground	V_{SS}/EV_{SS0}/AV_{SS1}	15	21
				EV_{SS1}		54
				REGC[Note]	14	20
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}/EV_{DD0}	17	23
				EV_{DD1}		63

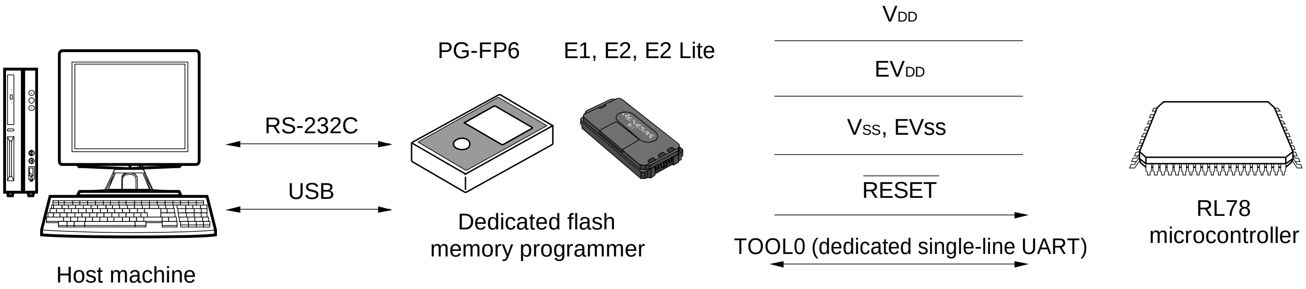
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash 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



A host machine that controls the dedicated flash memory programmer is necessary.

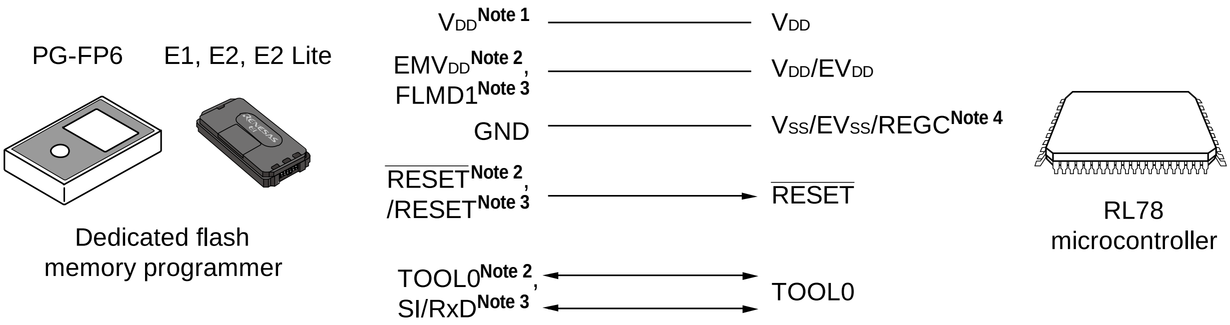
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

38.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 38-2. Communication with Dedicated Flash Memory Programmer


PG-FP6
E1, E2, E2 Lite
Dedicated flash memory programmer
VDD Note 1
EMVDD Note 2, FLMD1 Note 3
GND
RESET Note 2, /RESET Note 3
TOOL0 Note 2, SI/RxD Note 3
VDD
VDD/EVDD
VSS/EVSS/REGC Note 4
RESET
TOOL0
RL78 microcontroller

Notes 1. The name of the signal for connection in the case of the PG-FP6 is Vcc.

2. When using E1, E2, E2 Lite on-chip debugging emulator.

3. When using PG-FP6.

4. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP6, or E1, E2, E2 Lite on-chip debugging emulator for details.

Table 38-2. Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
Signal Name		I/O	Pin Function	Pin Name
PG-FP6	E1, E2, E2 Lite On-chip Debugging Emulator			
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, EV_{SS}, REGC$^{\text{Note}}$
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}, EV_{DD}
/RESET	—	Output	Reset signal	$\overline{\text{RESET}}$
—	$\overline{\text{RESET}}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

38.2 Serial Programming Using External Device (that Incorporates UART)

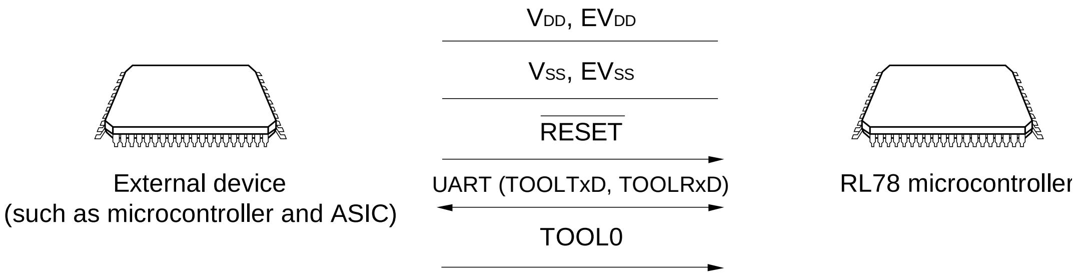
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

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



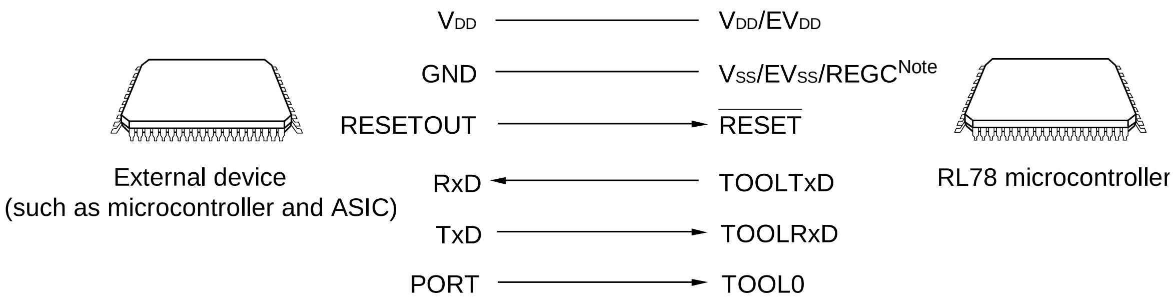
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

38.2.2 Communication mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 38-4. Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 38-3. Pin Connection

External Device			RL78 Microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}
GND	—	Ground	V_{SS}, EV_{SS}, REGC**Note**
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

38.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

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 kΩ 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 low-level is prohibited for t_{HD} period after external reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remarks 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **43.11 Timing of Entry to Flash Memory Programming Modes**).

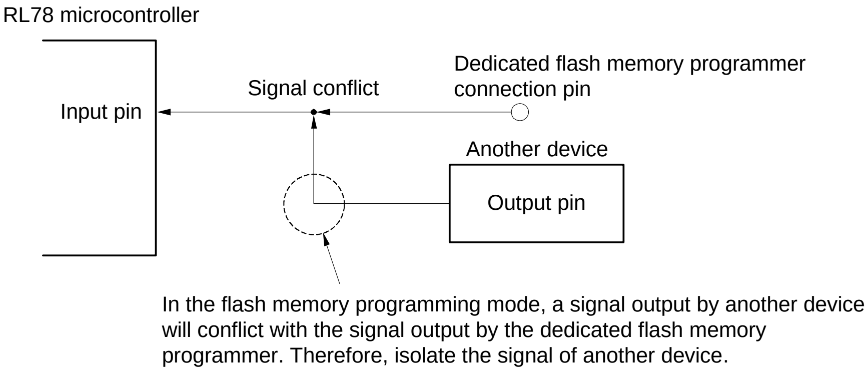
2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

38.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are 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)


RL78 microcontroller
Input pin
Signal conflict
Dedicated flash memory programmer connection pin
Another device
Output pin
In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

38.3.3 Port pins

Example When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD}, or V_{SS} via a resistor

38.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) 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 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

38.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD}**Note** of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

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 V_{DD} and V_{SS} pins to V_{DD}**Note** and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

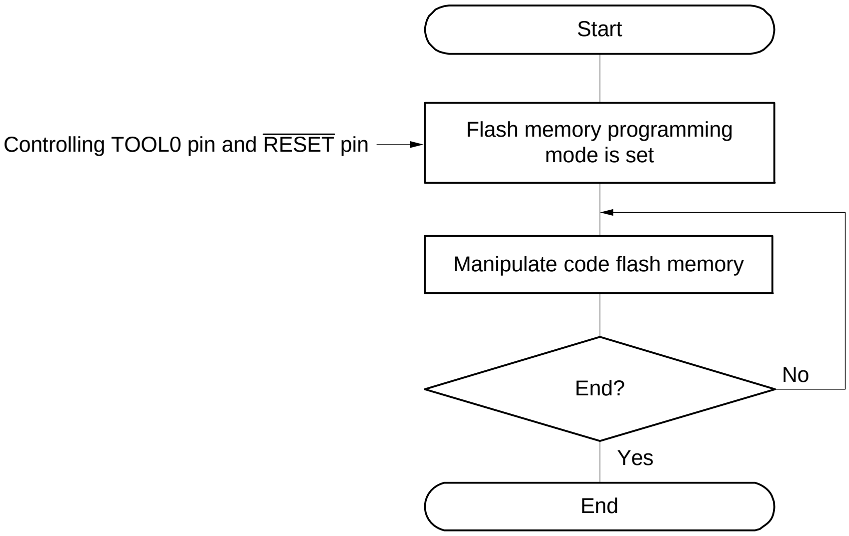
Note The name of the signal for connection in the case of the PG-FP6 is Vcc.

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


Start
Controlling TOOL0 pin and $\overline{\text{RESET}}$ pin
Flash memory programming mode is set
Manipulate code flash memory
End?
No
Yes
End

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. To enter the mode, set as follows.

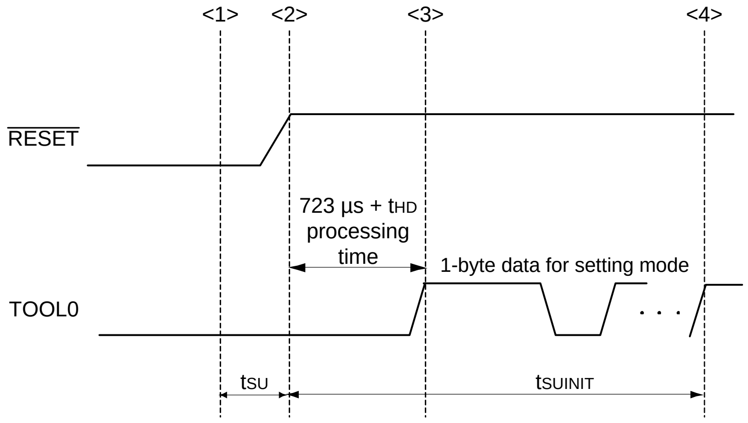
<When serial programming by using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 38-4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 38-7**. For details, refer to **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 38-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
EV_{DD0}	Normal operation mode
0 V	Flash memory programming mode

Figure 38-7. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<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 segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded).

For details, see **43.11 Timing of Entry to Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 38-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency (f_{CLK})	
$2.7\ V \le V_{DD} \le 5.5\ V$	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 32 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
$2.4\ V \le V_{DD} < 2.7\ V$	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
$1.8\ V \le V_{DD} < 2.4\ V$	Blank state		Wide voltage mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

2. For details about communication commands, see **38.4.4 Communication commands**.

38.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 38-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

38.4.4 Communication 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 **RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 38-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	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 check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory[Note].
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, 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	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Tables 38-8 and **38-9** show signature data list and example of signature data list.

Table 38-8. Signature Data List

Field Name	Description	Number 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 last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 7FFFFH (512 KB) → FFH, FFH, 07H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F17FFH (2 KB) → FFH, 17H, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 38-9. Signature Data List

Field Name	Description	Number of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F10NPL	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 4E = “N” 50 = “P” 4C = “L” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 7FFFFH (512 KB)	3 bytes	FFH FFH 07H
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FFH 17H 0FH
Firmware version	Ver.1.23	3 bytes	01 02 03

38.5 Processing Time for Each Command When Dedicated Flash Memory Programmer Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP6 is used as a dedicated flash memory programmer.

Table 38-10. Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

PG-FP6 Command	Port: TOOL0 (UART)
	Speed: 1 Mbps
	512 Kbytes
Erasing	3.5 s
Writing	9.5 s
Verification	7.3 s
Writing after erasing	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
Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

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 by using the RL78 microcontroller self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock (f_{SUB}).

2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, its clock should be operated (HIOSTOP = 0), and the flash self-programming library should be executed after 30 μs have elapsed. Stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (f_{OCO}).

4. When rewriting the flash memory, do not change the flash operation mode register (FLMODE). Rewrite the flash memory when the MCSEL bit in the regulator mode control register (PMMC) is 0.

Remark For details of the self-programming function, refer to the **RL78 Family Flash Self-Programming Library Type 01 User's Manual (R01US0050)**.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode. Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high-speed main) mode is specified. Specify the wide voltage mode when the LS (low-speed main) mode or LV (low-voltage main) mode is specified.

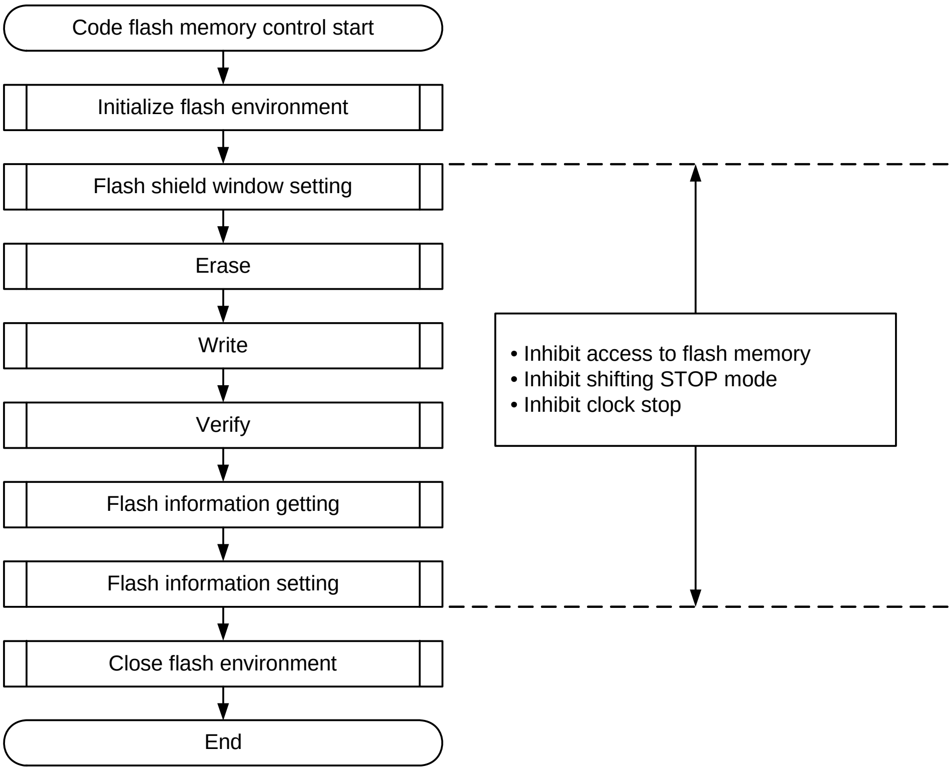
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

38.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 38-8. Flow of Self-Programming (Rewriting Flash Memory)


Code flash memory control start
Initialize flash environment
Flash shield window setting
Erase
Write
Verify
Flash information getting
Flash information setting
Close flash environment
End
• Inhibit access to flash memory
• Inhibit shifting STOP mode
• Inhibit clock stop

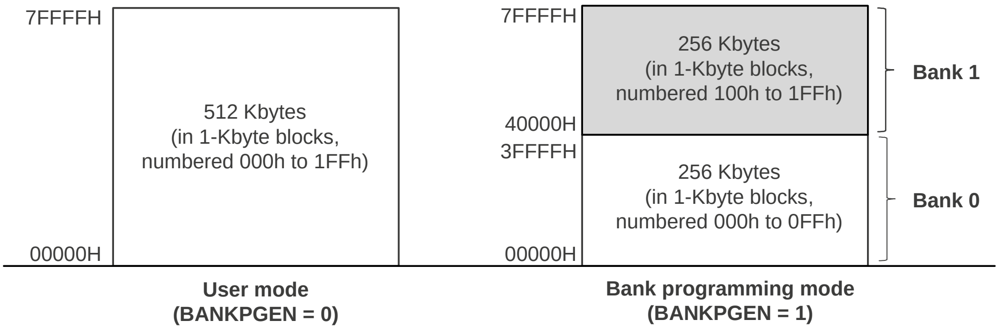
38.6.2 Bank programming function

Bank programming enables updating of the user program while it is running.

38.6.2.1 Switching bank modes

The RL78/I1C (512 KB) has a function for switching between two bank modes, user and bank programming. In user mode, the user area in the code flash memory is handled as a single area. In bank programming mode, on the other hand, the user area is divided into two banks. While the user program is running from one bank, you can update the program in the other bank. **Figure 38-9** shows a schematic view of switching of the bank modes. The setting of the BANKPGEN bit in the flash operating mode select register (FLMODE) determines the bank mode. Selecting the bank programming mode enables bank programming.

Figure 38-9. Switching Bank Modes

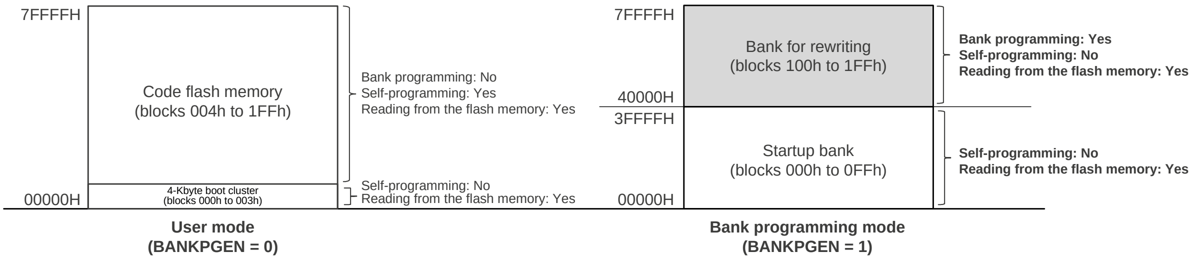


38.6.2.2 Bank programming

Selecting the bank programming mode (BANKPGEN = 1) enables updating (erasing and programming) and verifying of the user program in the bank for rewriting along with reading from all areas of the code flash memory while the user program is running from the startup bank. Self-programming is prohibited in this mode.

Selecting the user mode (BANKPGEN = 0) prevents bank programming but still allows reading from all areas of the code flash memory.

Figure 38-10. Bank Programming



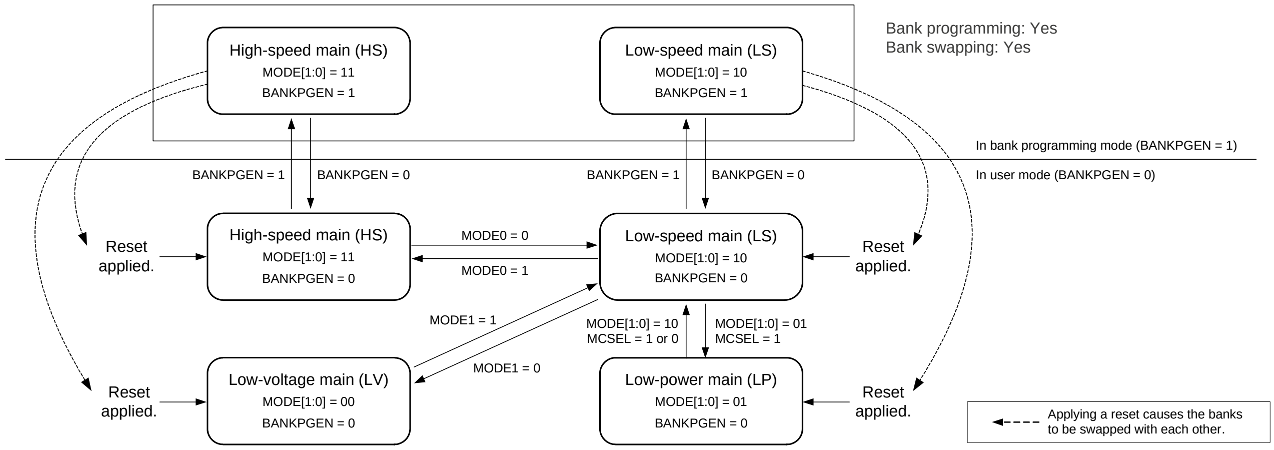
(**Cautions** are listed on the next page.)

Cautions 1. Start bank programming after having confirmed that the mode is high-speed main (HS) or low-speed main (LS) and used the voltage detector (LVD) to confirm that V_{DD} is no less than 2.7 V. Bank programming is not available in low-voltage main (LV) or low-power main (LP) mode.

The flash operation mode must be changed from low-voltage main (LV) or low-power main (LP) to low-speed main (LS) before bank programming and bank swapping can be switched on.

Figure 38-11 shows the operation mode transitions of the flash memory in relation to bank programming mode.

Figure 38-11. State Transitions of the Flash Memory in Relation to Bank Programming



2. Bank programming cannot be used at the same time as background operation (BGO) of the data flash memory.

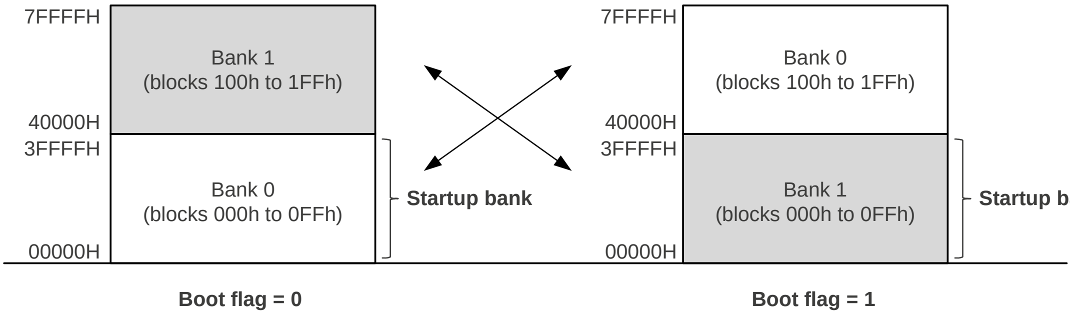
38.6.2.3 Bank swapping

As a countermeasure for the unintended interruption of rewriting due to a reset or for some other reason, bank swapping provides a way to safely update programs in bank programming mode (BANKPGEN = 1).

Figure 38-12 is a schematic view of bank swapping. After having updated the program in bank programming mode (BANKPGEN = 1), place the flash memory in user mode (BANKPGEN = 0). After that, invert the value of the boot flag of the flash memory by using the given function of the flash self-programming library (FSL) and apply a reset. This switches the address ranges of banks 0 and 1 so that booting up will be of the program in the updated area.

Caution The value of the boot flag cannot be changed in bank programming mode (BANKPGEN = 1). Be sure to do this in user mode (BANKPGEN = 0).

Figure 38-12. Bank Swapping (User Mode and Bank Programming Mode)



38.6.2.4 Flash operating mode select register (FLMODE)

The FLMODE register is an 8-bit register used to control flash operation modes and operation of the code flash memory.

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 while the setting of the FLMWEN bit of the flash operating mode protect register (FLMWRP) is 0.

Reset generation updates MODE1 and MODE0 with the set value of CMODE1 and CMODE0 in the option byte (address: 000C2H).

Figure 38-13. Format of Flash Operating Mode Select Register (FLMODE)

Address: F00AAH After reset: 00H/80H/C0H[Note] 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 and bank programming modes
0	User mode Bank programming is not possible.
1	Bank programming mode Bank programming is possible.

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

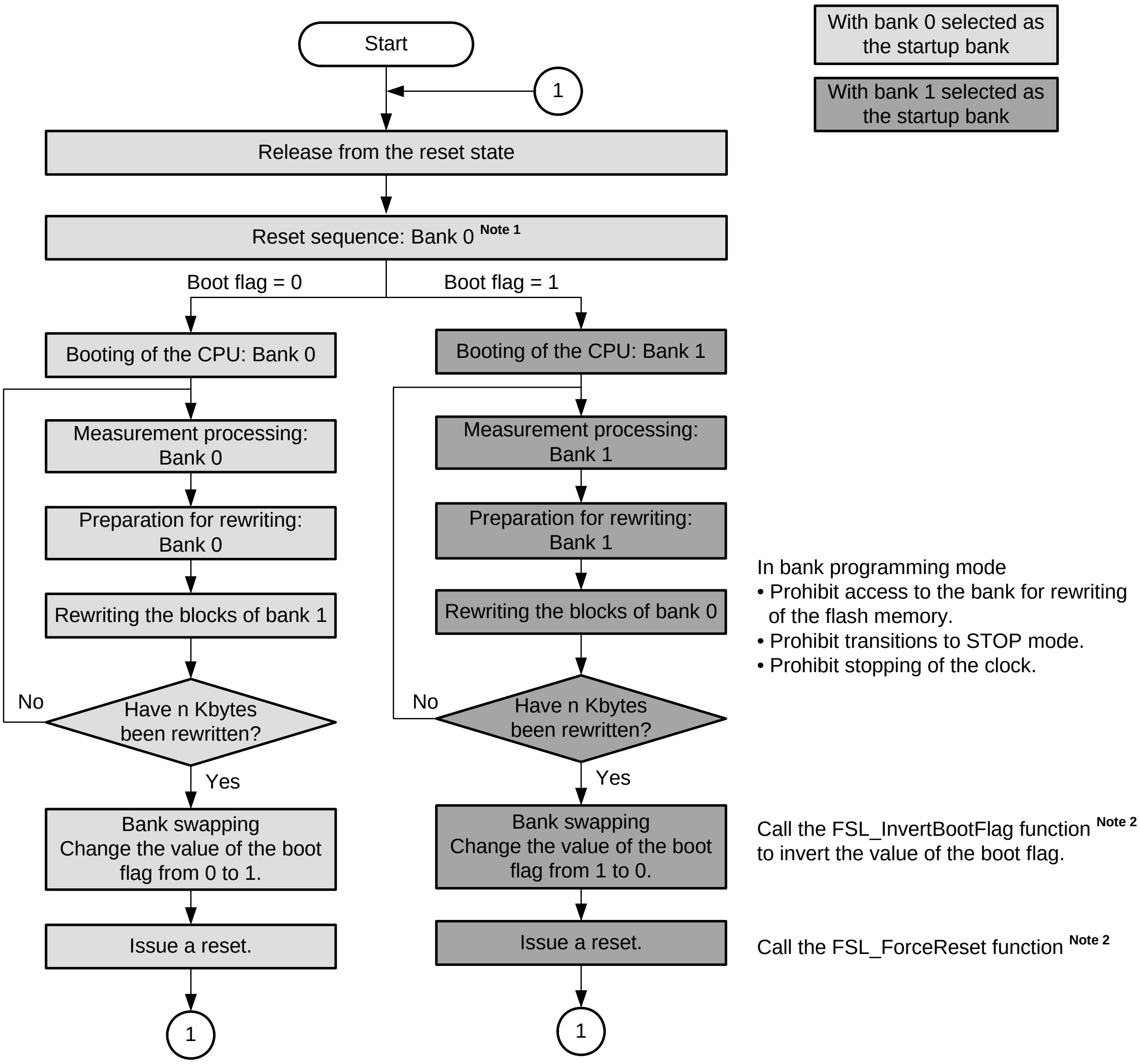
Caution The value of the FLMODE register cannot be changed while the setting of the FLMWEN bit of the flash operating mode protect register (FLMWRP) is 0. If the current setting of the FLMWEN bit is 0, set it to 1 before writing to the BANKPGEN bit.

In addition, restore the setting of the FLMWEN bit to 0 after having written to the BANKPGEN bit.

38.6.2.5 Procedure for updating a program

The flowchart below shows the procedure for using bank programming and bank swapping to update the user program while the original user program is running.

Figure 38-14. Procedure for Rewriting the Program


With bank 0 selected as the startup bank
With bank 1 selected as the startup bank
Start
1
Release from the reset state
Reset sequence: Bank 0 Note 1
Boot flag = 0
Boot flag = 1
Booting of the CPU: Bank 0
Booting of the CPU: Bank 1
Measurement processing: Bank 0
Measurement processing: Bank 1
Preparation for rewriting: Bank 0
Preparation for rewriting: Bank 1
Rewriting the blocks of bank 1
Rewriting the blocks of bank 0
No
Have n Kbytes been rewritten?
Yes
Bank swapping Change the value of the boot flag from 0 to 1.
Bank swapping Change the value of the boot flag from 1 to 0.
Issue a reset.
1
In bank programming mode
• Prohibit access to the bank for rewriting of the flash memory.
• Prohibit transitions to STOP mode.
• Prohibit stopping of the clock.
Call the FSL_InvertBootFlag function Note 2 to invert the value of the boot flag.
Call the FSL_ForceReset function Note 2

Notes 1. Since the option bytes are read from bank 0, set the same values in the option byte areas in both banks 0 and 1.

2. For details on the functions for use with the flash memory, refer to the **RL78 Family Flash Self-Programming Library Type 01 User's Manual (R01US0050)**.

Caution When bank 1 is selected as the startup bank (BANKPGEN = 1, boot flag = 1), block 100H at addresses from 40000H to 403FFH, to which the option bytes are allocated, must not be rewritten.
An attempt to do this may prevent correct booting up of the device. Specifically, if rewriting of bank 0 fails while in progress due to the supply of power being shut off, the generation of a reset due to an external factor, or for some other reason, the corrupted option bytes in bank 0 will be read in the reset sequence following rewriting, so correct booting up of the device will not be possible.

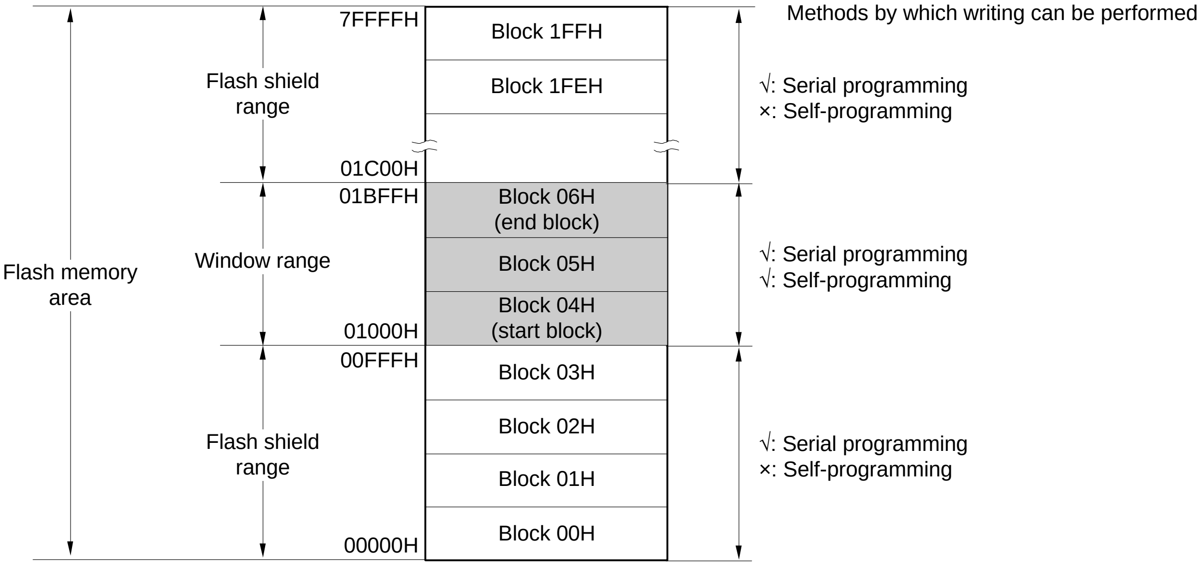
38.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 38-15. Flash Shield Window Setting Example
(Target Devices: R5F10NPL, R5F10NML, Start Block: 04H, End Block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 38-11. Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming Conditions	Window Range Setting/ Change Methods	Execution Commands	
		Block Erase	Write
Self-programming	Specify the starting and ending blocks by the flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **38.7 Security Settings** to prohibit writing/erasing during serial programming.

38.7 Security Settings

The RL78 microcontroller supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command.

- Disabling block erase

  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write

  Execution of the write command for entire blocks in the flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.

  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0

  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.

Table 38-12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

Caution The security function of the flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **38.6.3** for detail).

Table 38-12. Relationship Between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note 1
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0 Note 2	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0 Note 2	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Notes 1. Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

2. When bank programming mode is in use, the setting to prohibit the rewriting of boot cluster 0 cannot be made. Making the setting prohibits the execution of bank programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **38.6.3** for detail).

Table 38-13. Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0 Note		Cannot be disabled after set.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0 Note		Cannot be disabled after set.

Note When bank programming mode is in use, the setting to prohibit the rewriting of boot cluster 0 cannot be made. Making the setting prohibits the execution of bank programming.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

38.8 Data Flash

38.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the flash data library. For details, refer to **RL78 Family Flash Data Library User's Manual**.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1 KB) units.
- The data flash can be accessed only in 8-bit units.
- The data flash 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).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Bank programming in bank programming mode cannot be used at the same time as background operation (BGO).

Cautions 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.

Remark For the flash programming mode, see **38.6 Self-Programming**.

38.8.2 Register controlling data flash memory

38.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 38-16. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

38.8.3 Procedure for accessing data flash memory

The data flash memory is initially stopped after a reset ends and cannot be accessed (read or programmed). To access the memory, perform the following procedure:

<1> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).

<2> Wait for the setup to finish for software timer, etc.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>

- HS (high-speed main) mode: 5 μs
- LS (low-speed main) mode: 720 ns
- LP (low-power main) mode: 720 ns
- LV (low-voltage main) mode: 10 μs

<3> After the wait, the data flash memory can be accessed.

Cautions 1. Accessing the data flash memory is not possible during the setup time.

2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.

3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.

4. Once the data flash memory is read while the subsystem clock is selected as the CPU/peripheral hardware clock (CLS = 1), follow the procedure listed as steps (1) to (3) below, in that order, to read the data flash area after switching the CPU/peripheral hardware clock from the subsystem clock to the main system clock.

(1) Make sure the main system clock is selected as the CPU/peripheral hardware clock (CLS = 0).

(2) Read data from any location in the data flash area. The value read at this point is undefined.

(3) Wait for the time listed below according to the operating mode, then read data from the desired parts of the data flash area.

HS (high-speed main) mode: 5 μs
LS (low-speed main) mode: 1 μs
LV (low-voltage main) mode: 10 μs
LP (low-power main) mode: 1 μs

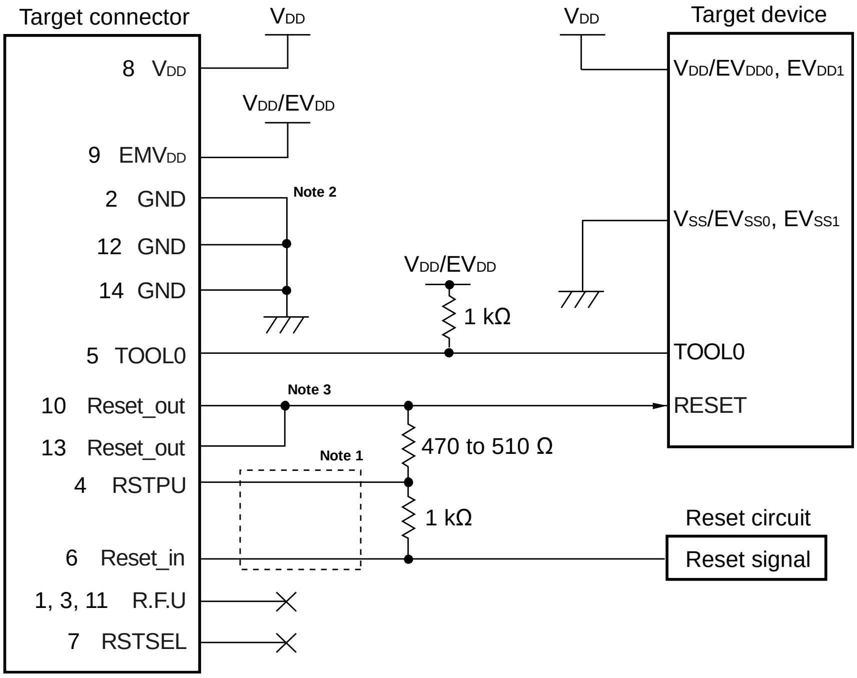
CHAPTER 39 ON-CHIP DEBUG FUNCTION

39.1 Connecting E1, E2, E2 Lite, E20 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{\text{RESET}}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1, E2, E2 Lite, E20 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 39-1. Example of Connections with the E1 On-chip Debugging Emulator



Notes 1. These connections must be made if the on-chip debugging function is to be used. Flash programming can proceed whether or not these connections are made.

2. Be sure to connect pins 2, 12, and 14 of the E1 emulator to the ground pins on the user system.
As well as being used as electrical ground pins, these pins are used for monitoring connection of an E1 emulator to the user system.

3. Pins 10 and 13 of the E1 emulator must be connected.

Caution The values in the connection example are for reference. Before proceeding with flash programming for mass production, sufficiently evaluate the values in use to confirm that they satisfy the specifications of the target device.

Remark With products not provided with an EV_{DD1} or EV_{SS1} pin, replace EV_{DD1} with V_{DD} or EV_{DD0}, or replace EV_{SS1} with V_{SS} or EV_{SS0}.

39.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 37 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When bank swapping is to be used during self-programming, set the same values at 400C3H and 400C4H to 400CDH as those at 000C3H and 000C4H to 000CDH in advance, because switching between 000C3H and 000C4H to 000CDH and 400C3H and 400C4H to 400CDH will proceed.

Table 39-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes[Note]
400C4H to 400CDH	

Note The setting FFFFFFFFFFFFFFFFFFFFH for the ID code is not possible.

39.3 Securing of User Resources

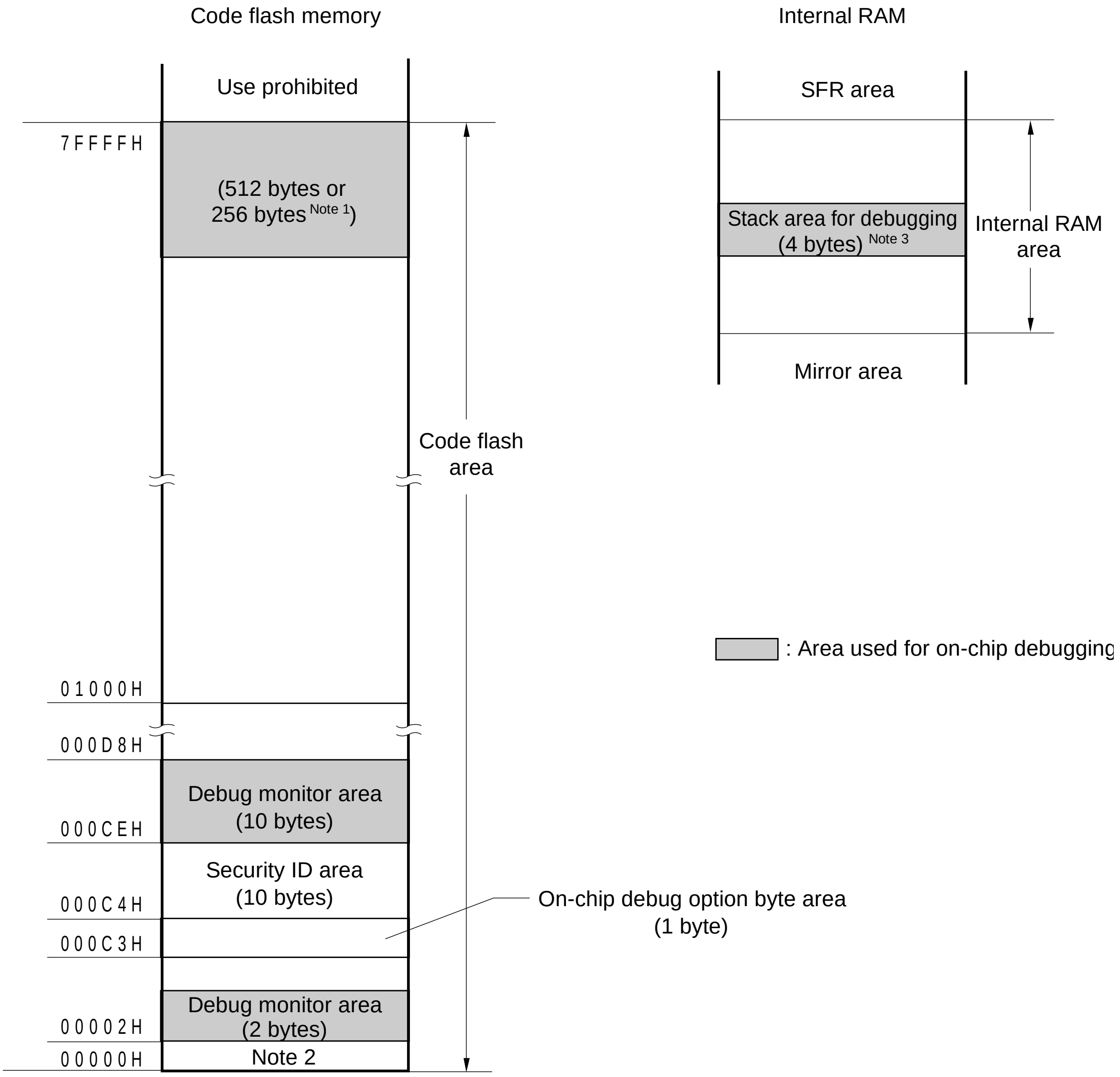
To perform communication between the RL78 microcontroller and E1, E2, E2 Lite, E20 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in **Figure 39-2** are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 39-2. Memory Spaces Where Debug Monitor Programs Are Allocated


Code flash memory
Use prohibited
7FFFFH
(512 bytes or 256 bytes Note 1)
Code flash area
01000H
000D8H
Debug monitor area (10 bytes)
000CEH
Security ID area (10 bytes)
000C4H
000C3H
On-chip debug option byte area (1 byte)
Debug monitor area (2 bytes)
00002H
Note 2
00000H
Internal RAM
SFR area
Stack area for debugging (4 bytes) Note 3
Internal RAM area
Mirror area
: Area used for on-chip debugging

Notes 1. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

39.4 On-chip Debug Function in Bank Programming Mode

The on-chip debug function cannot be used in bank programming mode (BANKPGEN = 1). For more on bank programming mode, see **38.6.2 Bank programming function**.

CHAPTER 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 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

40.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 40-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined 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: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code 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 value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 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	–

Examples 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	–

Examples 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: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 – 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

CHAPTER 41 32-BIT MULTIPLY-ACCUMULATOR

41.1 Function of 32-bit Multiply-accumulator

The 32-bit multiply-accumulator has the following function.

- 32 bits × 32 bits = 64 bits (Unsigned multiply)
- 32 bits × 32 bits = 64 bits (Signed multiply)
- 32 bits × 32 bits + 64 bits = 64 bits (Unsigned multiply-accumulate)
- 32 bits × 32 bits + 64 bits = 64 bits (Signed multiply-accumulate)
- Fixed point mode supported
- Interrupt output for the multiply-accumulation result in the case of overflow/underflow
- The results of multiply-and-accumulate operations (cumulative values) can be retained in any of 24 selectable buffer channels.

The data format is 2’s complement form.

- Signed (The highest-order bit is the signed bit.)
  7FFF (hexadecimal digit) = 32767 (decimal digit)
  0001 (hexadecimal digit) = 1 (decimal digit)
  0000 (hexadecimal digit) = 0 (decimal digit)
  FFFF (hexadecimal digit) = –1 (decimal digit)
  8000 (hexadecimal digit) = -32768 (decimal digit)
- Unsigned
  FFFF (hexadecimal digit) = 65535 (decimal digit)
  0000 (hexadecimal digit) = 0 (decimal digit)

41.2 Configuration of 32-bit Multiply-accumulator

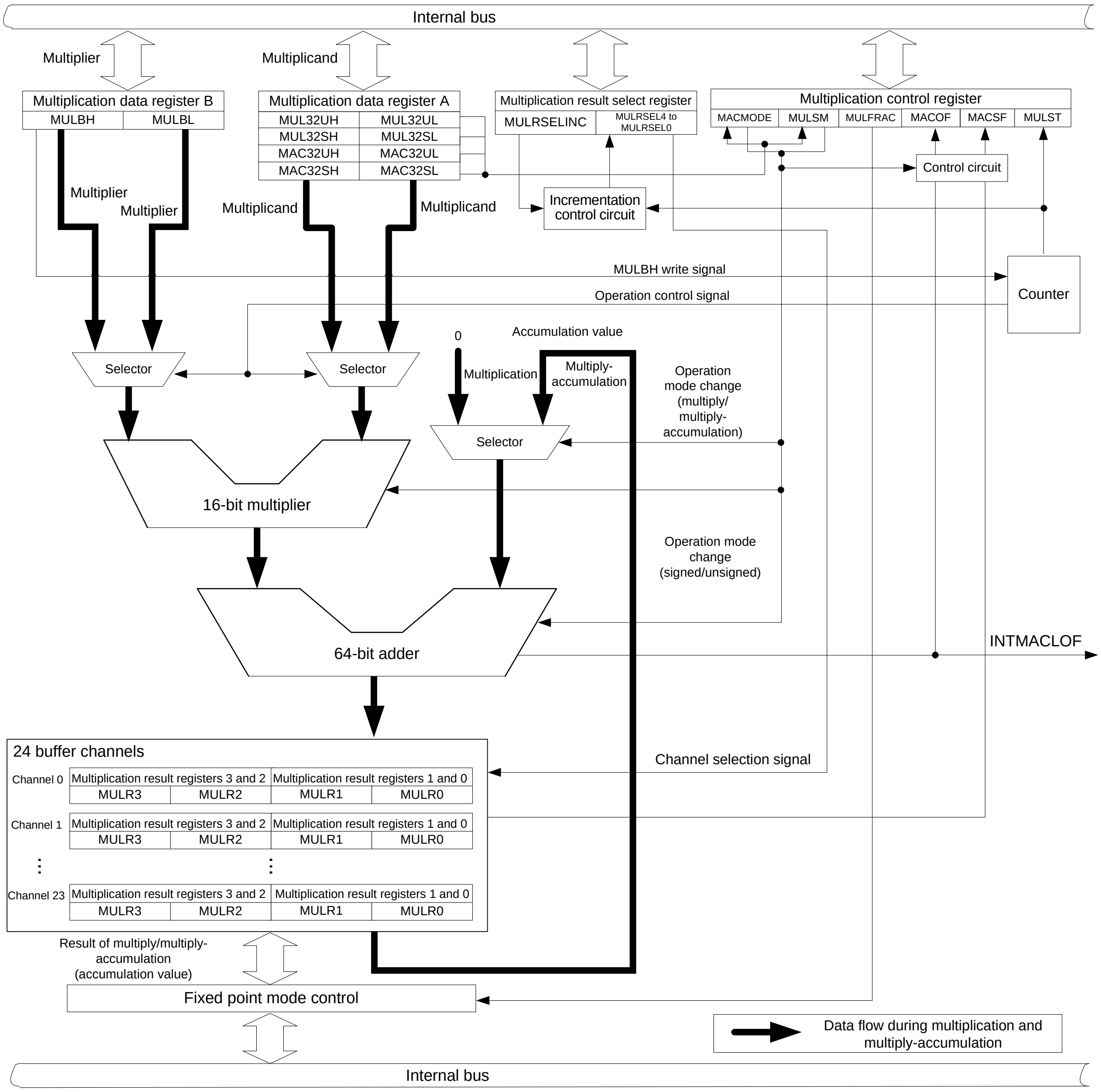
The 32-bit multiply-accumulator consists of the following hardware.

Table 41-1. Configuration of 32-bit Multiply-accumulator

Item	Configuration
Registers	Multiplication data register B (L) (MULBL) Multiplication data register B (H) (MULBH) Multiplication data register A (L) (Unsigned) (MUL32UL) Multiplication data register A (H) (Unsigned) (MUL32UH) Multiplication data register A (L) (Signed) (MUL32SL) Multiplication data register A (H) (Signed) (MUL32SH) Multiply-accumulation data register A (L) (Unsigned) (MAC32UL) Multiply-accumulation data register A (H) (Unsigned) (MAC32UH) Multiply-accumulation data register A (L) (Signed) (MAC32SL) Multiply-accumulation data register A (H) (Signed) (MAC32SH) Multiplication result register 0 (MULR0) Multiplication result register 1 (MULR1) Multiplication result register 2 (MULR2) Multiplication result register 3 (MULR3) Multiplication control register (MULC) Multiplication result select register (MULRSEL) Peripheral enable register 2 (PER2) Peripheral reset control register 2 (PRR2)

Figure 41-1 shows a block diagram of 32-bit multiply-accumulator.

Figure 41-1. Block Diagram of 32-bit Multiply-accumulator


Internal bus
Multiplier
Multiplicand
Multiplication data register B
MULBH
MULBL
Multiplication data register A
MUL32UH
MUL32UL
MUL32SH
MUL32SL
MAC32UH
MAC32UL
MAC32SH
MAC32SL
Multiplication result select register
MULRSELINC
MULRSEL4 to MULRSEL0
Multiplication control register
MACMODE
MULSM
MULFRAC
MACOF
MACSF
MULST
Control circuit
Multiplier
Multiplier
Multiplicand
Multiplicand
Incrementation control circuit
MULBH write signal
Counter
Operation control signal
Selector
Selector
0
Accumulation value
Multiplication
Multiply-accumulation
Operation mode change (multiply/ multiply-accumulation)
Selector
16-bit multiplier
Operation mode change (signed/unsigned)
64-bit adder
INTMACLOF
24 buffer channels
Channel 0
Multiplication result registers 3 and 2
Multiplication result registers 1 and 0
MULR3
MULR2
MULR1
MULR0
Channel 1
Multiplication result registers 3 and 2
Multiplication result registers 1 and 0
MULR3
MULR2
MULR1
MULR0
Channel 23
Multiplication result registers 3 and 2
Multiplication result registers 1 and 0
MULR3
MULR2
MULR1
MULR0
Channel selection signal
Result of multiply/multiply-accumulation (accumulation value)
Fixed point mode control
Data flow during multiplication and multiply-accumulation
Internal bus

41.2.1 Multiplication data register A (MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL)

Multiplication data register A specifies the multiplicand used for multiplication and multiply-accumulation.

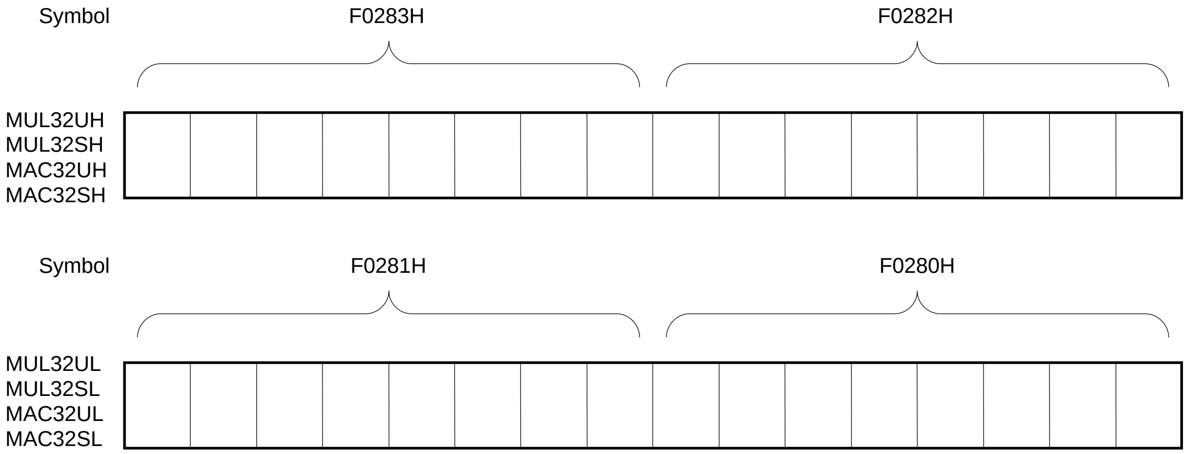
Multiplication data register A can be set by a 16-bit manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 41-2. Format of Multiplication Data Register A

(MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL)

Address: F0282H (MUL32UH), F0286H (MUL32SH), F028AH (MAC32UH), F028EH (MAC32SH)
F0280H (MUL32UL), F0284H (MUL32SL), F0288H (MAC32UL), F028CH (MAC32SL)
After reset: 0000H R/W



Caution Do not rewrite the values of the multiplication data register A (MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL) during the operation processing (when the bit 0 (MULST) of the multiplication control register (MULC) = 1). If this is done, the operation result will be an undefined value.

Multiplication data register A stores the written value. The MACMODE and MULSM bits of the MULC register are also rewritten with the values of the supported operation mode.

For the multiplication data register A, the operation mode can be switched by the register that specifies the multiplicand since the different register name and register address are set for each operation mode. The MACMODE and MULSM bits of the MULC register are also rewritten with the values of the supported operation mode.

For the MUL32UL, MUL32UH, MUL32SL, MUL32SH, MAC32UL, MAC32UH, MAC32SL, and MAC32SH registers, all the register values are rewritten when rewriting one register value since a common register is used for these registers.

The following table shows the relationship between the operation mode and register name.

Table 41-2. Relationship between Operation Mode and Register Name

Operation Mode	Register Name of Multiplication Data Register A	
	Bits 31 to 16 (MULAH)	Bits 15 to 0 (MULAL)
Multiplication mode (unsigned)	MUL32UH	MUL32UL
Multiplication mode (signed)	MUL32SH	MUL32SL
Multiply-accumulation mode (unsigned)	MAC32UH	MAC32UL
Multiply-accumulation mode (signed)	MAC32SH	MAC32SL

41.2.2 Multiplication data register B (MULBL, MULBH)

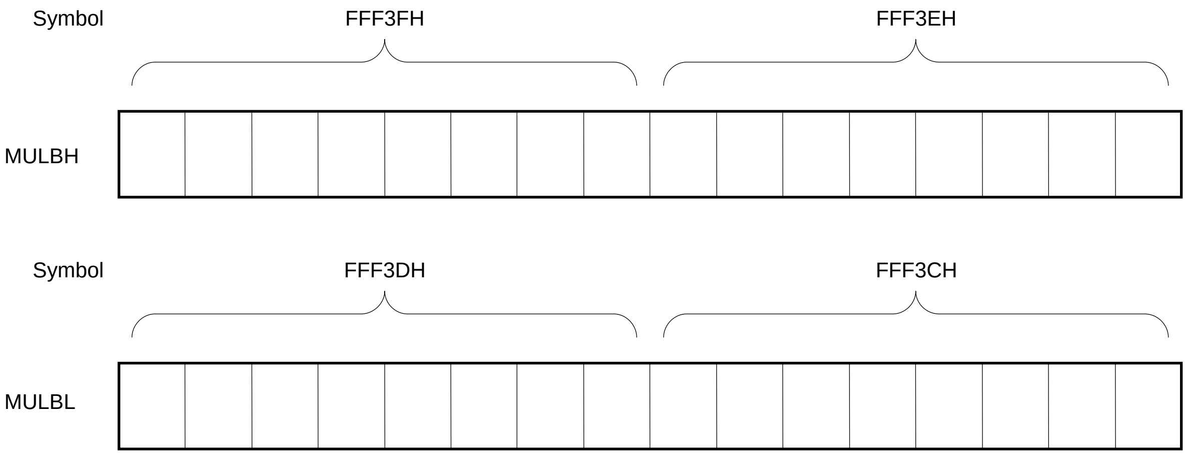
Multiplication data register B specifies the multiplier used for multiplication and multiply-accumulation.

Multiplication data register B can be set by a 16-bit manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 41-3. Format of Multiplication Data Register B (MULBH, MULBL)

Address: FFF3EH (MULBH), FFF3CH (MULBL) After reset: 0000H R/W



Cautions 1. Do not rewrite the values of the multiplication data register B with software during the operation (when bit 0 (MULST) of the multiplication control register (MULC) = 1). If this is done, the operation result will be an undefined value.

2. The operation starts by writing to the higher 16 bit (MULBH) of the multiplication data register B. Be sure to set the multiplier in the order from MULBL to MULBH. (The operation processing exits after the 5th cycle from writing of the MULBH register.)

The multiplication data register B is used as a common register in all the operation modes.

The following table shows the relationship between the operation mode and register name.

Table 41-3. Relationship between Operation Mode and Register Name

Operation Mode	Register Name of Multiplication Data Register B	
	Bits 31 to 16 (MULBH)	Bits 15 to 0 (MULBL)
Multiplication mode (unsigned)	MULBH	MULBL
Multiplication mode (signed)		
Multiply-accumulation mode (unsigned)		
Multiply-accumulation mode (signed)		

41.2.3 Multiplication result registers (MULR0, MULR1, MULR2, MULR3)

The multiplication result registers hold the results of operations.

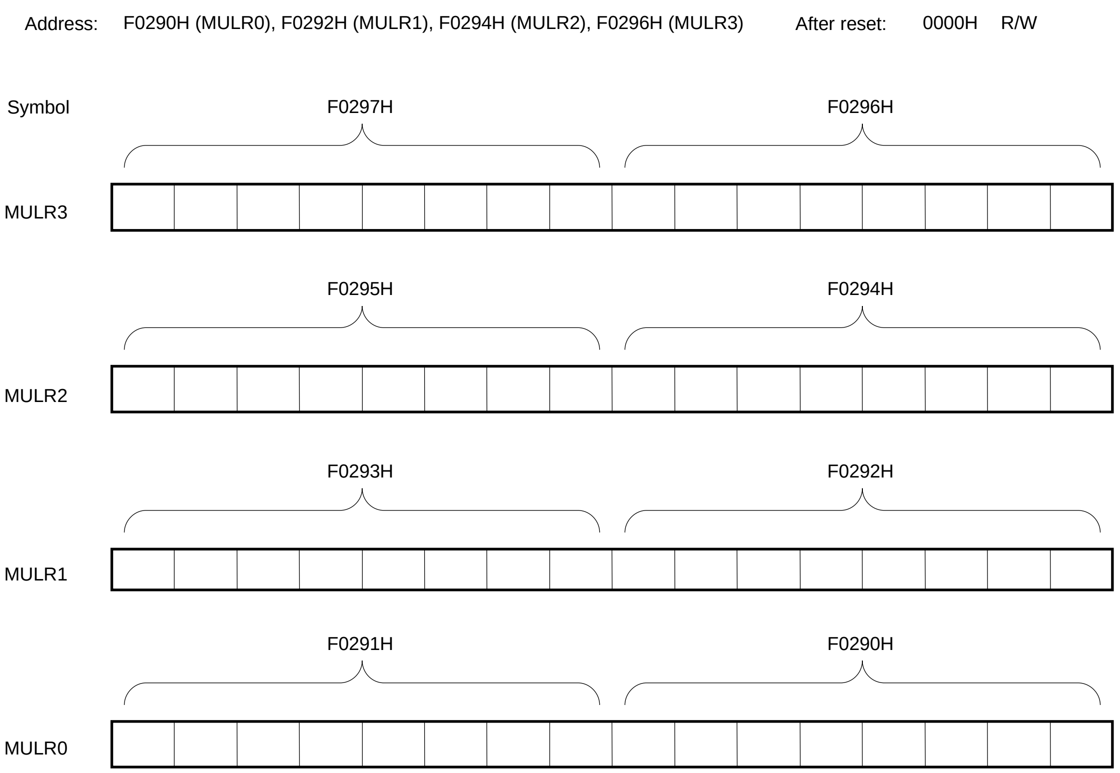
The sum-of-products block has a total of 24 buffer register channels, which can hold 24 results of operations.

The current set of multiplication result registers can be selected by the multiplication result select register (MULRSEL). The current value of the selected multiplication result registers is used as the initial value in accumulation of the results of multiplication, and the cumulative results of multiplication are stored in the multiplication result registers of the selected channel. The multiplication result registers of the selected channel can be read or written by access to the MULR0 to MULR3 registers.

The multiplication result registers can be set by 16-bit manipulation instructions.

Reset signal generation clears each of the registers to 0000H. The 24 multiplication result register channels can be cleared by selecting the channels by software and writing to clear them or by setting the MACRES bit of peripheral reset control register 2 (PRR2) to 1.

Figure 41-4. Format of Multiplication Result Registers (MULR0, MULR1, MULR2, MULR3)


Address: F0290H (MULR0), F0292H (MULR1), F0294H (MULR2), F0296H (MULR3) After reset: 0000H R/W
Symbol
F0297H
F0296H
MULR3
F0295H
F0294H
MULR2
F0293H
F0292H
MULR1
F0291H
F0290H
MULR0

Cautions 1. Do not rewrite the value of the multiplication result register during the operation processing (when bit 0 (MULST) of the multiplication control register (MULC) = 1). If this is done, the operation result will be an undefined value.

2. When the value of the multiplication result register is read out during the operation processing (MULST = 1), the value is not guaranteed. The values of the MULR3 to MULR0 registers can be read out even during the operation processing if the number of clocks required for the operation (refer to 41.4.2 Number of clocks for result availability) has been met in each multiplication result register since the operation results are stored in the MULR0, MULR1, MULR2, and MULR3 registers in this order (results are stored in the MULR2 and MULR3 registers simultaneously).

The multiplication result register is used as a common register in all the operation modes.

The following table shows the relationship between the operation mode and register name.

Table 41-4. Relationship between Operation Mode and Register Name

Operation Mode	Multiplication Result Register Name			
	Bits 63 to 48	Bits 47 to 32	Bits 31 to 16	Bits 15 to 0
Multiplication mode (unsigned)	MULR3	MULR2	MULR1	MULR0
Multiplication mode (signed)				
Multiply-accumulation mode (unsigned)				
Multiply-accumulation mode (signed)				

The operation result (multiplication) is stored for the multiplication, and the operation result (accumulation) is stored for the multiply-accumulation. Additionally, the accumulation initial value can be set for the multiply-accumulation.

Table 41-5. Details of Storing of Operation Modes and Multiplication Result Registers

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned)	–	MULR3 to MULR0: Multiplication (unsigned)Note
Multiplication mode (signed)	–	MULR3 to MULR0: Multiplication (signed)Note
Multiply-accumulation mode (unsigned)	MULR3 to MULR0: Accumulation initial value (unsigned)Note	MULR3 to MULR0: Accumulation value (unsigned)Note
Multiply-accumulation mode (signed)	MULR3 to MULR0: Accumulation initial value (signed)Note	MULR3 to MULR0: Accumulation value (signed)Note

Note The channel selected by the multiplication result select register (MULRSEL) is used.

Cautions 1. Ensure that software does not overwrite the value of the MULRSEL register while an operation is in progress. If this is done, the result of the operation is undefined as is the value in the multiplication result registers selected before and after the change.

2. If exceeding the maximum value of the value range that can be handled in 64 bit (= overflow), or if dropping below the minimum value (= underflow), the value is reversed, and the values plus overflow/underflow values are stored in the MULR3 to MULR0 registers.

■ Unsigned

- In case of overflow

  Processing) 2^{64} + MULR[63:0]

  Example)

  FFFF FFFF FFFF FFFFh + 0000 0000 0000 0001h = 0000 0000 0000 0000h

■ Signed

- In case of overflow

  Processing) 2^{63} + MULR[62:0]

  Example)

  7FFF FFFF FFFF FFFFh + 0000 0000 0000 0001h = 8000 0000 0000 0000h

- In case of underflow

  Processing) -2^{63} + MULR[62:0]

  Example)

  8000 0000 0000 0000h + FFFF FFFF FFFF FFFFh = 7FFF FFFF FFFF FFFFh

41.3 Register Controlling 32-bit Multiply-accumulator

32-bit multiply-accumulator is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Multiplication control register (MULC)
- Multiplication result select register (MULRSEL)

41.3.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise.

When the 32-bit multiply-accumulator is to be used, be sure to set bit 2 (MACEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 41-5. Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER2	TMKAEN	OSDCEN	UARTMG1EN	UARTMG0EN	0	MACEN	0	VRTCEN

MACEN	Control of 32-bit multiplier and accumulator input clock supply
0	Stops input clock supply. • SFR used by the 32-bit multiplier and accumulator cannot be written. The read value is 00H. However, the SFR is not initialized. [Note]
1	Enables input clock supply. • SFR used by the 32-bit multiplier and accumulator can be read and written.

Note To initialize the 32-bit multiply-accumulator and the SFR used by the 32-bit multiply-accumulator, use bit 2 (MACRES) of PRR2.

41.3.2 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.

To place the 32-bit multiply-accumulator in the reset state, be sure to set bit 2 (MACRES) to 1.

The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 41-6. Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	0
PRR2	TMKARES	OSDCRES	UARTMG1RES	UARTMG0RES	0	MACRES	0	0

MACRES	Control resetting of the 32-bit multiply-accumulator
0	The 32-bit multiply-accumulator is released from the reset state.
1	The 32-bit multiply-accumulator is in the reset state.

41.3.3 Multiplication control register (MULC)

MULC register is the 8-bit register that controls the operation of the 32-bit multiply-accumulator.

MULC register is set by 1-bit or 8-bit memory manipulation instruction.

However, for the overflow/underflow flag (MACOF) of the multiply-accumulation result (accumulation value), sign flag (MACSF) of the multiply-accumulation result (accumulation value), and the operation status flag (MULST), only the read-out operation is enabled.

Reset signal generation clears this register to 00H.

Figure 41-7. Format of Multiplication Control Register (MULC) (1/2)

Address: F029AH After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	<4>	3	<2>	<1>	<0>
MULC	MACMODE	MULSM	0	MULFRAC	0	MACOF	MACSF	MULST

<table>
<tr><th>MACMODE</th><th>MULSM</th><th>Selection of operation mode</th></tr>
<tr><td>0</td><td>0</td><td>Multiplication mode (unsigned) (default)</td></tr>
<tr><td>0</td><td>1</td><td>Multiplication mode (signed)</td></tr>
<tr><td>1</td><td>0</td><td>Multiply-accumulation mode (unsigned)</td></tr>
<tr><td>1</td><td>1</td><td>Multiply-accumulation mode (signed)</td></tr>
<tr><td colspan="3">The operation mode is automatically switched by the address of the multiplication data register A to be specified.
Reading out this bit enables to check the operation mode.
Writing in this bit enables to specify the operation mode as well.</td></tr>
</table>

<table>
<tr><th>MULFRAC</th><th>Selection of fixed point mode</th></tr>
<tr><td>0</td><td>Disabled</td></tr>
<tr><td>1</td><td>Enabled</td></tr>
<tr><td colspan="2">For details of the fixed point mode, refer to <b>41.4.6 Fixed point mode</b>.</td></tr>
</table>

<table>
<tr><th>MACOF</th><th>Overflow/underflow flag of multiply-accumulation result (accumulation value)</th></tr>
<tr><td>0</td><td>No overflow/underflow occurred.</td></tr>
<tr><td>1</td><td>Overflow/underflow occurred.</td></tr>
<tr><td colspan="2">[Conditions for setting]
● Multiply-accumulation mode (unsigned)
If exceeding the accumulation value FFFF FFFF FFFF FFFFh
● Multiply-accumulation mode (signed)
If the result is the negative value exceeding 7FFF FFFF FFFF FFFFh after adding the positive multiplication value to the positive accumulation value
If the result is the positive value exceeding 8000 0000 0000 0000h after adding the negative multiplication value to the negative accumulation value
[Timing of setting/clearing]
● At finish of operation (MULST = 1 → 0)</td></tr>
</table>

Figure 41-7. Format of Multiplication Control Register (MULC) (2/2)

Address: F029AH After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	<4>	3	<2>	<1>	<0>
MULC	MACMODE	MULSM	0	MULFRAC	0	MACOF	MACSF	MULST

MACSF	Sign flag of multiply-accumulation result (accumulation value)
0	Positive accumulation value
1	Negative accumulation value
Multiply-accumulation mode (unsigned): Always 0 Multiply-accumulation mode (signed): Displays the sign bit of the accumulation value.	

MULST	Operation processing status bit
0	Completion of operation processing
1	During operation processing
The operation starts by writing to the higher 16 bit (MULBH) of the multiplication data register B. The MULST bit is set (1) at start of the operation, and cleared (0) at completion of 5 cycles.	

Note Bits 0 to 2 are Read only.

Caution **Do not rewrite the value of the multiplication control register (MULC) during the operation processing (MULST = 1). Otherwise, the MACOF and MACSF bits of the multiplication result register will be undefined values.**

41.3.4 Multiplication result select register (MULRSEL)

MULRSEL is an 8-bit register that selects the current set of multiplication result registers (MULR3 to MULR0). A total of 24 multiplication result register channels are mounted in the sum-of-products block.

Set the MULRSEL[4:0] bits before the operation starts to select the channel of multiplication result register (the current set of MULR3 to MULR0).

The MULRSEL register can be set by 1-bit or 8-bit memory manipulation instructions.

Reset signal generation clears this register to 00H.

Figure 41-8. Format of Multiplication Result Select Register (MULRSEL) (1/2)

Address: F029BH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
MULRSEL	MULRSELINC	0	0	MULRSEL[4:0]				

MULRSELINC	Automatic incrementation of the multiplication result selection bits (MULRSEL[4:0])
0	Disabled
1	Enabled
When MULRSELINC is set to 1, the value of MULRSEL[4:0] is automatically incremented after the operation is completed. Channel 0 is selected following channel 23.	

(**Caution** is listed on the next page.)

Figure 41-8. Format of Multiplication Result Select Register (MULRSEL) (2/2)

Address: F029BH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
MULRSEL	MULRSELINC	0	0	MULRSEL[4:0]				

MULRSEL[4:0]					Channel	Register name of the multiplication result registers			
						Bits 63 to 48	Bits 47 to 32	Bits 31 to 16	Bits 15 to 0
0	0	0	0	0	0	MULR3	MULR2	MULR1	MULR0
0	0	0	0	1	1				
0	0	0	1	0	2				
0	0	0	1	1	3				
0	0	1	0	0	4				
0	0	1	0	1	5				
0	0	1	1	0	6				
0	0	1	1	1	7				
0	1	0	0	0	8				
0	1	0	0	1	9				
0	1	0	1	0	10				
0	1	0	1	1	11				
0	1	1	0	0	12				
0	1	1	0	1	13				
0	1	1	1	0	14				
0	1	1	1	1	15				
1	0	0	0	0	16				
1	0	0	0	1	17				
1	0	0	1	0	18				
1	0	0	1	1	19				
1	0	1	0	0	20				
1	0	1	0	1	21				
1	0	1	1	0	22				
1	0	1	1	1	23				
Other than above					Other settings are prohibited (if another setting is made, channel 0 will be selected)				

Caution Ensure that the value of the multiplication result select register (MULRSEL) is not overwritten while an operation is in progress (MULST = 1). If this is done, the values of the multiplication result registers selected before and after the change and of the MACOF and MACSF bits are undefined.

41.4 Operations of 32-bit Multiply-accumulator

41.4.1 Basic operation

The register configuration shows as follow when multiplication or multiply-accumulation is executed.

■ Register configuration during unsigned multiplication

<Multiplier A>		<Multiplier B>		<Product>
32-bit		32-bit		64-bit
Unsigned		Unsigned		Unsigned
[MUL32UH, MUL32UL]	×	[MDBH, MDBL]	=	[MULR3, MULR2, MULR1, MULR0]

■ Register configuration during signed multiplication

<Multiplier A>		<Multiplier B>		<Product>
32-bit		32-bit		64-bit
Signed		Signed		Signed
[MUL32SH, MUL32SL]	×	[MDBH, MDBL]	=	[MULR3, MULR2, MULR1, MULR0]

■ Register configuration during unsigned multiply-accumulation

<Multiplier A>		<Multiplier B>		<Accumulated value>		<Product>
32-bit		32-bit		64-bit		64-bit
Unsigned		Unsigned		Unsigned		Unsigned
[MAC32UH, MAC32UL]	×	[MDBH, MDBL]	+	[MULR3, MULR2, MULR1, MULR0]	=	[MULR3, MULR2, MULR1, MULR0]

■ Register configuration during signed multiply-accumulation

<Multiplier A>		<Multiplier B>		<Accumulated value>		<Product>
32-bit		32-bit		64-bit		64-bit
Signed		Signed		Signed		Signed
[MAC32SH, MAC32SL]	×	[MDBH, MDBL]	+	[MULR3, MULR2, MULR1,MULR0]	=	[MULR3, MULR2, MULR1, MULR0]

41.4.2 Number of clocks for result availability

In case of multiplication or multiply-accumulation, the calculation is started automatically by setting upper 16-bit of the multiplier data (MULBH). **Table 41-6** gives the number of clocks necessary to calculation.

Table 41-6. Number of Clocks Necessary to Calculation

Operation Mode	Operation	The Number of Clocks Necessary to Calculation				
		MULR0	MULR1	MULR2	MULR3	MACOF MACSF
Unsigned multiply	32 bits × 32 bits	2	4	5	5	5
Signed multiply	32 bits × 32 bits	2	4	5	5	5
Unsigned multiply-accumulate	32 bits × 32 bits + 64 bits	2	4	5	5	5
Signed multiply-accumulate	32 bits × 32 bits + 64 bits	2	4	5	5	5

Remark There is no difference in the clock number between the enabled and disabled fixed point modes.

41.4.3 Switch of operation mode

Writing the multiplicand in the multiplication data register A switches the operation mode. MACMODE (bit 7) and MULSM (bit 6) of the multiplication control register (MULC) enable to switch the modes and to check the operation mode. Hold the lastly-written mode so that the multiplication processing can be executed consecutively when writing in the multiplication data register B (H). The initial vale is "Unsigned multiplication mode".

41.4.4 Multiplication operation

- Select a set of multiplication result registers by using the multiplication result select register (MULRSEL) before starting multiplication.
- The multiplication automatically starts when setting the multiplier in the MULBH register. The operation does not start even when setting the MULBL register.
- By setting the multiplicand in the multiplication data register A, the operation mode is switched automatically. However, the operation does not start at this time.
- The MULST bit is set to 1 after starting the operation, and the bit is cleared to 0 after completion of the operation.
- Interrupt does not occur after completion of the operation. [Note]

Note In the case where multiplication mode (signed) is selected and the value of the multiplication result register (MULR3 to MULR0) before the operation is negative, a multiply-accumulation operation overflow/underflow interrupt (INTMACLOF) may be generated. If the multiplication mode (signed) is selected, set a “1” in the interrupt flag mask register (MACMK) to disable INTMACLOF or set the multiplication result register (MULR3) to “0000H” or a positive value before starting the operation by setting the values for multiplication in the MULBH register.

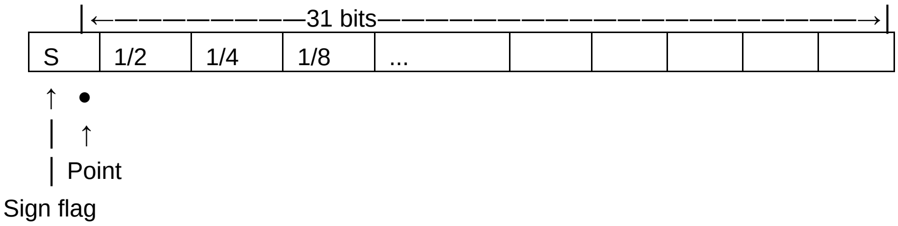
41.4.5 Multiply-accumulation operation

- Select a set of multiplication result registers by using the multiplication result select register (MULRSEL) before starting multiply-accumulation.
- The multiply-accumulation automatically starts when setting the multiplier in the MULBH register. The operation does not start even when setting the MULBL register.
- By setting the multiplicand in the multiplication data register A, the operation mode is switched automatically. However, the operation does not start at this time.
- The MULST bit is set to 1 after starting the operation, and the bit is cleared to 0 after completion of the operation.
- In case of overflow/underflow of the accumulation result after completion of the operation, interrupt is output.

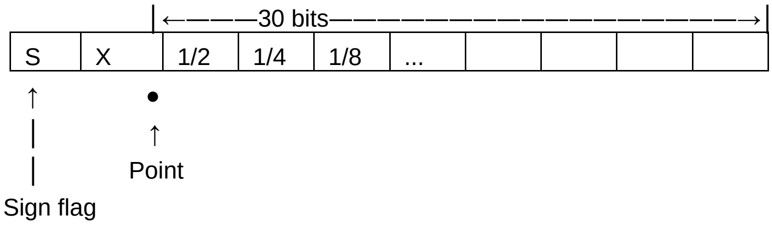
41.4.6 Fixed point mode

Fixed point mode supports the Q format.

(Example) Q31 format –1.0 to 0.999999999534 ≈ 1.0



(Example) Q30 format –2.0 to 1.999999999069 ≈ 2.0



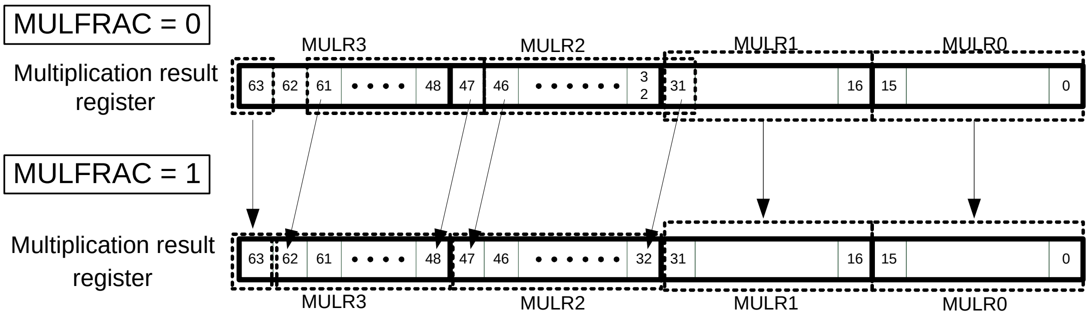
When executing multiplication of two 32-bit fixed points in the Q31 format, the operation result is stored in the multiplication result registers (MULR3 to MULR0) as the Q62 format. To manually covert into the Q31 format, the lower 31 bit of the operation result and extended sign bit must be removed. When the CPU reads the multiplication result registers (MULR3 and MULR2) with the fixed point mode enabled (MULFRAC = 1), the value of the operation result shifted to the left for 1 bit can be read out. By shifting the operation result to the left for 1 bit, the redundant sign bit is automatically removed, and the operation result in the Q31 format can be obtained.

In the fixed point mode, the value of the multiplication result register itself is not rewritten. Therefore, change of the fixed point mode flag value enables to read out both the Q31-format value shifted to the left for 1 bit and the operation result that is not shifted to the left.

41.4.7 Operation of fixed point mode

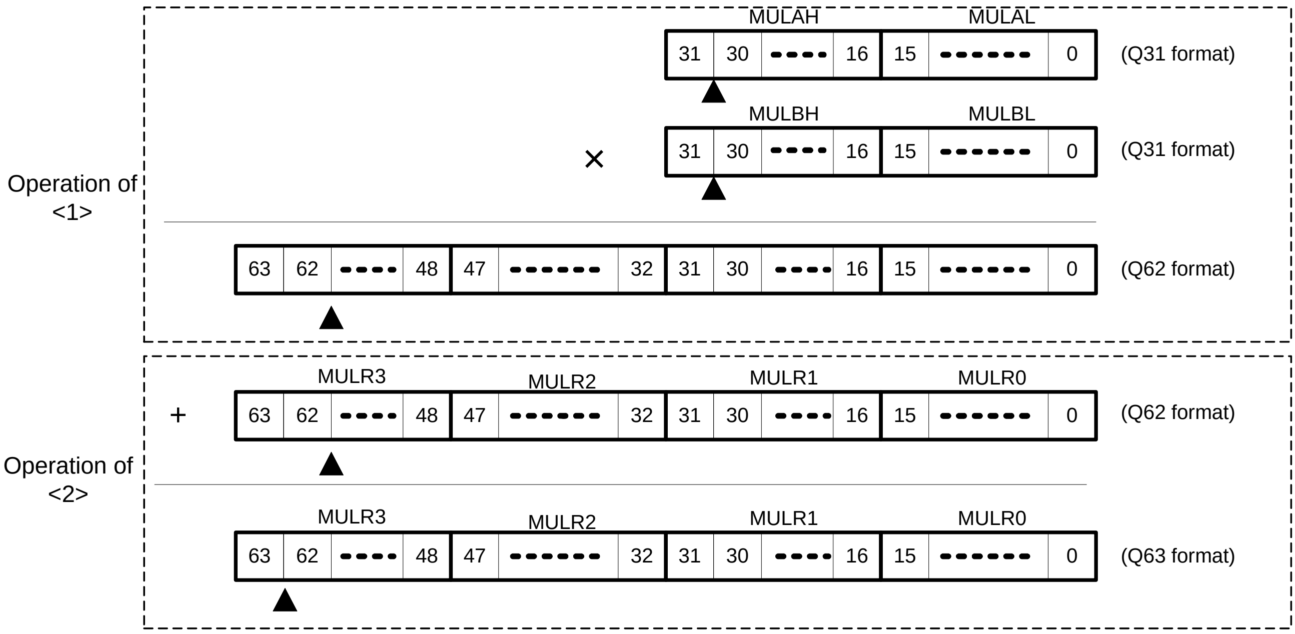
The operation is executed in the same way as the normal operation. The result of reading the multiplication result register only when the fixed point mode is enabled is shown below:

In case of MULFRAC = 1 Value at reading MULR3 = {MULR3[15], MULR3[13:0], MULR2[15]}
Value at reading MULR2 = {MULR2[14:0], MULR1[15]}
Value at reading MULR1 = {MULR1[15:0]}
Value at reading MULR0 = {MULR0[15:0]}



In case of the fixed point mode (MULFRAC = 1), fill MULA and MULB in the Q31 format. For multiply-accumulation, enter the accumulation initial value in the Q62 format (= 31 (31 format of MULA) + 31 (31 format of MULB)). At this time, MULR3 to MULR0 in the Q62 format must be filled since MULR3 to MULR0 must be filled according to the format output from MULA × MULB. An example is shown below:

Example) For multiply-accumulation $\underbrace{\{MULAH, MULAL\} \times \{MULBH, MULBL\}}_{<1>} + \underbrace{\{MULR3, MULR2, MULR1, MULR0\}}_{<2>}$



Caution The values of the MULR1 and MULR0 registers in the fixed point mode (MULFRAC = 1) is the lower 32 bit of the Q62 format that is not shifted to the left. Unless carry/borrow occurs from bit 62 of the multiplication result register with the fixed point mode disabled, interrupt of overflow/underflow does not occur even in the fixed point mode (MULFRAC = 1).

41.4.8 Interrupt

In case of overflow/underflow of the multiply-accumulation result, interrupt signal is generated.

41.5 Operation of 32-bit Multiply-accumulator

Execute the MACL operation according to the following setting procedure.

<1> An example of executing during multiplication

- ● Enables input clock supply

  The MACEN bit of the PER2 register is set (1), and supplying the clock starts.

- ● Initial setting

  <STEP-1> Write to MULRSEL (to select a set of multiplication result registers)
  <STEP-2> Write to MULC (to enable or disable fixed point mode)
  <STEP-3> Write to the multiplication data register A (L) (MUL32UL or MUL32SL)
  <STEP-4> Write to the multiplication data register A (H) (MUL32UH or MUL32SH)
  <STEP-5> Write to the multiplication data register B (L) (MULBL)
  <STEP-6> Write to the multiplication data register B (H) (MULBH)

  - ● MULST = 1

- ● Multiplication operation

  When a series of processing shown below finishes, the operation finishes.

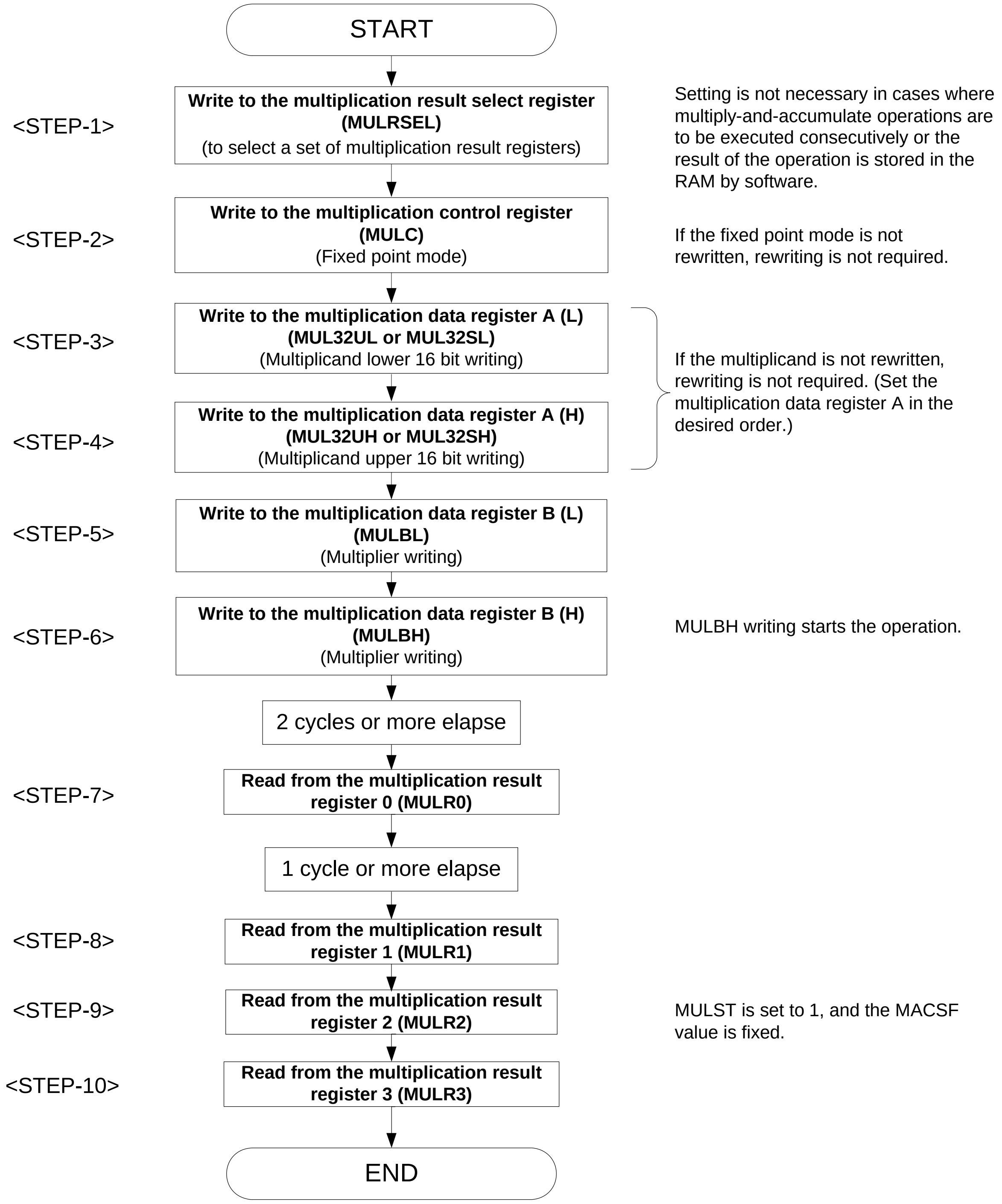
  - ● When all operations (5 cycles) of the multiplication operation process are finished, MULST is cleared to 0. (The sign flag MACSF is fixed to 0.)
  - ● The 2nd cycle after start of operation and subsequent cycles
    <STEP-7> Multiplication result register 0 (MULR0) can be read out.
  - ● The 4th cycle after start of operation and subsequent cycles
    <STEP-8> Multiplication result register 1 (MULR1) can be read out.
  - ● The 5th cycle after start of operation and subsequent cycles
    <STEP-9> Multiplication result register 2 (MULR2) can be read out.
    <STEP-10> Multiplication result register 3 (MULR3) can be read out.

- ● When executing the operation continuously

  (1) When switching the fixed point mode, execute <STEP-2>. For the other cases, proceed to (2).
  (2) When rewriting the multiplicand, execute <STEP-3> and <STEP-4>. For the other cases, proceed to (3).
  (3) Execute from <STEP-5>.

The operation flow when executing the multiplication is shown below:

Figure 41-9. MACL Operation Flow (for Multiplication)


START
<STEP-1>
Write to the multiplication result select register (MULRSEL)
(to select a set of multiplication result registers)
Setting is not necessary in cases where multiply-and-accumulate operations are to be executed consecutively or the result of the operation is stored in the RAM by software.
<STEP-2>
Write to the multiplication control register (MULC)
(Fixed point mode)
If the fixed point mode is not rewritten, rewriting is not required.
<STEP-3>
Write to the multiplication data register A (L) (MUL32UL or MUL32SL)
(Multiplicand lower 16 bit writing)
<STEP-4>
Write to the multiplication data register A (H) (MUL32UH or MUL32SH)
(Multiplicand upper 16 bit writing)
If the multiplicand is not rewritten, rewriting is not required. (Set the multiplication data register A in the desired order.)
<STEP-5>
Write to the multiplication data register B (L) (MULBL)
(Multiplier writing)
<STEP-6>
Write to the multiplication data register B (H) (MULBH)
(Multiplier writing)
MULBH writing starts the operation.
2 cycles or more elapse
<STEP-7>
Read from the multiplication result register 0 (MULR0)
1 cycle or more elapse
<STEP-8>
Read from the multiplication result register 1 (MULR1)
<STEP-9>
Read from the multiplication result register 2 (MULR2)
MULST is set to 1, and the MACSF value is fixed.
<STEP-10>
Read from the multiplication result register 3 (MULR3)
END

<2> An example when executing the multiply-accumulation

- Enables input clock supply

  The MACEN bit of the PER2 register is set (1), and supplying the clock starts.

- Initial setting

  <STEP-1> Write to MULRSEL (to select a set of multiplication result registers)

  <STEP-2> Write to MULC (to enable or disable fixed point mode)

  <STEP-3> Write to the multiplication result register 0 (MULR0) (accumulation initial setting)

  <STEP-4> Write to the multiplication result register 1 (MULR1) (accumulation initial setting)

  <STEP-5> Write to the multiplication result register 2 (MULR2) (accumulation initial setting)

  <STEP-6> Write to the multiplication result register 3 (MULR3) (accumulation initial setting)

  <STEP-7> Write to the multiplication data register A (L) (MAC32UL or MAC32SL)

  <STEP-8> Write to the multiplication data register A (H) (MAC32UH or MAC32SH)

  <STEP-9> Write to the multiplication data register B (L) (MULBL)

  <STEP-10> Write to the multiplication data register B (H) (MULBH)

  - MULST = 1

- During multiply-accumulation

  When a series of processing shown below finishes, the operation finishes.

  - In case of overflow/underflow, MACOF is set to 1, and interrupt (INTMACLOF = 1) occurs.
  - When all cycles (5 cycles) of the multiply-accumulation processing operation have been completed, MULST is cleared to 0, and the sign flag MACSF is set or cleared.

    (However, the MACSF flag is not set but fixed to 0 when MULSM is cleared to 0.)
  - The 2nd cycle after start of operation and subsequent cycles

    <STEP-11> Multiplication result register 0 (MULR0) can be read out.
  - The 4th cycle or after

    <STEP-12> Multiplication result register 1 (MULR1) can be read out.
  - The 5th cycle after start of operation and subsequent cycles

    <STEP-13> Multiplication result register 2 (MULR2) can be read out.

    <STEP-14> Multiplication result register 3 (MULR3) can be read out.

- When executing the operation continuously

  (1) When switching the fixed point mode, execute <STEP-2>. For the other cases, proceed to (2).

  (2) When rewriting accumulation initial value (multiplication result registers MULR0 to MULR3)

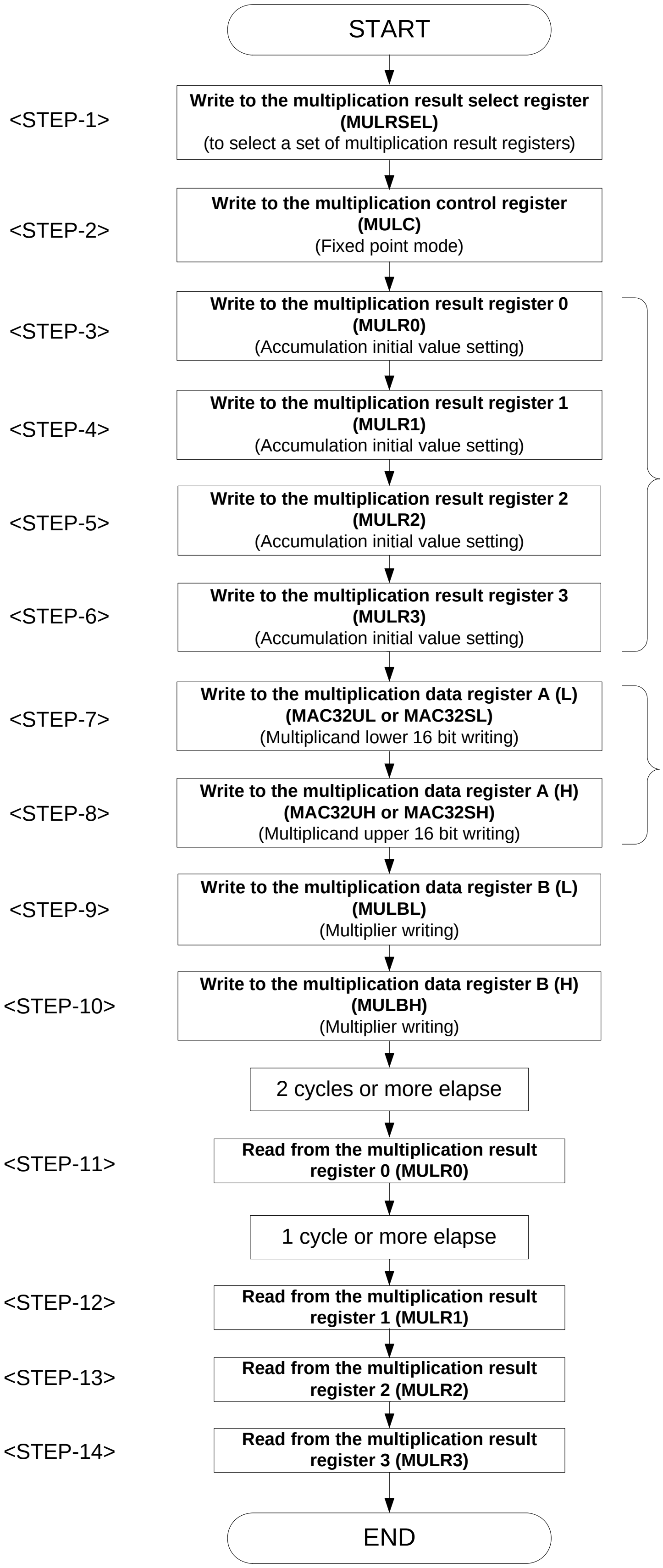
  Execute <STEP-3>, <STEP-4>, <STEP-5>, and <STEP-6>. For the other cases, proceed to (3).

  (3) When rewriting the multiplicand, execute <STEP-7> and <STEP-8>. For the other cases, proceed to (4).

  (4) Execute from <STEP-9>.

The operation flow when executing the multiply-accumulation is shown below:

Figure 41-10. MACL Operation Flow (for Multiply-accumulation)


START
<STEP-1>
Write to the multiplication result select register (MULRSEL)
(to select a set of multiplication result registers)
<STEP-2>
Write to the multiplication control register (MULC)
(Fixed point mode)
<STEP-3>
Write to the multiplication result register 0 (MULR0)
(Accumulation initial value setting)
<STEP-4>
Write to the multiplication result register 1 (MULR1)
(Accumulation initial value setting)
<STEP-5>
Write to the multiplication result register 2 (MULR2)
(Accumulation initial value setting)
<STEP-6>
Write to the multiplication result register 3 (MULR3)
(Accumulation initial value setting)
<STEP-7>
Write to the multiplication data register A (L) (MAC32UL or MAC32SL)
(Multiplicand lower 16 bit writing)
<STEP-8>
Write to the multiplication data register A (H) (MAC32UH or MAC32SH)
(Multiplicand upper 16 bit writing)
<STEP-9>
Write to the multiplication data register B (L) (MULBL)
(Multiplier writing)
<STEP-10>
Write to the multiplication data register B (H) (MULBH)
(Multiplier writing)
2 cycles or more elapse
<STEP-11>
Read from the multiplication result register 0 (MULR0)
1 cycle or more elapse
<STEP-12>
Read from the multiplication result register 1 (MULR1)
<STEP-13>
Read from the multiplication result register 2 (MULR2)
<STEP-14>
Read from the multiplication result register 3 (MULR3)
END

Setting is not necessary in cases where multiply-and-accumulate operations are to be executed consecutively or the result of the operation is stored in the RAM by software.

If the fixed point mode is not rewritten, rewriting is not required.

If the accumulation initial value is not rewritten, rewriting is not required. (At the accumulation initial value setting, set MULR0 to MULR3 in the desired order.)

If the multiplicand is not rewritten, rewriting is not required. (Set the multiplication data register A in the desired order.)

MULBH writing starts the operation.

MULST is set to 1, and the MACSF value is fixed.
In case of overflow/underflow of the multiply-accumulation, MACOF is set to 1, and interrupt (INTMACLOF = 1) occurs.

41.6 Precautions for 32-bit Multiply-accumulator

41.6.1 Precautions during operation (MULST = 1)

Rewriting of the multiplication data register A (L)/(H), multiplication data register B (L)/(H), multiplication result register 0/1/2/3, multiplication control register, and multiplication result select register is prohibited during the operation. Otherwise, the operation result will be an undefined value.

Before rewriting to the multiplication data register B (H) as the operation start, rewriting the multiplication data register A (L)/(H), multiplication data register B (L), multiplication result register 0/1/2/3, multiplication control register, and multiplication result select register must be completed.

CHAPTER 42 INSTRUCTION SET

This chapter 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).**

42.1 Conventions Used in Operation List

42.1.1 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 42-1. 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 FF1FH 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 1-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 SFR List** 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 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

42.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 42-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
⊻	Exclusive logical sum (exclusive OR)
–	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

42.1.3 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 42-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

42.1.4 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 42-4. 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.

42.2 Operation List

Table 42-5. 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 ← byte			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		CS, #byte	3	1	–	CS ← byte			
		ES, #byte	2	1	–	ES ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		[DE+byte], #byte	3	1	–	(DE+byte) ← byte			
		ES:[DE+byte],#byte	4	2	–	((ES, DE)+byte) ← byte			
		[HL+byte], #byte	3	1	–	(HL+byte) ← byte			
		ES:[HL+byte],#byte	4	2	–	((ES, HL)+byte) ← byte			
		[SP+byte], #byte	3	1	–	(SP+byte) ← byte			
		word[B], #byte	4	1	–	(B+word) ← byte			
		ES:word[B], #byte	5	2	–	((ES, B)+word) ← byte			
		word[C], #byte	4	1	–	(C+word) ← byte			
		ES:word[C], #byte	5	2	–	((ES, C)+word) ← byte			
		word[BC], #byte	4	1	–	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	–	((ES, BC)+word) ← byte			
		A, r [Note 3]	1	1	–	A ← r			
		r, A [Note 3]	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	–	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	–	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	–	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			
		[DE+byte], A	2	1	–	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	–	((ES, DE) + byte) ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	–	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	–	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	–	A ← (SP + byte)			
		[SP+byte], A	2	1	–	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	–	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	–	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	–	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	–	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	–	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← (HL + B)	
		[HL+B], A	2	1	–	(HL + B) ← A	
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL+B], A	3	2	–	((ES, HL) + B) ← A	
		A, [HL+C]	2	1	4	A ← (HL + C)	
		[HL+C], A	2	1	–	(HL + C) ← A	
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL+C], A	3	2	–	((ES, HL) + C) ← A	
		X, !addr16	3	1	4	X ← (addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		B, saddr	2	1	–	B ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		ES, saddr	3	1	–	ES ← (saddr)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, ES:!addr16	5	3	–	A ←→ (ES, addr16)	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)	
		A, [DE+byte]	3	2	–	A ←→ (DE + byte)	
		A, ES:[DE+byte]	4	3	–	A ←→ ((ES, DE) + byte)	
		A, [HL+byte]	3	2	–	A ←→ (HL + byte)	
		A, ES:[HL+byte]	4	3	–	A ←→ ((ES, HL) + byte)	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ←→ (HL+B)			
		A, ES:[HL+B]	3	3	–	A ←→ ((ES, HL)+B)			
		A, [HL+C]	2	2	–	A ←→ (HL+C)			
		A, ES:[HL+C]	3	3	–	A ←→ ((ES, HL)+C)			
	ONEB	A	1	1	–	A ← 01H			
		X	1	1	–	X ← 01H			
		B	1	1	–	B ← 01H			
		C	1	1	–	C ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
		saddr	2	1	–	(saddr) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	–	(HL+byte) ← X	×		×
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word			
		saddrp, #word	4	1	–	(saddrp) ← word			
		sfrp, #word	4	1	–	sfrp ← word			
		AX, rp Note 3	1	1	–	AX ← rp			
		rp, AX Note 3	1	1	–	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, saddrp	2	1	–	AX ← (saddrp)			
		saddrp, AX	2	1	–	(saddrp) ← AX			
		AX, sfrp	2	1	–	AX ← sfrp			
		sfrp, AX	2	1	–	sfrp ← AX			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE+byte)			
		[DE+byte], AX	2	1	–	(DE+byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	–	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	–	AX ← (SP + byte)			
		[SP+byte], AX	2	1	–	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B+ word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (fCLK) when the code flash area is accessed.

3. Except rp = AX

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 42-5. 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 ← A+byte+CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, rv Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL) – CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte) – CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B) – CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← A∨byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr)∨byte	×		
		A, r [Note 3]	2	1	–	A ← A∨r	×		
		r, A	2	1	–	r ← r∨A	×		
		A, !addr16	3	1	4	A ← A∨(addr16)	×		
		A, ES:!addr16	4	2	5	A ← A∨(ES:addr16)	×		
		A, saddr	2	1	–	A ← A∨(saddr)	×		
		A, [HL]	1	1	4	A ← A∨(H)	×		
		A, ES:[HL]	2	2	5	A ← A∨(ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A∨(HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A∨((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A∨(HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A∨((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A∨(HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A∨((ES:HL)+C)	×		
	XOR	A, #byte	2	1	–	A ← A⊻byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr)⊻byte	×		
		A, r [Note 3]	2	1	–	A ← A⊻r	×		
		r, A	2	1	–	r ← r⊻A	×		
		A, !addr16	3	1	4	A ← A⊻(addr16)	×		
		A, ES:!addr16	4	2	5	A ← A⊻(ES:addr16)	×		
		A, saddr	2	1	–	A ← A⊻(saddr)	×		
		A, [HL]	1	1	4	A ← A⊻(HL)	×		
		A, ES:[HL]	2	2	5	A ← A⊻(ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A⊻(HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A⊻((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A⊻(HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A⊻((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A⊻(HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A⊻((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	×	×	×
	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← AX+word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL)+byte)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← A × X			
	MULHU		3	2	–	BCAX ← AX × BC (unsigned)			
	MULH		3	2	–	BCAX ← AX × BC (signed)			
	DIVHU		3	9	–	AX (quotient), DE (remainder) ← AX ÷ DE (unsigned)			
	DIVWU		3	17	–	BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned)			
	MACHU		3	3	–	MACR ← MACR + AX × BC (unsigned)		×	×
	MACH		3	3	–	MACR ← MACR + AX × BC(signed)		×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Remarks 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.

2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 42-5. 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 ← r+1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16)+1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	×	×	
		saddr	2	2	–	(saddr) ← (saddr)+1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp+1			
		!addr16	3	2	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	2	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	1	–	rp ← rp – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) ×cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) ×cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) ×cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) ×cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) ×cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) ×cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) ×cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) ×cnt			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

Remarks 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.

2. cnt indicates the bit shift count.

Table 42-5. 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, A_7 ← A_0, A_{m-1} ← A_m)×1			×
	ROL	A, 1	2	1	–	(CY, A_0 ← A_7, A_{m+1} ← A_m)×1			×
	RORC	A, 1	2	1	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m)×1			×
	ROLC	A, 1	2	1	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m)×1			×
	ROLWC	AX,1	2	1	–	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) ×1			×
		BC,1	2	1	–	(CY ← BC_{15}, BC_0 ← CY, BC_{m+1} ← BC_m) ×1			×
Bit manipulate	MOV1	CY, A.bit	2	1	–	CY ← A.bit			×
		A.bit, CY	2	1	–	A.bit ← CY			
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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 ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	A.bit	2	1	–	A.bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	–	A.bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		saddr.bit	3	2	–	(saddr).bit ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{\text{CY}}$			×

Notes 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

2. Number of CPU clocks (fCLK) when the code flash area is accessed.

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 42-5. Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC+2)$_S$, (SP – 3) ← (PC+2)$_H$, (SP – 4) ← (PC+2)$_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← (PC+3)$_S$, (SP – 3) ← (PC+3)$_H$, (SP – 4) ← (PC+3)$_L$, PC ← PC+3+jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← (PC+3)$_S$, (SP – 3) ← (PC+3)$_H$, (SP – 4) ← (PC+3)$_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← (PC+4)$_S$, (SP – 3) ← (PC+4)$_H$, (SP – 4) ← (PC+4)$_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC+2)$_S$, (SP – 3) ← (PC+2)$_H$, (SP – 4) ← (PC+2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0000, addr5+1), PC$_L$ ← (0000, addr5), SP ← SP – 4			
	BRK	–	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC+2)$_S$, (SP – 3) ← (PC+2)$_H$, (SP – 4) ← (PC+2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0007FH), PC$_L$ ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	–	1	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP+1), PC$_S$ ← (SP+2), SP ← SP+4			
	RETI	–	2	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP+1), PC$_S$ ← (SP+2), PSW← (SP+3), SP ← SP+4	R	R	R
	RETB	–	2	6	–	PC$_L$ ← (SP), PC$_H$ ← (SP+1), PC$_S$ ← (SP+2), PSW ← (SP+3), SP ← SP+4	R	R	R

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

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 42-5. 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) ← PSW, (SP – 2) ← 00H, SP ← SP–2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Un-conditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 Note 3	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 Note 3	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 Note 3	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 Note 3	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 Note 3	–	PC ← PC + 3 + jdisp8 if (Z∨CY) = 0			
	BNH	$addr20	3	2/4 Note 3	–	PC ← PC + 3 + jdisp8 if (Z∨CY) = 1			
	BT	saddr.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 Note 3	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 42-5. 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	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 3	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 3	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 3	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 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 Note 4	RBn	2	1	–	RBS[1:0] ← n			
	NOP	–	1	1	–	No Operation			
	EI	–	3	4	–	IE ← 1 (Enable Interrupt)			
	DI	–	3	4	–	IE ← 0 (Disable Interrupt)			
	HALT	–	2	3	–	Set HALT Mode			
	STOP	–	2	3	–	Set STOP Mode			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

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.

CHAPTER 43 ELECTRICAL SPECIFICATIONS

Cautions 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. See 2.1 Port Function to 2.2.1 Functions mounted on the products with different numbers of pins.

Remark In the descriptions in this chapter, read EV_{DD} as EV_{DD0} and EV_{DD1}, and EV_{SS} as EV_{SS0} and EV_{SS1}.

43.1 Absolute Maximum Ratings

Absolute Maximum Ratings (1/3)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
	EV_{DD}		–0.5 to +6.5	V
	V_{RTC}		–0.5 to +6.5	V
	AV_{DD}	$AV_{DD} = V_{DD}$	–0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3[Note 1]	V
Input voltage	V_{I1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	–0.3 to EV_{DD} +0.3 and –0.3 to V_{DD} +0.3[Note 2]	V
	V_{I2}	P60 to P62, P150 to P152 (N-ch open-drain)	–0.3 to +6.5	V
	V_{I3}	P121, P122, P137, EXCLK	–0.3 to V_{DD} +0.3[Note 2]	V
	V_{I4}	$\overline{\text{RESET}}$	–0.3 to +6.5	V
	V_{I5}	P123, P124, EXCLKS	–0.3 to V_{RTC} +0.3[Note 2]	V
	V_{I6}	P20 to P25	–0.3 to AV_{DD} +0.3[Note 2]	V
Output voltage	V_{O1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P60 to P62, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130, P150 to P152	–0.3 to EV_{DD} +0.3 and –0.3 to V_{DD} +0.3[Note 2]	V
	V_{O2}	P20 to P25	–0.3 to AV_{DD} +0.3 [Note 2]	V
Analog input voltage	V_{AI1}	ANI0 to ANI5	–0.3 to AV_{DD} +0.3 and –0.3 to $AV_{REF(+)}$ +0.3[Notes 2, 3]	V
	V_{AI2}	ANIP0 to ANIP3, ANIN0 to ANIN3	–0.6 to +2.8 and –0.6 to AREGC +0.3[Note 4]	V
Reference supply voltage	V_{IDSAD}	AREGC, AVCM, AVRT	–0.3 to +2.8 and –0.3 to AV_{DD} +0.3[Note 5]	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

4. The ΔΣ A/D conversion target pin must not exceed AREGC +0.3 V.

5. Connect AREGC, AVCM, and AVRT terminals to V_{SS} via capacitor (0.47 μF).
This value defines the absolute maximum rating of AREGC, AVCM, and AVRT terminal. Do not use with voltage applied.

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.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF (+)}$: Positive reference voltage of the 12-bit A/D converter

3. V_{SS}: Reference voltage

Absolute Maximum Ratings (2/3)

Parameter	Symbols	Conditions		Ratings	Unit
LCD voltage	V_{LI1}	V_{L1} voltage[Note 1]		–0.3 to 2.8 and –0.3 to V_{L4} +0.3	V
	V_{LI2}	V_{L2} voltage[Note 1]		–0.3 to V_{L4} +0.3[Note 2]	V
	V_{LI3}	V_{L3} voltage[Note 1]		–0.3 to V_{L4} +0.3[Note 2]	V
	V_{LI4}	V_{L4} voltage[Note 1]		–0.3 to +6.5	V
	V_{LCAP}	CAPL, CAPH voltage[Note 1]		–0.3 to V_{L4} +0.3[Note 2]	V
	V_{OUT}	COM0 to COM7, SEG0 to SEG41, output voltage	External resistance division method	–0.3 to V_{DD} +0.3[Note 2]	V
			Capacitor split method	–0.3 to V_{DD} +0.3[Note 2]	V
			Internal voltage boosting method	–0.3 to V_{L4} +0.3[Note 2]	V

Notes 1. This value only indicates the absolute maximum ratings when applying voltage to the V_{L1}, V_{L2}, V_{L3}, and V_{L4} pins; it does not mean that applying voltage to these pins is recommended. When using the internal voltage boosting method or capacitance split method, connect these pins to V_{SS} via a capacitor (0.47 μF ± 30%) and connect a capacitor (0.47 μF ± 30%) between the CAPL and CAPH pins.

2. Must be 6.5 V or lower.

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 V_{SS}: Reference voltage

Absolute Maximum Ratings (3/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130	–40	mA
		Total of all pins –170 mA	P02 to P07, P40, P42, P43, P130	–70	mA
			P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	–100	mA
	I_{OH2}	Per pin	P20 to P25	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130	40	mA
		Total of all pins 170 mA	P02 to P07, P40, P42, P43, P130	70	mA
			P10 to P17, P30 to P37, P50 to P57, P60 to P62, P70 to P77, P80 to P85, P90 to P97, P125 to P127	100	mA
	I_{OL2}	Per pin	P20 to P25	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			–65 to +150	°C

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 Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

43.2 Oscillator Characteristics

43.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +85°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	MHz
		1.8 V ≤ V_{DD} < 2.4 V	1.0		8.0	MHz
		1.6 V ≤ V_{DD} < 1.8 V	1.0		4.0	MHz

(T_A = –40 to +85°C, 1.6 V ≤ V_{RTC} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
XT1 clock oscillation frequency (f_{XT})Note	Crystal resonator	1.6 V ≤ V_{RTC} ≤ 5.5 V	32	32.768	35	kHz
			31	38.4	39	kHz

Note Indicates only permissible oscillator frequency ranges. See **43.4 AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, see **6.4 System Clock Oscillator**.

43.2.2 On-chip oscillator characteristics

(T_A = –40 to +85°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency[Notes 1, 2]	f_{IH}			1.0		32	MHz
High-speed on-chip oscillator clock frequency accuracy		–20 to +85°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1.0		+1.0	%
			1.6 V ≤ V_{DD} < 1.8 V	–5.0		+5.0	%
		–40 to –20°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1.5		+1.5	%
			1.6 V ≤ V_{DD} < 1.8 V	–5.5		+5.5	%
Middle-speed on-chip oscillator clock frequency[Note 2]	f_{IM}			1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy		1.8 V ≤ V_{DD} ≤ 5.5 V		–12		+12	%
Low-speed on-chip oscillator clock frequency	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				–15		+15	%

Notes 1. The high-speed on-chip oscillator frequency is selected by using bits 0 to 3 of option byte (000C2H/400C2H) and bits 0 to 2 of the HOCODIV register.

2. This indicates the oscillator characteristics only. See **43.4 AC Characteristics** for the instruction execution time.

43.2.3 PLL oscillator characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
PLL input frequency [Note]	f_{PLLIN}	f_{IH}		4		MHz
PLL output frequency [Note]	f_{PLL}			32		MHz
Lockup wait time		Wait time from PLL output enable to frequency stabilization	40			µs
Interval wait time		Wait time from PLL stop to PLL restart setting	4			µs
Setting wait time		Wait time from PLL input clock stabilization and PLL setting fixedness to start-up setting	1			µs

Note Indicates only permissible oscillator frequency ranges.

43.3 DC Characteristics

43.3.1 Pin characteristics

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130	1.6 V ≤ EV_{DD} ≤ 5.5 V			–10.0[Note 2]	mA
		Total of P02 to P07, P40, P42, P43, P130 (When duty ≤ 70%[Note 3])	4.0 V ≤ EV_{DD} ≤ 5.5 V			–55.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–10.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–5.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			–2.5	mA
		Total of P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127 (When duty ≤ 70%[Note 3])	4.0 V ≤ EV_{DD} ≤ 5.5 V			–80.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			–19.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			–10.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			–5.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])				–100.0	mA
	I_{OH2}	Per pin for P20 to P25	1.6 V ≤ AV_{DD} ≤ 5.5 V			–0.1[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	1.6 V ≤ AV_{DD} ≤ 5.5 V			–0.6	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the EV_{DD}, V_{DD}, and AV_{DD} pins to an output pin.

2. Do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OH} = –10.0 mA

  Total output current of pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.

A current higher than the absolute maximum rating must not flow into one pin.

Caution P02 to P07, P15 to P17, P51, P54, P56, P57, P80 to P82, P84, and P85 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130				20.0[Note 2]	mA
		Per pin for P60 to P62, P150 to P152				15.0[Note 2]	mA
		Total of P02 to P07, P40, P42, P43, P130 (When duty ≤ 70%[Note 3])	4.0 V ≤ EV_{DD} ≤ 5.5 V			70.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			9.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			4.5	mA
		Total of P10 to P17, P30 to P37, P50 to P57, P60 to P62, P70 to P77, P80 to P85, P90 to P97, P125 to P127 (When duty ≤ 70%[Note 3])	4.0 V ≤ EV_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			20.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])				150.0	mA
	I_{OL2}	Per pin for P20 to P25	1.6 V ≤ AV_{DD} ≤ 5.5 V			0.4[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	1.6 V ≤ AV_{DD} ≤ 5.5 V			2.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the EV_{SS}, V_{SS}, and AV_{SS} pins.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OL} = 10.0 mA

  Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	Normal input buffer	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH2}	P02, P03, P05, P06, P15, P16, P42, P43, P52, P53, P55, P57, P80, P81, P84	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	2.2		EV_{DD}	V
			TTL input buffer 3.3 V ≤ EV_{DD} < 4.0 V	2.0		EV_{DD}	V
			TTL input buffer 1.6 V ≤ EV_{DD} < 3.3 V	1.5		EV_{DD}	V
	V_{IH3}	P20 to P25		$0.7AV_{DD}$		AV_{DD}	V
	V_{IH4}	P60 to P62		$0.7EV_{DD}$		6.0	V
	V_{IH5}	P121, P122, P137, P150 to P152, EXCLK		$0.8V_{DD}$		V_{DD}	V
	V_{IH6}	$\overline{\text{RESET}}$		$0.8V_{DD}$		6.0	V
	V_{IH7}	P123, P124, EXCLKS		$0.8V_{RTC}$		V_{RTC}	V
<R>	V_{IH8}	P150 to P152 Note		$0.8V_{DD}$		6.0	V
	V_{IH9}	RTCIC0 to RTCIC2Note		$0.8V_{RTC}$		6.0	V
Input voltage, low	V_{IL1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	Normal input buffer	0		$0.2EV_{DD}$	V
	V_{IL2}	P02, P03, P05, P06, P15, P16, P42, P43, P52, P53, P55, P57, P80, P81, P84	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ EV_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 1.6 V ≤ EV_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P25		0		$0.3AV_{DD}$	V
	V_{IL4}	P60 to P62		0		$0.3EV_{DD}$	V
	V_{IL5}	P121, P122, P137, P150 to P152, EXCLK, $\overline{\text{RESET}}$		0		$0.2V_{DD}$	V
	V_{IL6}	P123, P124, EXCLKS		0		$0.2V_{RTC}$	V
<R>	V_{IL7}	P150 to P152 Note		0		$0.2V_{DD}$	V
	V_{IL8}	RTCIC0 to RTCIC2 Note		0		$0.2V_{RTC}$	V

Caution The maximum value of V_{IH} of pins P02 to P07, P15 to P17, P51, P54, P56, P57, P80 to P82, P84, and P85 is EV_{DD}, even in the N-ch open-drain mode.

<R> **Note** When a high-level signal is to be input to any pin among pins P150 to P152 (including if the pin is in use for a multiplexed pin function rather than for GPIO), the pin should always be individually connected via a resistor to V_{DD} or VRTC, whichever of the voltages is higher at the time, or to a voltage higher than both but no higher than 6 V.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –10.0 mA	EV_{DD} – 1.5			V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	EV_{DD} – 0.7			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	EV_{DD} – 0.6			V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	EV_{DD} – 0.5			V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	EV_{DD} – 0.5			V
	V_{OH2}	P20 to P25	1.6 V ≤ AV_{DD} ≤ 5.5 V, I_{OH2} = –100 µA	AV_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127, P130	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 20 mA			1.3	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 0.3 mA			0.4	V
	V_{OL2}	P150 to P152	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤5.5 V, I_{OL2} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 2.0 mA			0.4	V
			1.6 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 1.0 mA			0.4	V
	V_{OL3}	P60 to P62	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 15.0 mA			2.0	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 5.0 mA			0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 3.0 mA			0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 2.0 mA			0.4	V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 1.0 mA			0.4	V
	V_{OL4}	P20 to P25	1.6 V ≤ AV_{DD} ≤ 5.5 V, I_{OL4} = –100 µA			0.4	V

(**Caution** and **Remark** are listed on the next page.)

Caution **P02 to P07, P15 to P17, P51, P54, P56, P57, P80 to P82, P84, and P85 do not output high level in N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P60 to P62, P70 to P77, P80 to P85, P90 to P97, P125 to P127	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P137, P150 to P152, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
	I_{LIH4}	P123, P124 (XT1, XT2, EXCLKS)	$V_I = V_{RTC}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
	I_{LIH5}	P20 to P25	$V_I = AV_{DD}$				1	μA
Input leakage current, low	I_{LIL1}	P02 to P07, P10 to P17, P30 to P37, P40, P42, P43, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	$V_I = EV_{SS}$				–1	μA
	I_{LIL2}	P137, P150 to P152, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
	I_{LIL4}	P123, P124 (XT1, XT2, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
	I_{LIL5}	P20 to P25	$V_I = AV_{SS1}$				–1	μA
On-chip pull-up resistance	R_{U1}	P10 to P17, P30 to P37, P50 to P57, P70 to P77, P80 to P85, P90 to P97, P125 to P127	$V_I = EV_{SS}$	2.4 V ≤ EV_{DD} ≤ 5.5 V	10	20	100	kΩ
				1.6 V ≤ EV_{DD} < 2.4 V	10	30	100	kΩ
	R_{U2}	P02 to P07, P40, P42, P43	$V_I = EV_{SS}$		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

43.3.2 Supply current characteristics

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (1/6)

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS (high-speed main) mode[Note 5]	f_{IH} = 32 MHz[Note 3]	Basic operation	V_{DD} = 5.0 V		2.5		mA
						V_{DD} = 3.0 V		2.5		mA
					Normal operation	V_{DD} = 5.0 V		5.6	9.1	mA
						V_{DD} = 3.0 V		5.6	9.1	mA
				f_{CLK} = 32 MHz[Note 3] PLL operation	Normal operation	V_{DD} = 5.0 V		5.8	9.0	mA
						V_{DD} = 3.0 V		5.8	9.0	mA
				f_{IH} = 24 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V		4.5	7.3	mA
						V_{DD} = 3.0 V		4.5	7.3	mA
				f_{IH} = 16 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V		3.2	5.8	mA
						V_{DD} = 3.0 V		3.2	5.8	mA
				f_{IH} = 12 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V		2.6	4.9	mA
						V_{DD} = 3.0 V		2.6	4.9	mA
				f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V		1.9	3.0	mA
						V_{DD} = 3.0 V		1.9	3.0	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 5.0 V		1.5	2.4	mA
						V_{DD} = 3.0 V		1.5	2.4	mA
			LS (low-speed main) mode[Note 5] (MCSEL = 0)	f_{IH} = 8 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		1.5	2.5	mA
						V_{DD} = 2.0 V		1.5	2.5	mA
				f_{IH} = 6 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		1.2	2.3	mA
						V_{DD} = 2.0 V		1.2	2.3	mA
				f_{IH} = 3 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		0.8	1.6	mA
						V_{DD} = 2.0 V		0.8	1.6	mA
			LS (low-speed main) mode[Note 5] (MCSEL = 1)	f_{IH} = 4 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		1.0	1.7	mA
						V_{DD} = 2.0 V		1.0	1.7	mA
				f_{IM} = 4 MHz[Note 6]	Normal operation	V_{DD} = 3.0 V		0.8	1.5	mA
						V_{DD} = 2.0 V		0.8	1.5	mA
			LV (low-voltage main) mode[Note 5]	f_{IH} = 4 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		1.7	2.8	mA
						V_{DD} = 2.0 V		1.7	2.8	mA
			LP (low-power main) mode[Note 5] (MCSEL = 1)	f_{IH} = 1 MHz[Note 3]	Normal operation	V_{DD} = 3.0 V		330	550	µA
						V_{DD} = 2.0 V		330	550	µA
				f_{IM} = 1 MHz[Note 6]	Normal operation	V_{DD} = 3.0 V		170	360	µA
						V_{DD} = 2.0 V		170	360	µA

(**Notes** and **Remarks** are listed on the page after the next page.)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (2/6)

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	HS (high-speed main) mode[Note 5]	f_{MX} = 20 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input		3.4	6.4	mA
						Resonator connection		3.7	6.5	mA
				f_{MX} = 20 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		3.4	6.4	mA
						Resonator connection		3.7	6.5	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input		2.4	4.6	mA
						Resonator connection		2.6	4.7	mA
				f_{MX} = 12 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		2.4	4.6	mA
						Resonator connection		2.6	4.7	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 5.0 V	Normal operation	Square wave input		2.1	4.1	mA
						Resonator connection		2.4	4.1	mA
				f_{MX} = 10 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		2.1	4.1	mA
						Resonator connection		2.4	4.1	mA
			LS (low-speed main) mode[Note 5] (MCSEL = 0)	f_{MX} = 8 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		1.2	2.4	mA
						Resonator connection		1.3	2.4	mA
				f_{MX} = 8 MHz[Note 2], V_{DD} = 2.0 V	Normal operation	Square wave input		1.2	2.4	mA
						Resonator connection		1.3	2.4	mA
			LS (low-speed main) mode[Note 5] (MCSEL = 1)	f_{MX} = 4 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		0.8	1.6	mA
						Resonator connection		0.8	1.5	mA
				f_{MX} = 4 MHz[Note 2], V_{DD} = 2.0 V	Normal operation	Square wave input		0.8	1.6	mA
						Resonator connection		0.8	1.5	mA
			LP (low-power main) mode[Note 5] (MCSEL = 1)	f_{IH} = 1 MHz[Note 2], V_{DD} = 3.0 V	Normal operation	Square wave input		150	320	µA
						Resonator connection		190	360	µA
				f_{IH} = 1 MHz[Note 2], V_{DD} = 2.0 V	Normal operation	Square wave input		150	320	µA
						Resonator connection		190	360	µA
			Sub clock operation	f_{SUB} = 32.768 kHz[Note 4], T_A = –40°C	Normal operation	Square wave input		5.7	11.0	µA
						Resonator connection		5.9	11.1	µA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +25°C	Normal operation	Square wave input		6.1	8.0	µA
						Resonator connection		6.4	8.1	µA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +50°C	Normal operation	Square wave input		6.3	9.3	µA
						Resonator connection		6.6	9.4	µA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +70°C	Normal operation	Square wave input		6.9	12.6	µA
						Resonator connection		7.1	12.7	µA
				f_{SUB} = 32.768 kHz[Note 4], T_A = +85°C	Normal operation	Square wave input		7.8	21.7	µA
						Resonator connection		7.9	21.8	µA
				f_{IL} = 15 kHz, T_A = –40°C [Note 7]	Normal operation			2.8	9	µA
				f_{IL} = 15 kHz, T_A = +25°C [Note 7]	Normal operation			3.1	9	µA
				f_{IL} = 15 kHz, T_A = +85°C [Note 7]	Normal operation			5.0	13	µA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, EV_{DD}, and V_{RTC} including the input leakage current flowing when the level of the input pin is fixed to V_{DD}, EV_{DD}, V_{RTC} or V_{SS}, EV_{SS}. The following points apply in the HS (high-speed main), LS (low-speed main), LV (low-voltage 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 peripheral modules, except for those flowing into the LCD controller/driver, 12-bit A/D converter, ΔΣA/D converter, LVD circuit, battery backup circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the independent power supply RTC.

2. When high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

3. When high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

4. When high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped. When setting ultra-low current consumption (AMPHS1 = 1).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode:	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz to 32 MHz
	$2.4 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz to 16 MHz
	$2.1 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz to 6 MHz
LS (low-speed main) mode:	$1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz to 8 MHz
LP (low-power main) mode:	$1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz
LV (low-voltage main) mode:	$1.6 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$@1 MHz to 4 MHz

6. When high-speed on-chip oscillator, low-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped.

7. When high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{IM}: Middle-speed on-chip oscillator clock frequency

4. f_{IL}: Low-speed on-chip oscillator clock frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

6. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (3/6)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD2}[Note 2]	HALT mode	HS (high-speed main) mode[Note 7]	f_{IH} = 32 MHz[Note 4]	V_{DD} = 5.0 V		0.69	1.9	mA
					V_{DD} = 3.0 V		0.68	1.9	mA
				f_{CLK} = 32 MHz[Note 4], PLL operation	V_{DD} = 5.0 V		1.2	2.2	mA
					V_{DD} = 3.0 V		1.2	2.2	mA
				f_{IH} = 24 MHz[Note 4]	V_{DD} = 5.0 V		0.56	1.5	mA
					V_{DD} = 3.0 V		0.56	1.5	mA
				f_{IH} = 16 MHz[Note 4]	V_{DD} = 5.0 V		0.49	1.2	mA
					V_{DD} = 3.0 V		0.49	1.2	mA
				f_{IH} = 12 MHz[Note 4]	V_{DD} = 5.0 V		0.41	1.0	mA
					V_{DD} = 3.0 V		0.41	1.0	mA
				f_{IH} = 6 MHz[Note 4]	V_{DD} = 5.0 V		0.36	0.8	mA
					V_{DD} = 3.0 V		0.36	0.8	mA
				f_{IH} = 3 MHz[Note 4]	V_{DD} = 5.0 V		0.33	0.7	mA
					V_{DD} = 3.0 V		0.33	0.7	mA
			LS (low-speed main) mode[Note 7] (MCSEL = 0)	f_{IH} = 8 MHz[Note 4]	V_{DD} = 3.0 V		290	755	µA
					V_{DD} = 2.0 V		290	755	µA
				f_{IH} = 6 MHz[Note 4]	V_{DD} = 3.0 V		240	655	µA
					V_{DD} = 2.0 V		240	655	µA
				f_{IH} = 3 MHz[Note 4]	V_{DD} = 3.0 V		210	556	µA
					V_{DD} = 2.0 V		210	556	µA
			LS (low-speed main) mode[Note 7] (MCSEL = 1)	f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V		220	450	µA
					V_{DD} = 2.0 V		220	450	µA
				f_{IM} = 4 MHz[Note 5]	V_{DD} = 3.0 V		60	350	µA
					V_{DD} = 2.0 V		60	350	µA
			LV (low-voltage main) mode[Note 7]	f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V		625	1200	µA
					V_{DD} = 2.0 V		625	1200	µA
			LP (low-power main) mode[Note 7] (MCSEL = 1)	f_{IH} = 1 MHz[Note 4]	V_{DD} = 3.0 V		200	410	µA
					V_{DD} = 2.0 V		200	410	µA
				f_{IM} = 1 MHz[Note 5]	V_{DD} = 3.0 V		35	150	µA
					V_{DD} = 2.0 V		35	150	µA
			HS (high-speed main) mode[Note 7]	f_{MX} = 20 MHz[Note 3], V_{DD} = 5.0 V	Square wave input		0.31	1.15	mA
					Resonator connection		0.53	1.35	mA
				f_{MX} = 20 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		0.31	1.15	mA
					Resonator connection		0.53	1.35	mA
				f_{MX} = 12 MHz[Note 3], V_{DD} = 5.0 V	Square wave input		0.23	0.85	mA
					Resonator connection		0.41	0.95	mA
				f_{MX} = 12 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		0.23	0.85	mA
					Resonator connection		0.41	0.95	mA
				f_{MX} = 10 MHz[Note 3], V_{DD} = 5.0 V	Square wave input		0.21	0.75	mA
					Resonator connection		0.36	0.86	mA
				f_{MX} = 10 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		0.21	0.75	mA
					Resonator connection		0.36	0.86	mA

(**Notes** and **Remarks** are listed on the page after the next page.)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (4/6)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD2}[Note 2]	HALT mode	LS (low-speed main) mode[Note 7] (MCSEL = 0)	f_{MX} = 8 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		113	420	µA
					Resonator connection		176	485	µA
				f_{MX} = 8 MHz[Note 3], V_{DD} = 2.0 V	Square wave input		113	420	µA
					Resonator connection		176	485	µA
			LS (low-speed main) mode[Note 7] (MCSEL = 1)	f_{MX} = 4 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		41	240	µA
					Resonator connection		94	290	µA
				f_{MX} = 4 MHz[Note 3], V_{DD} = 2.0 V	Square wave input		41	240	µA
					Resonator connection		94	290	µA
			LP (low-power main) mode[Note 7] (MCSEL = 1)	f_{MX} = 1 MHz[Note 3], V_{DD} = 3.0 V	Square wave input		14	110	µA
					Resonator connection		70	210	µA
				f_{MX} = 1 MHz[Note 3], V_{DD} = 2.0 V	Square wave input		14	110	µA
					Resonator connection		70	210	µA
			Sub clock operation	f_{SUB} = 32.768 kHz[Note 6], T_A = –40°C	Square wave input		0.80	6.6	µA
					Resonator connection		1.00	6.8	µA
				f_{SUB} = 32.768 kHz[Note 6], T_A = +25°C	Square wave input		1.0	4.1	µA
					Resonator connection		1.4	4.3	µA
				f_{SUB} = 32.768 kHz[Note 6], T_A = +50°C	Square wave input		1.2	5.6	µA
					Resonator connection		1.6	5.7	µA
				f_{SUB} = 32.768 kHz[Note 6], T_A = +70°C	Square wave input		1.6	9.0	µA
					Resonator connection		2.0	10.6	µA
				f_{SUB} = 32.768 kHz[Note 6], T_A = +85°C	Square wave input		2.80	16.2	µA
					Resonator connection		3.00	19.6	µA
				f_{IL} = 15 kHz[Note 9], T_A = –40°C			0.83	1.85	µA
									µA
				f_{IL} = 15 kHz[Note 9], T_A = +25°C			1.07	2.25	µA
									µA
				f_{IL} = 15 kHz[Note 9], T_A = +85°C			2.68	28.1	µA
									µA
	I_{DD3}	STOP mode[Note 8]	T_A = –40°C				0.47	0.95	µA
			T_A = +25°C				0.66	1.60	µA
			T_A = +50°C				0.84	4.80	µA
			T_A = +70°C				1.22	10.60	µA
			T_A = +85°C				1.94	13	µA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD}, EV_{DD}, and V_{RTC} including the input leakage current flowing when the level of the input pin is fixed to V_{DD}, EV_{DD}, V_{RTC} or V_{SS}, EV_{SS}. The following points apply in the HS (high-speed main), LS (low-speed main), LV (low-voltage 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 peripheral modules, except for those flowing into LCD controller/driver, the 12-bit A/D converter, ΔΣA/D converter, LVD circuit, battery backup circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the independent power supply RTC.

In the STOP mode, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

4. When high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock are stopped.

5. When high-speed on-chip oscillator system clock, low-speed on-chip oscillator, high-speed system clock, and Subsystem clock are stopped.

6. When operating independent power supply RTC and setting ultra-low current consumption (AMPHS1 = 1). When high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.

7. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode:	2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 32 MHz
	2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz
	2.1 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 6 MHz
LS (low-speed main) mode:	1.8 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 8 MHz
LP (low-power main) mode:	1.8 V ≤ V_{DD} ≤ 5.5 V@1 MHz
LV (low-voltage main) mode:	1.6 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 4 MHz

8. If operation of the subsystem clock when STOP mode, same as when HALT mode of subsystem clock operation.

9. When high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: High-speed on-chip oscillator clock frequency

3. f_{IM}: Middle-speed on-chip oscillator clock frequency

4. f_{IL}: Low-speed on-chip oscillator clock frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

6. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (5/6)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Independent power supply RTC operating current	I_{RTC}[Note 2]	f_{SUB} = 32.768 kHz			0.70		μA
12-bit interval timer operating current	I_{TMKA}[Notes 1, 3]	f_{SUB} = 38.4 kHz, f_{MAIN} is stopped			0.04		μA
		f_{SUB} = 32.768 kHz, f_{MAIN} is stopped			0.04		μA
8-bit interval timer operating current	I_{TMT}[Notes 1, 4]	f_{SUB} = 38.4 kHz, f_{MAIN} is stopped, per unit	8-bit counter mode × 2 ch operation		0.14		μA
			16-bit counter mode operation		0.12		μA
		f_{SUB} = 32.768 kHz, f_{MAIN} is stopped, per unit	8-bit counter mode × 2 ch operation		0.12		μA
			16-bit counter mode operation		0.10		μA
Watchdog timer operating current	I_{WDT}[Notes 1, 5]	f_{IL} = 15 kHz, f_{MAIN} is stopped			0.22		μA
LVD operating current	I_{LVD}[Note 6]				0.10		μA
LVDVDD operating current	I_{LVDVDD}	Current flowing to V_{DD}			0.05		μA
LVDVBAT operating current	$I_{LVDVBAT}$	Current flowing to LVDVBAT			0.04		μA
		Current flowing to V_{DD}			0.05		μA
LVDVRTC operating current	$I_{LVDVRTC}$	Current flowing to V_{RTC}			0.04		μA
		Current flowing to V_{DD}			0.05		μA
LVDEXLVD operating current	$I_{LVDEXLVD}$	Current flowing to EXLVD			0.16		μA
		Current flowing to V_{DD}			0.05		μA
Oscillation stop detection circuit operating current	I_{OSDC}				0.02		μA
12-bit A/D converter operating current	I_{ADC}[Note 7]	AV_{REFP} = 5.0 V, when conversion at maximum speed [Note 8]			1.2	1.8	mA
12-bit A/D converter AVREF(+) current	I_{ADREF}[Note 9]	AV_{REFP} = 5.0 V, HVSEL[1:0] = 01B[Note 10]			50	80	μA
Temperature sensor operating current	I_{TMPS}				125		μA
BGO operating current	I_{BGO}[Note 11]				2.00	12.20	mA
Bank programming operating current	I_{BNKP}				5.60	12.20	mA
Self-programming operating current	I_{FSP}[Note 12]				2.00	12.20	mA

(**Notes** and **Remarks** are listed on the page after the next page.)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (6/6)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
24-Bit ΔΣ A/D Converter operating current	I_{DSAD} Note 13	In 4 ch ΔΣ A/D converter operation				1.45	2.30	mA
		In 3 ch ΔΣ A/D converter operation				1.14	1.85	mA
		In 1 ch ΔΣA/D converter operation				0.52	0.94	mA
SNOOZE operating current	I_{SNOZ} Note 14	Simplified SPI (CSI)/UART operation				0.70	1.05	mA
		DTC operation				2.20		mA
LCD operating current	I_{LCD1} Notes 15, 16	External resistance division method	f_{LCD} = f_{SUB} (32.768 kHz) LCD clock = 128 Hz 1/3 bias, four-time-slices	V_{DD} = 5.0 V, V_{L4} = 5.0 V		0.06		μA
	I_{LCD2} Note 15	Internal voltage boosting method	f_{LCD} = f_{SUB} (32.768 kHz) LCD clock = 128 Hz 1/3 bias, four-time-slices	V_{DD} = 3.0 V, V_{L4} = 3.0 V (VLCD = 04H)		0.85		μA
				V_{DD} = 5.0 V, V_{L4} = 5.1 V (VLCD = 12H)		1.55		μA
	I_{LCD3} Note 15	Capacitor split method	f_{LCD} = f_{SUB} (32.768 kHz) LCD clock = 128 Hz 1/3 bias, four-time-slices	V_{DD} = 3.0 V, V_{L4} = 3.0 V		0.20		μA
Timer RJ operating current	I_{TMRJ} Note 17	f_{SX} = 32.768 or 38.4 kHz, f_{MAIN} is stopped, per unit				0.10		μA
Serial interface UARTMG operating current	I_{UARTMG} Note 18	f_{SX} = 38.4 kHz, f_{MAIN} is stopped, per unit				0.12		μA
Sampling output timer detector operating current	I_{SMOTD} Note 19	f_{SX} = 32.768 or 38.4 kHz, f_{MAIN} is stopped, per unit				0.10		μA
VREFADC operating current	$I_{VREFOUT}$ Note 20	VREFAMPCNT.BGREN = 1 VREFAMPCNT.VREFADCEN = 1 Note 8				73	130	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. When high speed on-chip oscillator and high-speed system clock are stopped.

2. Current flowing to V_{RTC} pin, including RTC power supply, subsystem clock oscillator circuit, and RTC.

3. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and XT1 oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{TMKA}, when the 12-bit interval timer operates in the operating mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

4. Current flowing only to the 8-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and XT1 oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{TMT}, when the 8-bit interval timer operates in the operating mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1}, I_{DD2}, or I_{DD3}, and I_{WDT} when the watchdog timer operates.

6. Current flowing only to the LVD circuit. The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1}, I_{DD2}, or I_{DD3}, and I_{LVD} when the LVD circuit operates.

7. Current flowing only to the 12-bit A/D converter. The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{ADC} when the 12-bit A/D converter operates in the operating mode or HALT mode.

8. Current flowing to AV_{DD}.

9. Current flowing from the reference voltage source of the 12-bit A/D converter.

10. Current flowing to AV_{REFP}.

11. Current flowing only during rewrite of 1 KB data flash memory.

12. Current flowing only during self programming.

13. Current flowing only to the 24-bit ΔΣ A/D converter. The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{DSAD} when the 24-bit ΔΣ A/D converter operates.

14. For shift time to the SNOOZE mode, see **29.3.3 SNOOZE mode**.

15. Current flowing only to the LCD controller/driver. The supply of current to an RL78 microcontroller is the sum of the supply current (I_{DD1} or I_{DD2}) and the LCD operating current (I_{LCD1}, I_{LCD2}, or I_{LCD3}) when the LCD controller/driver operates in the operating mode or HALT mode. Not including the current that flows through the LCD panel. Conditions of the TYP. value and MAX. value are as follows.

- Setting 20 pins as the segment function and blinking all
- Selecting f_{SUB} for system clock when LCD clock = 128 Hz (LCDC0 = 07H)
- Setting four time slices and 1/3 bias

16. Not including the current flowing into the external division resistor when using the external resistance division method.

17. Current flowing only to the timer RJ (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{TMRJ}, when the timer RJ operates in the operating mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

18. Current flowing only to the serial interface UARTMG (excluding the operating current of the XT1 oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{UARTMG}, when the serial interface UARTMG operates in the operating mode or HALT mode.

19. Current flowing only to the sampling output timer detector (excluding the operating current of the XT1 oscillator). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and I_{SMOTD}, when the sampling output timer detector operates in the operating mode or HALT mode.

20. Current flowing only to the voltage reference (VREFADC). The supply of current to an RL78 microcontroller is the sum of the values of I_{DD1} or I_{DD2}, and $I_{VREFOUT}$, when the voltage reference operates in the operating mode or HALT mode.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. f_{CLK}: CPU/peripheral hardware clock frequency

4. Temperature condition of the TYP. value is T_A = 25°C

43.4 AC Characteristics

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (1/2)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
				2.1 V ≤ V_{DD} < 2.4 V	0.16667		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
			LS (low-speed main) mode (MCSEL = 1)	1.8 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
			LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V		1		μs
			LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
		Subsystem clock (f_{SUB}) operation	f_{XT} = 38.4 kHz	1.8 V ≤ V_{DD} ≤ 5.5 V		26.0		μs
			f_{XT} = 32.768 kHz	1.8 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
				2.1 V ≤ V_{DD} < 2.4 V	0.16667		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
			LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1		20	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1		16	MHz
		1.8 V ≤ V_{DD} < 2.4 V			1		8	MHz
		1.6 V ≤ V_{DD} < 1.8 V			1		4	MHz
	f_{EXS}	f_{EX} = 38.4 kHz			31		39	kHz
		f_{EX} = 32.768 kHz			32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
		1.8 V ≤ V_{DD} < 2.4 V			60			ns
		1.6 V ≤ V_{DD} < 1.8 V			120			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				$1/f_{MCK}+10$			ns
Timer RJ input cycle	t_C	TRJIO0, TRJIO1		2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
				1.8 V ≤ V_{DD} < 2.7 V	300			ns
Timer RJ input high-level width, low-level width	t_{TJIH}, t_{TJIL}	TRJIO0, TRJIO1		2.7 V ≤ V_{DD} ≤ 5.5 V	40			ns
				1.8 V ≤ V_{DD} < 2.7 V	120			ns

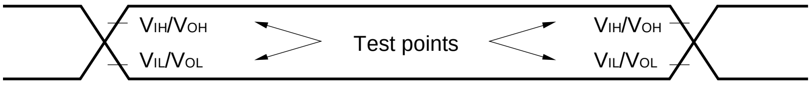
(**Remark** is listed on the next page.)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (2/2)

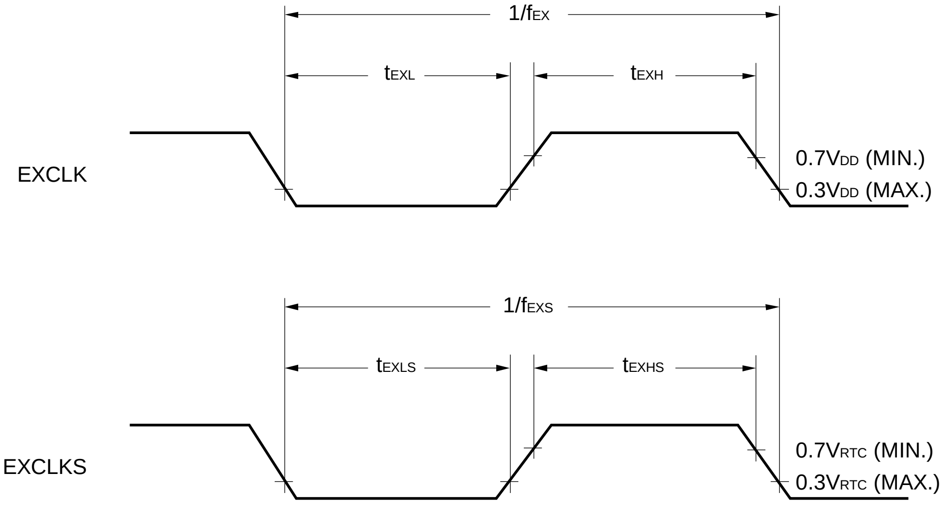
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Timer output frequency	f_{TO}	TO00 to TO07, TRJIO0, TRJIO1, TRJO0, TRJO1	HS (high-speed main) mode	4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
				2.4 V ≤ EV_{DD} < 2.7 V			4	MHz
				2.1 V ≤ EV_{DD} < 2.4 V			4	MHz
			LS (low-speed main) mode	1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
			LP (low-power main) mode	1.8 V ≤ EV_{DD} ≤ 5.5 V			0.5	MHz
			LV (low-voltage main) mode	1.6 V ≤ EV_{DD} ≤ 5.5 V			2	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
				2.4 V ≤ EV_{DD} < 2.7 V			4	MHz
				2.1 V ≤ EV_{DD} < 2.4 V			4	MHz
		LS (low-speed main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
		LP (low-power main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			1	MHz
		LV (low-voltage main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
				1.6 V ≤ EV_{DD} < 1.8 V			2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0, INTP8, INTP9, INTP12 to INTP14		1.6 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP1 to INTP7		1.6 V ≤ EV_{DD} ≤ 5.5 V	1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR7		1.8 V ≤ EV_{DD} ≤ 5.5 V	250			ns
				1.6 V ≤ EV_{DD} < 1.8 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn0, CKSmn1 bits of timer mode register mn (TMRmn)
m: Unit number (m = 0), n: Channel number (n = 0 to 7))

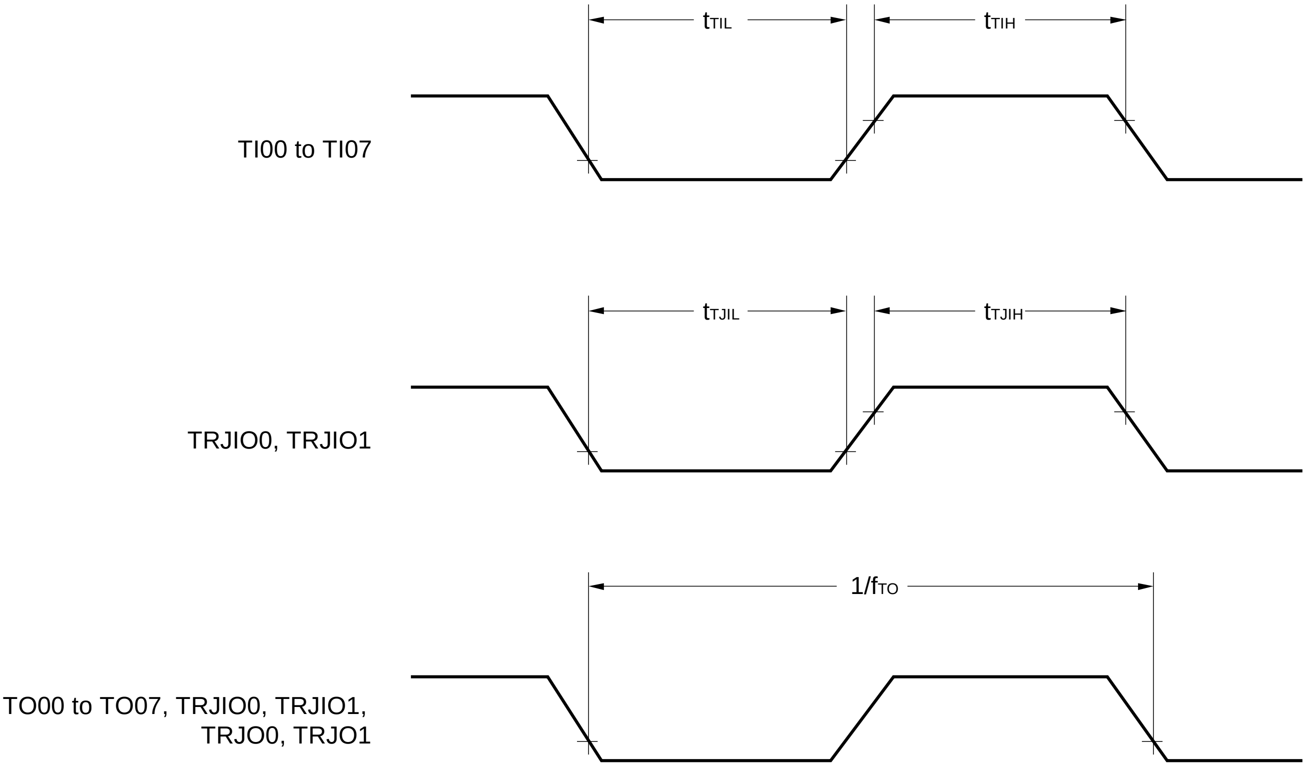
AC Timing Test Points



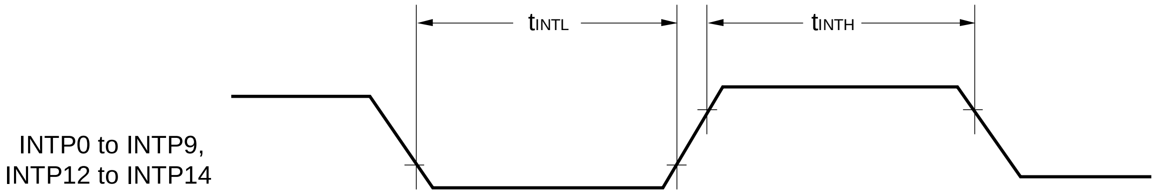
External System Clock Timing



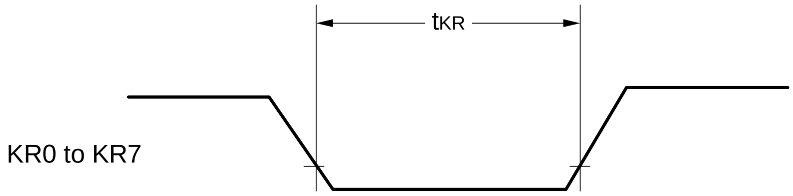
TI/TO Timing



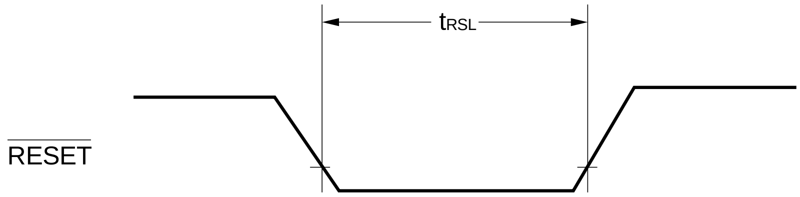
Interrupt Request Input Timing



Key interrupt Input Timing

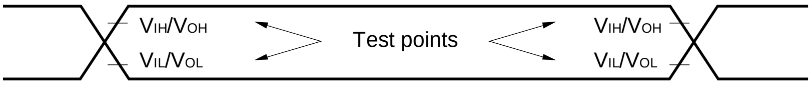


$\overline{\text{RESET}}$ Input Timing



43.5 Peripheral Functions Characteristics

AC Timing Test Points



43.5.1 Serial array unit

(1) During communication at same potential (UART mode) (dedicated baud rate generator output)
(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate[Note 1]		2.7 V ≤ EV_{DD} ≤ 5.5 V			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		5.3		1.3		0.1		0.6	Mbps
		2.4 V ≤ EV_{DD} ≤ 5.5 V			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		4.0		1.3		0.1		0.6	Mbps
		2.1 V ≤ EV_{DD} ≤ 5.5 V			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]		1.0		1.3		0.1		0.6	Mbps
		1.8 V ≤ EV_{DD} ≤ 5.5 V					f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]				1.3		0.1		0.6	Mbps
		1.6 V ≤ EV_{DD} ≤ 5.5 V									f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK}[Note 2]								0.6	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

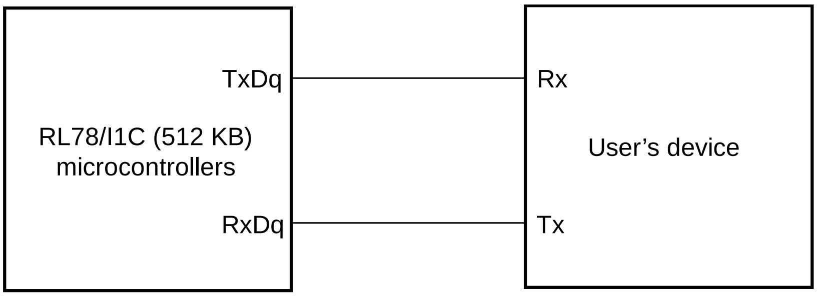
2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode:	32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
	16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
	6 MHz (2.1 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode:	1 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

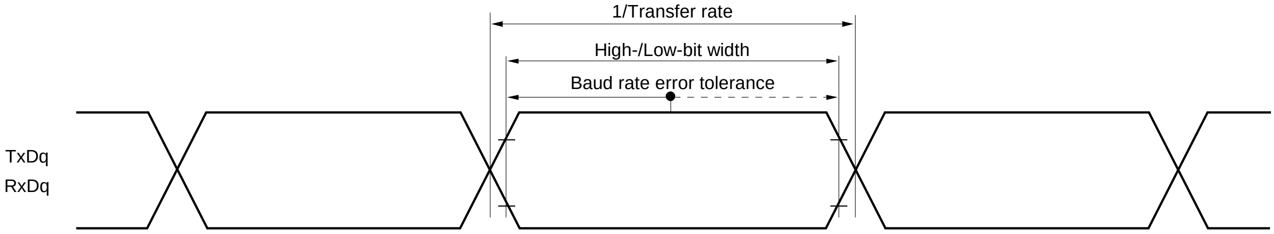
Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

(**Remarks** are listed on the next page.)

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0 to 4), g: PIM and POM number (g = 0, 1, 5, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13, 20, 21))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	2.7 V ≤ EV_{DD} ≤ 5.5 V	125		500		4000		1000		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	250		500		4000		1000		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V	667		500		4000		1000		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			500		4000		1000		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V							1000		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/ 2 – 12		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/ 2 – 18		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/ 2 – 38		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		t_{KCY1}/ 2 – 50		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V							t_{KCY1}/ 2 – 100		ns
SIp setup time (to SCKp↑)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	44		110		110		110		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	44		110		110		110		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	75		110		110		110		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V	110		110		110		110		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			110		110		110		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V							220		ns
SIp hold time (from SCKp↑)Note 2	t_{KSI1}	1.8 V ≤ EV_{DD} ≤ 5.5 V	19		19		19		19		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V							19		ns
Delay time from SCKp↓ to SOp outputNote 3	t_{KSO1}	C = 30 pFNote 3, 1.8 V ≤ EV_{DD} ≤ 5.5 V		25		25		25		25	ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V								25	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” and the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to Sop output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00, 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), g: PIM and POM numbers (g = 0, 1, 5, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)
(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time[Note 4]	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		–		–		–		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		–		–		–		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V		6/f_{MCK} and 500		6/f_{MCK} and 500		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V		6/f_{MCK} and 750		6/f_{MCK} and 750		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V				6/f_{MCK} and 750		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V								6/f_{MCK} and 1500		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		t_{KCY2}/2 – 7		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		t_{KCY2}/2 – 8		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V				t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		t_{KCY2}/2 – 18		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V								t_{KCY2}/2 – 66		ns
SIp setup time (to SCKp↑)[Note 1]	t_{SIK2}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+20		1/f_{MCK}+30		1/f_{MCK}+30		1/f_{MCK}+30		ns
		2.1 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+30		1/f_{MCK}+30		1/f_{MCK}+30		1/f_{MCK}+30		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V				1/f_{MCK}+30		1/f_{MCK}+30		1/f_{MCK}+30		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V								1/f_{MCK}+40		ns
SIp hold time (from SCKp↑)[Note 1]	t_{KSI2}	2.1 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK}+31		1/f_{MCK}+31		1/f_{MCK}+31		1/f_{MCK}+31		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V				1/f_{MCK}+31		1/f_{MCK}+31		1/f_{MCK}+31		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V								1/f_{MCK}+250		ns
Delay time from SCKp↓ to SOp output[Note 2]	t_{KSO2}	C = 30 pF[Note 3]	2.7 V ≤ EV_{DD} ≤ 5.5 V		2/f_{MCK}+ 44		2/f_{MCK}+ 110		2/f_{MCK}+ 110		2/f_{MCK}+ 110	ns
			2.4 V ≤ EV_{DD} ≤ 5.5 V		2/f_{MCK}+ 75		2/f_{MCK}+ 110		2/f_{MCK}+ 110		2/f_{MCK}+ 110	ns
			2.1 V ≤ EV_{DD} ≤ 5.5 V		2/f_{MCK}+ 100		2/f_{MCK}+ 110		2/f_{MCK}+ 110		2/f_{MCK}+ 110	ns
			1.8 V ≤ EV_{DD} ≤ 5.5 V				2/f_{MCK}+ 110		2/f_{MCK}+ 110		2/f_{MCK}+ 110	ns
			1.6 V ≤ EV_{DD} ≤ 5.5 V								2/f_{MCK}+ 220	ns

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

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” and the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to Sop output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. C is the load capacitance of the SOp output lines.

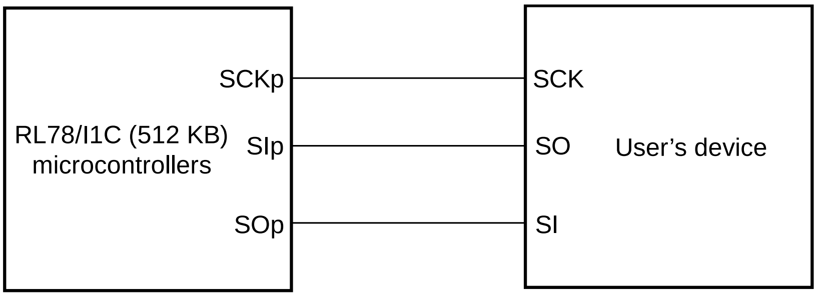
4. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

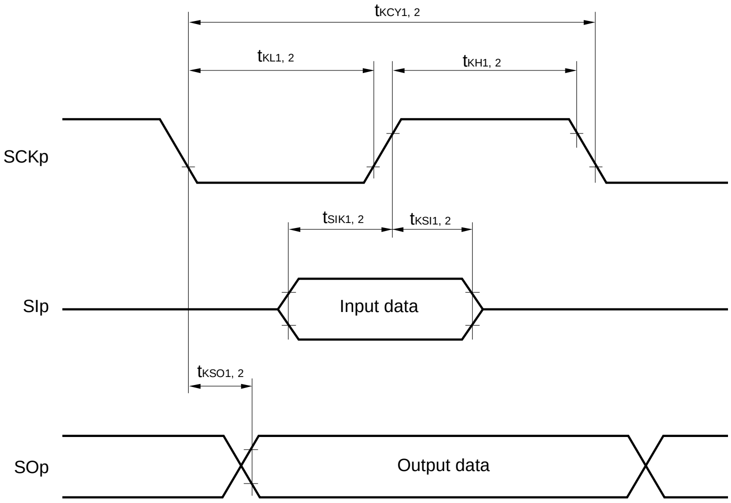
Remarks 1. p: CSI number (p = 00, 10, 30), m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), g: PIM number (g = 0, 1, 5, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

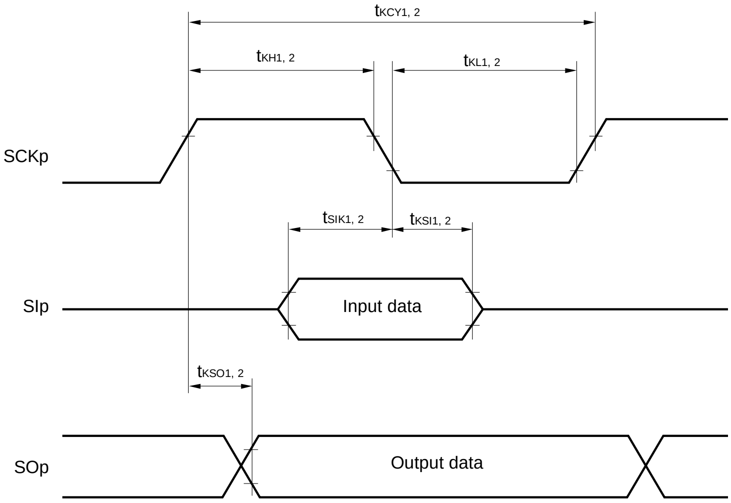
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Remarks 1. p: CSI number (p = 00, 10, 30)

2. m: Unit number, n: Channel number (mn = 00, 02, 12)

(4) During communication at same potential (simplified I^{2}C mode)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000[Note 1]		400[Note 1]		400[Note 1]		400[Note 1]	kHz
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400[Note 1]		400[Note 1]		400[Note 1]		400[Note 1]	kHz
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ				300[Note 1]		300[Note 1]		300[Note 1]	kHz
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ								250[Note 1]	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ			1550		1550		1550		ns
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ							1850		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ			1550		1550		1550		ns
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ							1850		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 145 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		1/f_{MCK} + 145 Notes 1, 2		ns
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ			1/f_{MCK} + 230 Notes 1, 2		1/f_{MCK} + 230 Notes 1, 2		1/f_{MCK} + 230 Notes 1, 2		ns
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ							1/f_{MCK} + 290 Notes 1, 2		ns

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

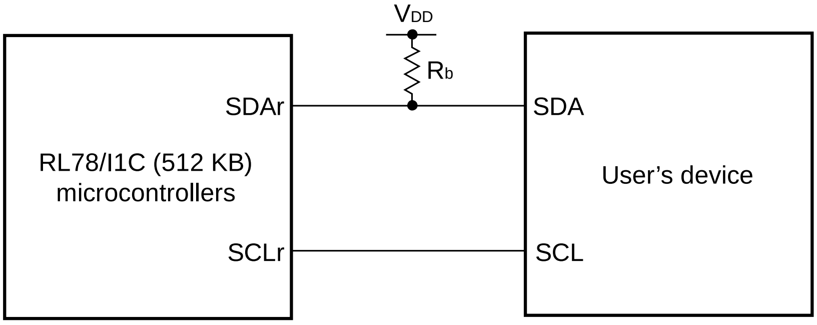
(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	0	305	ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ			0	405	0	405	0	405	ns
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ							0	405	ns

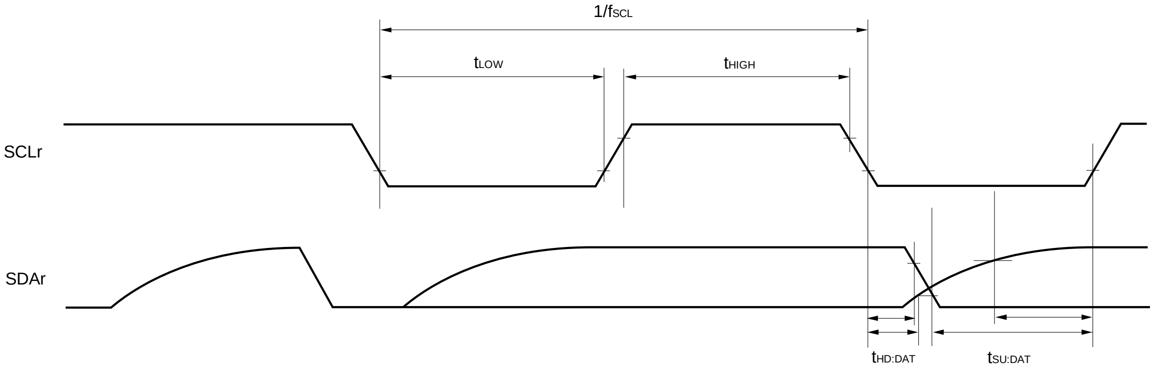
Notes 1. The value must also be equal to or less than f_{MCK}/4.

2. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

Simplified I²C mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Caution Select the normal input buffer and the N-ch open drain output (EV_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

2. r: IIC number (r = 00, 10, 30), g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12))

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (1/2)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate		Reception	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$[Note 3]		5.3		1.3		0.1		0.6	Mbps
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]		$f_{MCK}/6$[Note 1]	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$[Note 3]		5.3		1.3		0.1		0.6	Mbps
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$[Note 3]		5.3		1.3		0.1		0.6	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. Use it with EV_{DD} ≥ V_b.

3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode:	32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), 16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V), 6 MHz (2.1 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode:	1 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{DD} tolerance) mode for the TxDq pin by using 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 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0 to 4), g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13, 20, 21))

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (2/2)
(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate		Transmission	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Notes 1, 2**		**Notes 1, 2**		**Notes 1, 2**		**Notes 1, 2**	bps
			Theoretical value of the maximum transfer rate[Note 9] C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8[Note 3]		2.8[Note 3]		2.8[Note 3]		2.8[Note 3]	Mbps
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		**Notes 2, 4**		**Notes 2, 4**		**Notes 2, 4**		**Notes 2, 4**	bps
			Theoretical value of the maximum transfer rate[Note 9] C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2[Note 5]		1.2[Note 5]		1.2[Note 5]		1.2[Note 5]	Mbps
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		**Notes 2, 6, 7**		**Notes 2, 6, 7**		**Notes 2, 6, 7**		**Notes 2, 6, 7**	bps
			Theoretical value of the maximum transfer rate[Note 9] C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43[Note 8]		0.43[Note 8]		0.43[Note 8]		0.43[Note 8]	Mbps

Notes 1. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. Transfer rate in the SNOOZE mode is 4800 bps only.

3. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

4. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ EV_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Notes 5. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 4** above to calculate the maximum transfer rate under conditions of the customer.

6. Use it with $EV_{DD} \geq V_b$.

7. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when $1.8\ V \leq EV_{DD} < 3.3\ V$ and $1.6\ V \leq V_b \leq 2.0\ V$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3}\ [\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

8. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 7** above to calculate the maximum transfer rate under conditions of the customer.

9. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode:	32 MHz ($2.7\ V \leq V_{DD} \leq 5.5\ V$),
	16 MHz ($2.4\ V \leq V_{DD} \leq 5.5\ V$),
	6 MHz ($2.1\ V \leq V_{DD} \leq 5.5\ V$)
LS (low-speed main) mode:	8 MHz ($1.8\ V \leq V_{DD} \leq 5.5\ V$)
LP (low-power main) mode:	1 MHz ($1.8\ V \leq V_{DD} \leq 5.5\ V$)
LV (low-voltage main) mode:	4 MHz ($1.6\ V \leq V_{DD} \leq 5.5\ V$)

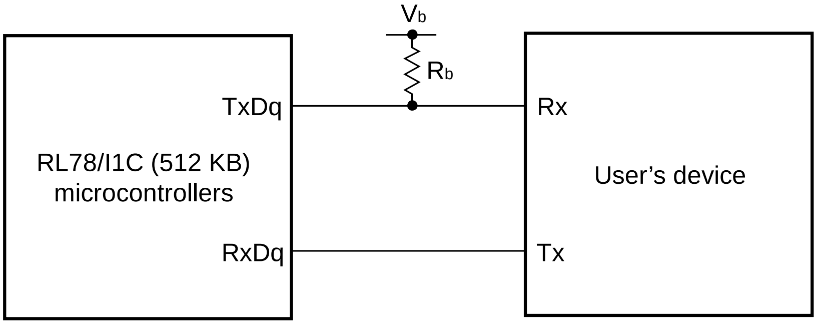
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. $R_b[\Omega]$:Communication line (TxDq) pull-up resistance,
$C_b[F]$: Communication line (TxDq) load capacitance, $V_b[V]$: Communication line voltage

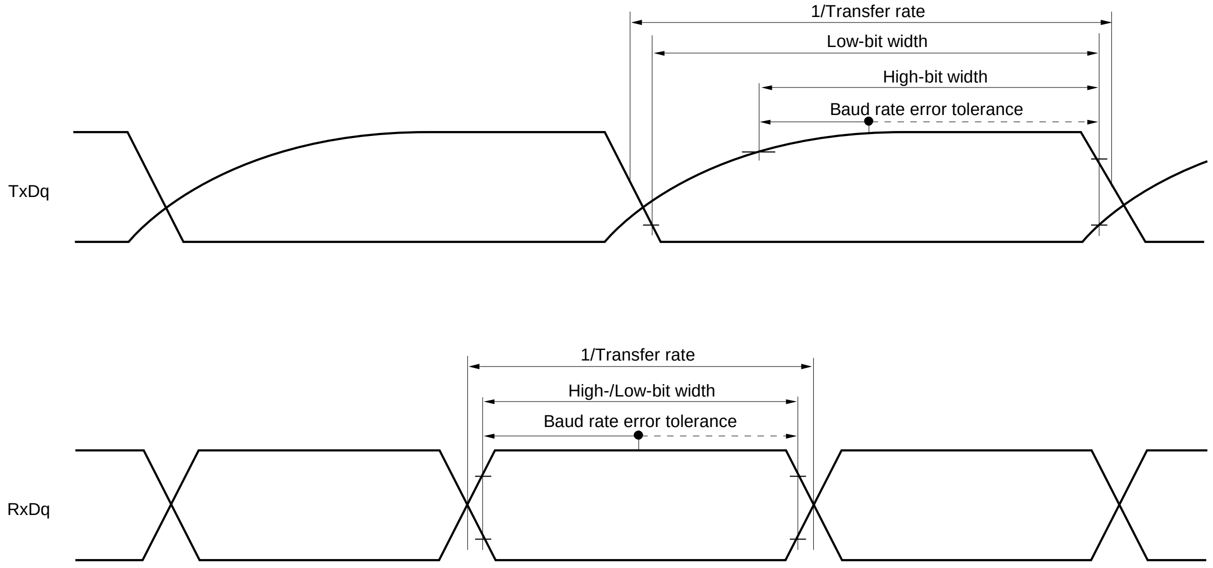
2. q: UART number (q = 0 to 4), g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13, 20, 21))

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV$_{DD}$ tolerance) mode for the TxDq pin by using 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 1. R$_b$[Ω]: Communication line (TxDq) pull-up resistance, V$_b$[V]: Communication line voltage

2. q: UART number (q = 0 to 4), g: PIM and POM number (g = 0, 1, 5, 8)

(6) Communication at different potential (2.5 V, 3 V) (f_{MCK}/2) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only) (1/2)
(T_A = –40 to +85°C, 2.7 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK} 4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		1150		1150		1150		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		1150		1150		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 120		t_{KCY1}/2 – 120		t_{KCY1}/2 – 120		t_{KCY1}/2 – 120		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 7		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 10		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	58		479		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	121		479		479		479		ns
SIp hold time (from SCKp↑)Note 1	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	10		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	10		10		10		10		ns
Delay time from SCKp↓ to SOp outputNote 1	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		60		60		60		60	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		130		130		130		130	ns
SIp setup time (to SCKp↓)Note 2	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	23		110		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	33		110		110		110		ns

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

(6) Communication at different potential (2.5 V, 3 V) (f_{MCK}/2) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only) (2/2)
(T_A = –40 to +85°C, 2.7 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

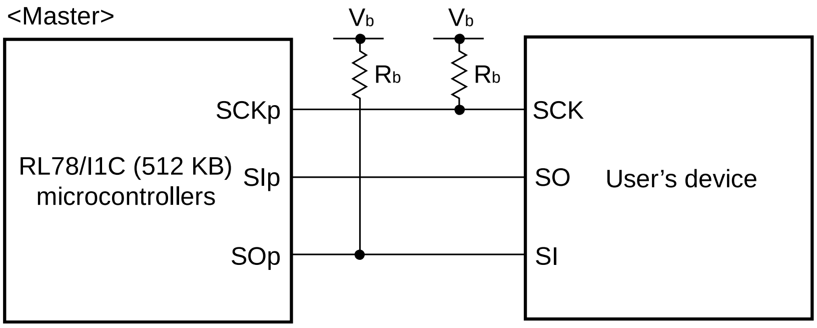
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SIp hold time (from SCKp↓)[Note 2]	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	10		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	10		10		10		10		ns
Delay time from SCKp↑ to SOp output[Note 2]	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		10	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		10	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV_{DD} tolerance) mode for the SOp pin and SCKp pin by using 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.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00, 10, 30), m: Unit number, n: Channel number (mn = 00, 02, 12), g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

4. This specification is valid only when CSI00's peripheral I/O redirect function is not used.

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (f_{MCK}/4) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (1/2)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		1150		1150		ns
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		1150		1150		ns
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		1150		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 75		t_{KCY1}/2 – 75		t_{KCY1}/2 – 75		t_{KCY1}/2 – 75		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 12		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 18		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		81		479		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		177		479		479		479		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		479		479		479		479		ns

(**Notes**, **Caution**, and **Remarks** are listed on the page after the next page.)

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (f_{MCK}/4) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (2/2)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SIp hold time (from SCKp↑)[Note 1]	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		19		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		19		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 3], C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		19		ns
Delay time from SCKp↓ to SOp output[Note 1]	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100		100		100	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195		195	ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 3], C_b = 30 pF, R_b = 5.5 kΩ		483		483		483		483	ns
SIp setup time (to SCKp↓)[Note 2]	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		110		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		110		110		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 3], C_b = 30 pF, R_b = 5.5 kΩ	110		110		110		110		ns
SIp hold time (from SCKp↓)[Note 2]	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		19		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		19		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 3], C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		19		ns
Delay time from SCKp↑ to SOp output[Note 2]	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25		25	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25		25	ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 3], C_b = 30 pF, R_b = 5.5 kΩ		25		25		25		25	ns

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

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. Use it with $EV_{DD} \geq V_b$.

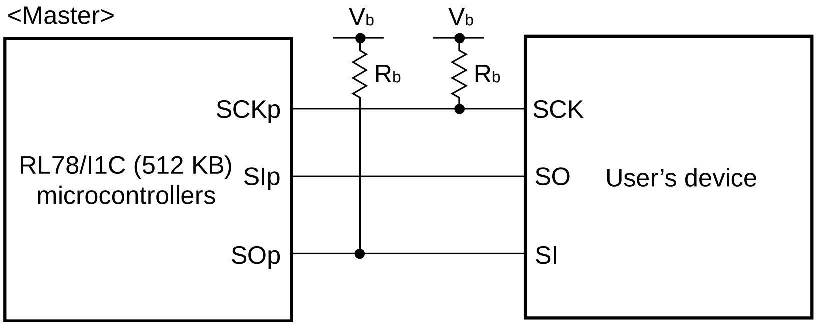
Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

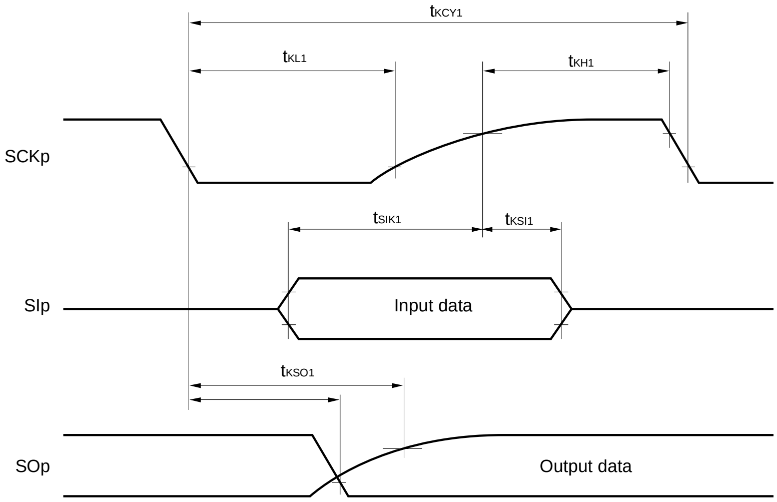
2. p: CSI number (p = 00, 10, 30), m: Unit number , n: Channel number (mn = 00, 02, 12), g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

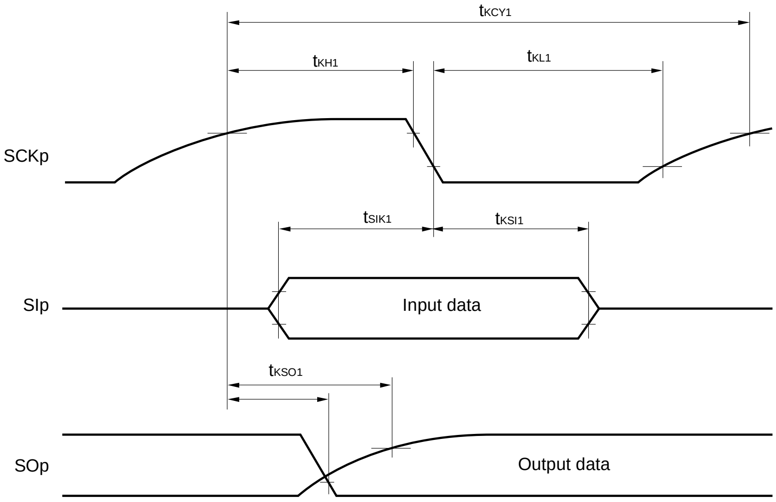
Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV$_{DD}$ tolerance) mode for the SOp pin and SCKp pin by using 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.

Remark p: CSI number (p = 00, 10, 30), m: Unit number, n: Channel number (mn = 00, 02, 12), g: PIM and POM number (g = 0, 1, 5, 8)

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (slave mode, SCKp ... external clock input)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time[Note 1]	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	14/f_{MCK}		–		–		–		ns
			20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		–		–		–		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		–		–		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		–		–		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	24 MHz < f_{MCK}	20/f_{MCK}		–		–		–		ns
			20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		–		–		–		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		–		–		–		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		–		–		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		–		–		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2]	24 MHz < f_{MCK}	48/f_{MCK}		–		–		–		ns
			20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		–		–		–		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		–		–		–		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		–		–		–		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		–		–		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 12		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 18		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2]		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		t_{KCY2}/2 – 50		ns
SIp setup time (to SCKp↑)[Note 3]	t_{SIK2}	2.7 V ≤ EV_{DD} ≤ 5.5 V, 2.3 V ≤ V_b ≤ 4.0 V[Note 2]		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2]		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑)[Note 3]	t_{KSI2}	2.7 V ≤ EV_{DD} ≤ 5.5 V, 2.3 V ≤ V_b ≤ 4.0 V[Note 2]		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2]		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns

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

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (slave mode, SCKp ... external clock input)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Delay time from SCKp↓ to SOp output[Note 4]	t_{KSO2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		2/f_{MCK} + 120		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		2/f_{MCK} + 214		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2], C_b = 30 pF, R_b = 5.5 kΩ		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Notes 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

2. Use it with EV_{DD} ≥ V_b.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” and the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to Sop output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

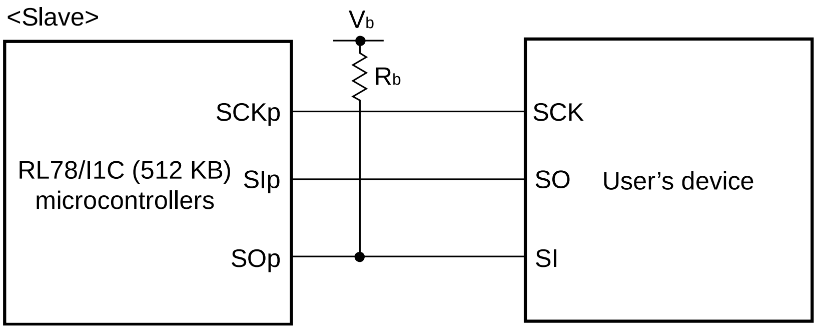
Caution Select the TTL input buffer for the SIp pin and SCKp pin and the N-ch open drain output (EV_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. R_b[Ω]:Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

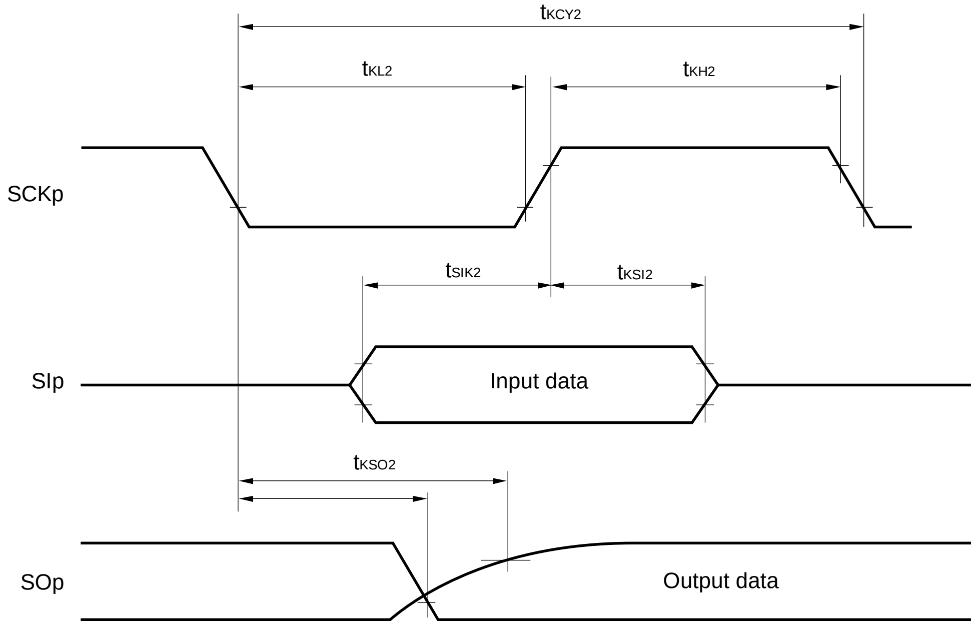
2. p: CSI number (p = 00, 10, 30), m: Unit number, n: Channel number (mn = 00, 02, 12),
g: PIM and POM number (g = 0, 1, 5, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

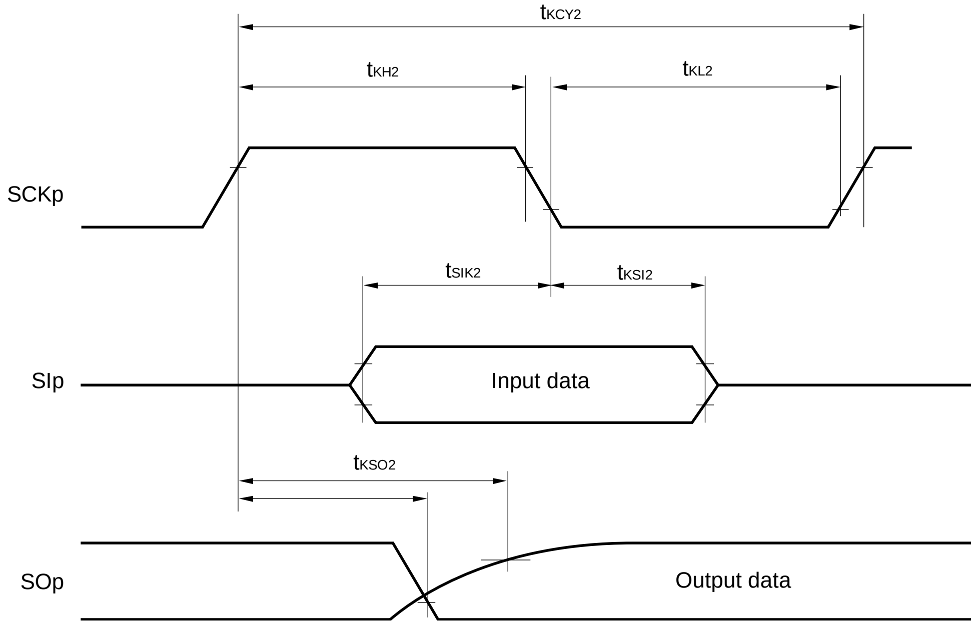
Simplified SPI (CSI) mode connection diagram (during communication at different potential)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for the SIp pin and SCKp pin and the N-ch open drain output (EV$_{DD}$ tolerance) mode for the SOp pin by using 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.

Remark p: CSI number (p = 00, 10, 30), m: Unit number, n: Channel number (mn = 00, 02, 12),
g: PIM and POM number (g = 0, 1, 5, 8)

(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^2C mode) (1/2)
(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000[Note 1]		300[Note 1]		300[Note 1]		300[Note 1]	kHz
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000[Note 1]		300[Note 1]		300[Note 1]		300[Note 1]	kHz
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400[Note 1]		300[Note 1]		300[Note 1]		300[Note 1]	kHz
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400[Note 1]		300[Note 1]		300[Note 1]		300[Note 1]	kHz
		1.8 V ≤ EV_{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]		300[Note 1]	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		1550		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		1550		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1150		1150		1150		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V[Note 2], C_b = 100 pF, R_b = 5.5 kΩ	1150		1150		1150		1150		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		610		610		610		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		610		610		610		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		610		610		610		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		610		610		610		ns
		1.8 V ≤ EV_{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		610		ns

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

(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^{2}C mode) (2/2)

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ EV_{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} + 190[Note 3]		1/f_{MCK} + 190[Note 3]		1/f_{MCK} + 190[Note 3]		ns
		2.7 V ≤ EV_{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} + 190[Note 3]		1/f_{MCK} + 190[Note 3]		1/f_{MCK} + 190[Note 3]		ns
		4.0 V ≤ EV_{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]		1/f_{MCK} + 190[Note 3]		ns
		2.7 V ≤ EV_{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]		1/f_{MCK} + 190[Note 3]		ns
		1.8 V ≤ EV_{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]		1/f_{MCK} + 190[Note 3]		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ EV_{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	0	305	ns
		2.7 V ≤ EV_{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	0	305	ns
		4.0 V ≤ EV_{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	0	355	ns
		2.7 V ≤ EV_{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	0	355	ns
		1.8 V ≤ EV_{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	0	405	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

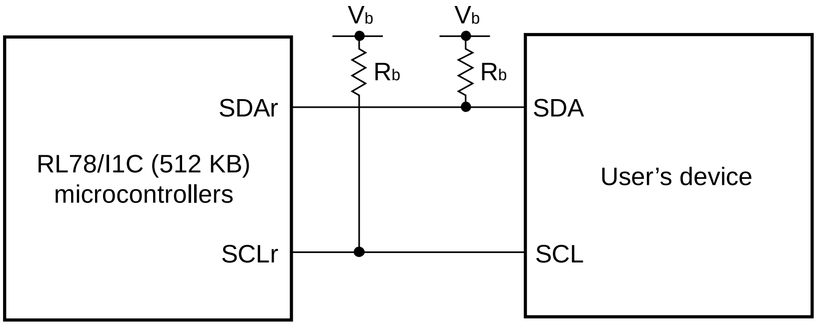
2. Use it with EV_{DD} ≥ V_b.

3. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

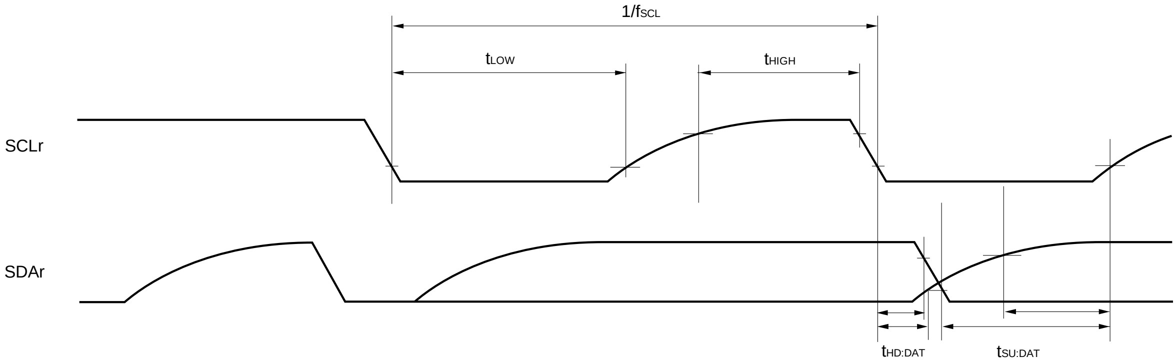
Caution Select the TTL input buffer and the N-ch open drain output (EV_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (EV_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified I^{2}C mode connection diagram (during communication at different potential)



Simplified I^{2}C mode serial transfer timing (during communication at different potential)



Caution **Select the TTL input buffer and the N-ch open drain output (EV$_{DD}$ tolerance) mode for the SDAr pin and the N-ch open drain output (EV$_{DD}$ tolerance) mode for the SCLr pin by using 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 1. R$_b$[Ω]:Communication line (SDAr, SCLr) pull-up resistance, C$_b$[F]: Communication line (SDAr, SCLr) load capacitance, V$_b$[V]: Communication line voltage

2. r: IIC number (r = 00, 10, 30), g: PIM, POM number (g = 0, 1, 5, 8)

3. f$_{MCK}$: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 02, 12))

43.5.2 Serial interface UARTMG

(T$_A$ = –40 to +85°C, 1.8 V ≤ AV$_{DD}$ = EV$_{DD}$ = V$_{DD}$ ≤ 5.5 V, AV$_{SS}$ = V$_{SS}$ = EV$_{SS}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		f$_{SX}$ = 38.4 kHz	200		9600	bps
		f$_{SX}$ = 38.4 kHz (when the clock doubler is in use)	200		19200	bps

43.5.3 Serial interface IICA

(1) I^2C standard mode (1/2)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V	0	100	0	100	0	100	0	100	kHz
			2.1 V ≤ EV_{DD} ≤ 5.5 V	0	100	0	100	0	100	0	100	kHz
			1.8 V ≤ EV_{DD} ≤ 5.5 V	–	–	0	100	0	100	0	100	kHz
			1.6 V ≤ EV_{DD} ≤ 5.5 V	–	–	–	–	–	–	0	100	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	4.7		4.7		4.7		µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	4.7		µs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	4.0		4.0		4.0		µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	4.0		µs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	4.7		4.7		4.7		µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	4.7		µs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	4.0		4.0		4.0		µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	4.0		µs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		250		250		250		250		µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		250		250		250		250		µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	250		250		250		µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	250		µs
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0	3.45	0	3.45	0	3.45	0	3.45	µs
		2.1 V ≤ EV_{DD} ≤ 5.5 V		0	3.45	0	3.45	0	3.45	0	3.45	µs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		–	–	0	3.45	0	3.45	0	3.45	µs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		–	–	–	–	–	–	0	3.45	µs

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

(1) I²C standard mode (2/2)

(T_A = –40 to +85°C, 1.6 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ EV_{DD} ≤ 5.5 V	4.0		4.0		4.0		4.0		μs
		2.1 V ≤ EV_{DD} ≤ 5.5 V	4.0		4.0		4.0		4.0		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V	–	–	4.0		4.0		4.0		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V	–	–	–	–	–	–	4.0		μs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD} ≤ 5.5 V	4.7		4.7		4.7		4.7		μs
		2.1 V ≤ EV_{DD} ≤ 5.5 V	4.7		4.7		4.7		4.7		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V	–	–	4.7		4.7		4.7		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V	–	–	–	–	–	–	4.7		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

(2) I^2C fast mode

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = EV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V	0	400	0	400	–	–	0	400	kHz
			1.8 V ≤ EV_{DD} ≤ 5.5 V	0	400	0	400	–	–	0	400	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		–	–	1.3		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		–	–	1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		100		100		–	–	100		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V		100		100		–	–	100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0	0.9	0	0.9	–	–	0	0.9	μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0	0.9	0	0.9	–	–	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		–	–	0.6		μs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		–	–	1.3		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		–	–	1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

(3) I²C fast mode plus

(T_A = –40 to +85°C, 2.7 V ≤ AV$_{DD}$ = EV$_{DD}$ = V$_{DD}$ ≤ 5.5 V, AV$_{SS}$ = V$_{SS}$ = EV$_{SS}$ = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (low-power main) Mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V	0	1000	–	–	–	–	–	–	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.26		–	–	–	–	–	–	μs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.26		–	–	–	–	–	–	μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.5		–	–	–	–	–	–	μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.26		–	–	–	–	–	–	μs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		50		–	–	–	–	–	–	ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0	0.45	–	–	–	–	–	–	μs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.26		–	–	–	–	–	–	μs
Bus-free time	t_{BUF}	2.7 V ≤ EV$_{DD}$ ≤ 5.5 V		0.5		–	–	–	–	–	–	μs

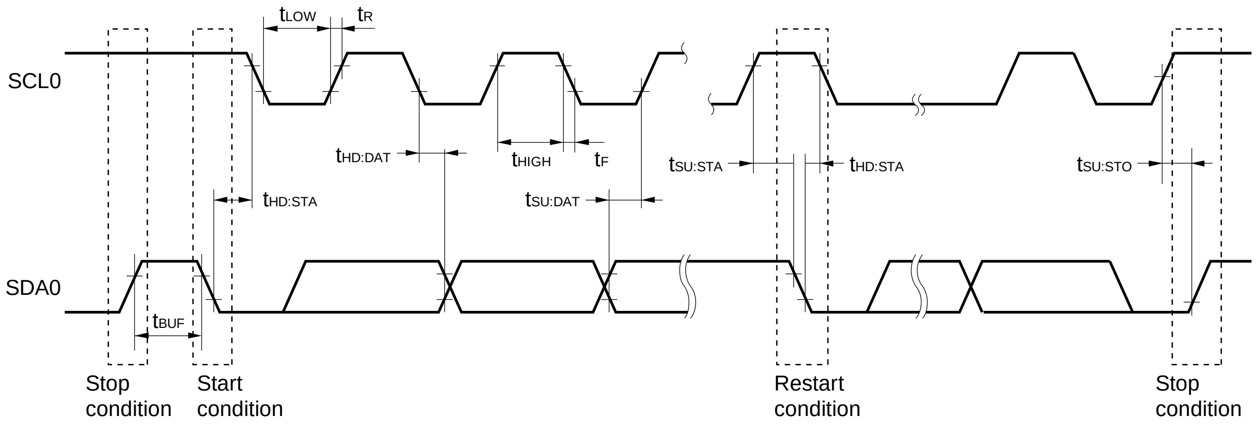
Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



43.6 Analog Characteristics

43.6.1 12-bit A/D converter characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS1} = V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (–) = AV_{REFM} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			–	–	12	bit
Analog capacitance	C_S			–	–	8	pF
Analog input resistance	R_S	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V		–	–	6.7	kΩ
		1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V		–	–	8.2	kΩ
		1.6 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V		–	–	14.3	kΩ
Input capacitance Note 1	Cin	ANI0 to ANI5	V_I = AV_{DD}	–	8	–	pF
Internal reference voltage	V_{BGR}	2.4 V ≤ AV_{DD} ≤ 5.5 V, HS (high-speed main) mode		1.38	1.45	1.5	V
Frequency	ADCLK	High-speed mode	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	1	–	32	MHz
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	1	–	16	MHz
		Normal mode	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	1	–	24	MHz
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	1	–	16	MHz
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	1	–	8	MHz
Conversion time Note 2	t_{CONV}	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 0.5 kΩ ADCLK = 32 MHz When the channel-dedicated sample-and-hold circuits are not in use.	2.3	–	–	μs
			2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 0.5 kΩ ADCLK = 32 MHz When the channel-dedicated sample-and-hold circuits are not in use. Note 3	3.3	–	–	μs
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 1.3 kΩ ADCLK = 16 MHz	4.5	–	–	μs
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 1.1 kΩ ADCLK = 24 MHz When the channel-dedicated sample-and-hold circuits are not in use.	3.4	–	–	μs
			2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 1.1 kΩ ADCLK = 24 MHz When the channel-dedicated sample-and-hold circuits are not in use. Note 3	5	–	–	μs

Notes 1. The value listed to the right is only for reference.

2. The conversion time is the sum of sampling time and comparison time. The values indicated in the table are those in the case of 40 clock cycles of ADCLK per sampling state.

3. When the channel-dedicated sample-and-hold circuits are in use, the input voltages on the ANIn (n = 0, 1, 2) pins must be kept within the following range.
0.25 V ≤ ANIn ≤ AV_{DD} - 0.25 V

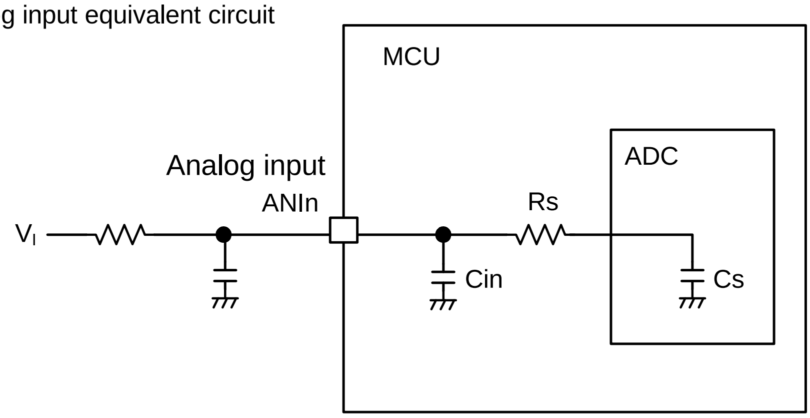
(T_A = –40 to +85°C, 1.8 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS1} = V_{SS} = EV_{SS} = 0 V, reference voltage (+) = AV_{REFP}, reference voltage (–) = AV_{REFM} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Conversion time[Note]	t_{CONV}	Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 2.2 kΩ ADCLK = 16 MHz	5.1	–	–	μs
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V Permissible signal source impedance max = 5 kΩ ADCLK = 8 MHz	10.1	–	–	μs
Overall error	AINL	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.25	±5.0	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.25	±5.0	LSB
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.25	±5.0	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.25	±5.0	LSB
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±3.0	±8.0	LSB
Zero-scale error	EZS	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.5	±4.5	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.5	±4.5	LSB
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.5	±4.5	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.5	±4.5	LSB
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1	±7.5	LSB
Full-scale error	EFS	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.75	±4.5	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.75	±4.5	LSB
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.75	±4.5	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±0.75	±4.5	LSB
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.5	±7.5	LSB
DNL differential linearity error	DLE	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	–	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	–	LSB
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	–	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	–	LSB
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	–	LSB
INL integral linearity error	ILE	High-speed mode ADCSR.ADHSC = 0 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	±3.0	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	±4.5	LSB
		Normal mode ADCSR.ADHSC = 1 ADSSTRn = 28 H	2.7 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	±3.0	LSB
			2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.0	±3.0	LSB
			1.8 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 5.5 V	–	±1.25	±3.0	LSB

(**Caution** is listed on the next page.)

Caution The characteristics above only apply when pins other than those of the 12-bit A/D converter are not in use. Each of the overall error, offset error, full-scale error, DNL differential non-linearity error, and INL integral non-linearity error does not include the quantization error.

Analog input equivalent circuit



43.6.2 Voltage reference characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS1} = V_{SS} = EV_{SS} = 0 V)

Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage range	VREFAMPCNT.VREFADCG[1:0] = 0xb, AV_{DD} ≥ 1.8 V Note 3	1.41	1.5	1.59	V
	VREFAMPCNT.VREFADCG[1:0] = 10b, AV_{DD} ≥ 2.2 V Note 4	1.88	2.0	2.12	
	VREFAMPCNT.VREFADCG[1:0] = 11b, AV_{DD} ≥ 2.7 V Note 5	2.35	2.5	2.65	
Temperature coefficient for VREFOUT Note 1	1.8 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V	–	50	–	ppm/ºC
BGR stabilization time Note 2 (after BGR is enabled)	VREFAMPCNT.BGREN = 1	–	–	150	μs
VREFAMP stabilization time Note 2 (after VREFAMP is enabled)	VREFAMPCNT.VREFADCEN = 1	–	–	1500	μs
Overcurrent detection Note 2	VREFAMPCNT.OLDETEN = 1	–	20	40	mA
Load capacitance Note 1	–	0.75	1	1.25	μF

Notes

1. Connect capacitors as stabilization capacitance between the AV_{REFP}/VREFOUT and AV_{REFM} pins when the voltage reference (VREFADC) is in use.

2. These values are based on simulation. They are not tested at the time of shipment.

3. A value in the range listed to the right cannot be used as the high-potential reference voltage for the 12-bit A/D converter.

4. When a value in the range listed to the right is in use as the high-potential reference voltage for the 12-bit A/D converter, the following conditions must be satisfied.
In normal mode: 2.2 V ≤ AV_{DD} ≤ 5.5 V and ADCLK = 1 MHz to 8 MHz

5. When a value in the range listed to the right is in use as the high-potential reference voltage for the 12-bit A/D converter, the following conditions must be satisfied.
In normal mode: 2.7 V ≤ AV_{DD} ≤ 5.5 V and ADCLK = 1 MHz to 24 MHz
In high-speed mode: 2.7 V ≤ AV_{DD} ≤ 5.5 V and ADCLK = 1 MHz to 32 MHz

43.6.3 24-bit ΔΣ A/D converter characteristics

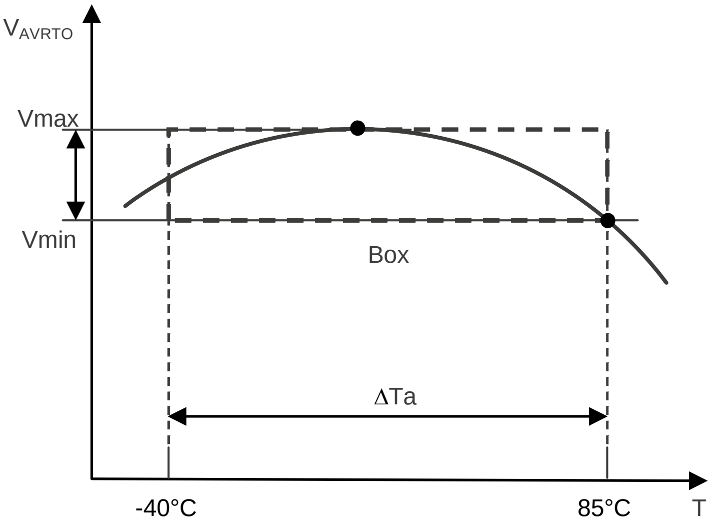
(1) Reference voltage

(T_A = –40 to +85°C, 2.4 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS0} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal reference voltage	V_{AVRTO}			0.8		V
Temperature coefficient for internal reference voltage Note	TC_{BOX}	0.47 µF capacitor connected to AREGC, AVRT, and AVCM pins		10		ppm/°C

Note This is as stipulated by the BOX method.

$$TC_{BOX} = \frac{1}{V_{min}} \cdot \frac{Vmax - Vmin}{\Delta T_a}$$



(2) Analog input

(T_A = –40 to +85°C, 2.4 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS0} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage range (differential voltage)	V_{AIN}	x1 gain	–500		500	mV
		x2 gain	–250		250	
		x4 gain	–125		125	
		x8 gain	–62.5		62.5	
		x16 gain	–31.25		31.25	
		x32 gain	–15.625		15.625	
Input gain	ainGAIN	x1 gain		1		Times
		x2 gain		2		
		x4 gain		4		
		x8 gain		8		
		x16 gain		16		
		x32 gain		32		
Input impedance	ainRIN	Differential voltage	150	360		kΩ
		Single-ended voltage	100	240		

(3) 4 kHz sampling mode

(T_A = –40 to +85°C, 2.4 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS0} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operation clock	f_{DSAD}	f_X oscillation clock, input external clock or high-speed on-chip oscillator clock is used		12		MHz
Sampling frequency	f_S			3906.25		Hz
Oversampling frequency	f_{OS}			1.5		MHz
Output data rate	T_{DATA}			256		µs
Data width	RES			24		bit
SNDR	SNDR	x1 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)		80		dB
		x16 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	69	74		
		x32 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	65	69		
Passband (low pass band)	f_{Chpf}	At –3 dB (phase in high pass filter not adjusted) Bits 7 and 6 of DSADHPFCR register (DSADCOF1, DSADCOF0) = 00		0.607		Hz
		At –3 dB (phase in high pass filter not adjusted) Bits 7 and 6 of DSADHPFCR register (DSADCOF1, DSADCOF0) = 01		1.214		Hz
		At –3 dB (phase in high pass filter not adjusted) Bits 7 and 6 of DSADHPFCR register (DSADCOF1, DSADCOF0) = 10		2.429		Hz
		At –3 dB (phase in high pass filter not adjusted) Bits 7 and 6 of DSADHPFCR register (DSADCOF1, DSADCOF0) = 11		4.857		Hz
In-band ripple 1	rp1	45 Hz to 55 Hz @50 Hz 54 Hz to 66 Hz @60 Hz	–0.01		0.01	dB
In-band ripple 2	rp2	45 Hz to 275 Hz @50 Hz 54 Hz to 330 Hz @60 Hz	–0.1		0.1	
In-band ripple 3	rp3	45 Hz to 1100 Hz @50 Hz 54 Hz to 1320 Hz @60 Hz	–0.1		0.1	
Passband (high pass band)	f_{Clpf}	–3 dB		1672		Hz
Stopband (high pass band)	fatt	–80 dB		2545		Hz
Out-band attenuation	ATT1	f_S	–80			dB
	ATT2	2 f_S	–80			dB

(4) 2 kHz sampling mode

(T_A = –40 to +85°C, 2.4 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS0} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operation clock	f_{DSAD}	f_X oscillation clock, input external clock or high-speed on-chip oscillator clock is used		12		MHz
Sampling frequency	f_S			1953.125		Hz
Oversampling frequency	f_{OS}			0.75		MHz
Output data rate	T_{DATA}			512		µs
Data width	RES			24		bit
SNDR	SNDR	x1 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)		80		dB
		x16 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	69	74		
		x32 gain High-speed system clock is selected as operating clock of 24-bit ΔΣ A/D converter (bit 0 of PCKC register (DSADCK) = 1)	65	69		
Passband (low pass band)	f_{Chpf}	At –3 dB (phase in high pass filter not adjusted)		0.303		Hz
In-band ripple 1	rp1	45 Hz to 55 Hz @50 Hz 54 Hz to 66 Hz @60 Hz	–0.01		0.01	dB
In-band ripple 2	rp2	45 Hz to 275 Hz @50 Hz 54 Hz to 330 Hz @60 Hz	–0.1		0.1	
In-band ripple 3	rp3	45 Hz to 660 Hz @50 Hz 54 Hz to 550 Hz @60 Hz	–0.1		0.1	
Passband (high pass band)	f_{Clpf}	–3 dB		836		Hz
Stopband (high pass band)	fatt	–80 dB		1273		Hz
Out-band attenuation	ATT1	f_S	–80			dB
	ATT2	2 f_S	–80			dB

43.6.4 Temperature sensor 2 characteristics

(T_A = –40 to +85°C, 2.4 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS1} = V_{SS} = EV_{SS} = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor 2 output voltage	V_{OUT}			0.67		V
Temperature coefficient	F_{VTMPS2}	Temperature sensor that depends on the temperature	–11.7	–10.7	–9.7	mV/°C
Operation stabilization wait time	t_{TMPON}	Operable		15	50	µs
	t_{TMPCHG}	Switching mode		5	15	µs
Sampling time	-		5			µs

43.6.5 POR circuit characteristics

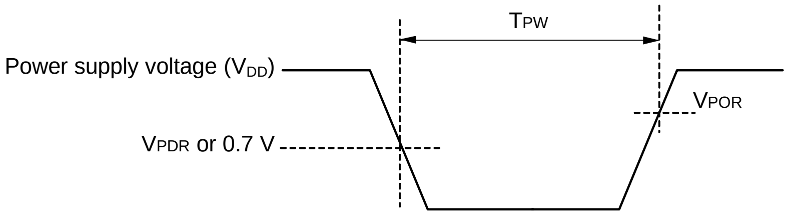
(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	The power supply voltage is rising.Note 1	1.47	1.51	1.55	V
	V_{PDR}	The power supply voltage is falling.Note 2	1.46	1.50	1.54	V
Minimum pulse widthNote 3	T_{PW}		300			μs

Notes 1. Be sure to maintain the reset state until the power supply voltage rises over the minimum V_{DD} value in the operating voltage range specified in **43.4 AC Characteristics**, by using the voltage detector or external reset pin.

2. If the power supply voltage falls while the voltage detector is off, be sure to either shift to STOP mode or execute a reset by using the voltage detector or external reset pin before the power supply voltage falls below the minimum operating voltage specified in **43.4 AC Characteristics**.

3. This is the minimum time required for a power-on reset (POR) when V_{DD} has fallen below V_{PDR}. This is also the minimum time required for a POR following V_{DD} falling below 0.7 V to when V_{DD} exceeds V_{POR} while the chip is in STOP mode or the main system clock is stopped by the settings of bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



43.6.6 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +85°C, $V_{PDR} \leq AV_{DD} = V_{DD} = EV_{DD} \leq 5.5$ V, $AV_{SS} = V_{SS} = EV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVD0}	The power supply voltage is rising.	3.98	4.06	4.24	V
		The power supply voltage is falling.	3.90	3.98	4.16	V
	V_{LVD1}	The power supply voltage is rising.	3.68	3.75	3.92	V
		The power supply voltage is falling.	3.60	3.67	3.84	V
	V_{LVD2}	The power supply voltage is rising.	3.07	3.13	3.29	V
		The power supply voltage is falling.	3.00	3.06	3.22	V
	V_{LVD3}	The power supply voltage is rising.	2.96	3.02	3.18	V
		The power supply voltage is falling.	2.90	2.96	3.12	V
	V_{LVD4}	The power supply voltage is rising.	2.86	2.92	3.07	V
		The power supply voltage is falling.	2.80	2.86	3.01	V
	V_{LVD5}	The power supply voltage is rising.	2.76	2.81	2.97	V
		The power supply voltage is falling.	2.70	2.75	2.91	V
	V_{LVD6}	The power supply voltage is rising.	2.66	2.71	2.86	V
		The power supply voltage is falling.	2.60	2.65	2.80	V
	V_{LVD7}	The power supply voltage is rising.	2.56	2.61	2.76	V
		The power supply voltage is falling.	2.50	2.55	2.70	V
	V_{LVD8}	The power supply voltage is rising.	2.45	2.50	2.65	V
		The power supply voltage is falling.	2.40	2.45	2.60	V
	V_{LVD9}	The power supply voltage is rising.	2.05	2.09	2.23	V
		The power supply voltage is falling.	2.00	2.04	2.18	V
	V_{LVD10}	The power supply voltage is rising.	1.94	1.98	2.12	V
		The power supply voltage is falling.	1.90	1.94	2.08	V
	V_{LVD11}	The power supply voltage is rising.	1.84	1.88	2.01	V
		The power supply voltage is falling.	1.80	1.84	1.97	V
	V_{LVD12}	The power supply voltage is rising.	1.74	1.77	1.81	V
		The power supply voltage is falling.	1.70	1.73	1.77	V
	V_{LVD13}	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	t_{LW}		300			μs
Detection delay time					300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +85°C, V_{PDR} ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVD13}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 0, falling reset voltage			1.60	1.63	1.66	V
	V_{LVD12}		LVIS1, LVIS0 = 1, 0 (+0.1 V)	Rising release reset voltage	1.74	1.77	1.81	V
				Falling interrupt voltage	1.70	1.73	1.91	V
	V_{LVD11}		LVIS1, LVIS0 = 0, 1 (+0.2 V)	Rising release reset voltage	1.84	1.88	1.87	V
				Falling interrupt voltage	1.80	1.84	2.97	V
	V_{LVD4}		LVIS1, LVIS0 = 0, 0 (+1.2 V)	Rising release reset voltage	2.86	2.92	2.91	V
				Falling interrupt voltage	2.80	2.86	3.22	V
	V_{LVD11}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage			1.80	1.84	1.97	V
	V_{LVD10}		LVIS1, LVIS0 = 1, 0 (+0.1 V)	Rising release reset voltage	1.94	1.98	2.12	V
				Falling interrupt voltage	1.90	1.94	2.08	V
	V_{LVD9}		LVIS1, LVIS0 = 0, 1 (+0.2 V)	Rising release reset voltage	2.05	2.09	2.23	V
				Falling interrupt voltage	2.00	2.04	2.18	V
	V_{LVD2}		LVIS1, LVIS0 = 0, 0 (+1.2 V)	Rising release reset voltage	3.07	3.13	3.29	V
				Falling interrupt voltage	3.00	3.06	3.22	V
	V_{LVD8}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage			2.40	2.45	2.60	V
	V_{LVD7}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.56	2.61	2.76	V
				Falling interrupt voltage	2.50	2.55	2.70	V
	V_{LVD6}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.66	2.71	2.86	V
				Falling interrupt voltage	2.60	2.65	2.80	V
	V_{LVD1}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.68	3.75	3.92	V
				Falling interrupt voltage	3.60	3.67	3.84	V
	V_{LVD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage			2.70	2.75	2.91	V
	V_{LVD4}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.86	2.92	3.07	V
				Falling interrupt voltage	2.80	2.86	3.01	V
	V_{LVD3}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.96	3.02	3.18	V
				Falling interrupt voltage	2.90	2.96	3.12	V
	V_{LVD0}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.98	4.06	4.24	V
				Falling interrupt voltage	3.90	3.98	4.16	V

43.6.7 Power supply voltage rising slope characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	SV_{DDR} SV_{RTCR}				54	V/ms

Cautions 1. Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 43.4 AC Characteristics.

2. When the voltages for V_{DD} and AV_{DD} differ and they rise at different rates, if AV_{DD} is lower than 0.8 V at the time of the release from the internal reset state by the power-on reset (POR) circuit, the chip may not start normally.

In such cases, apply either of the following countermeasures.

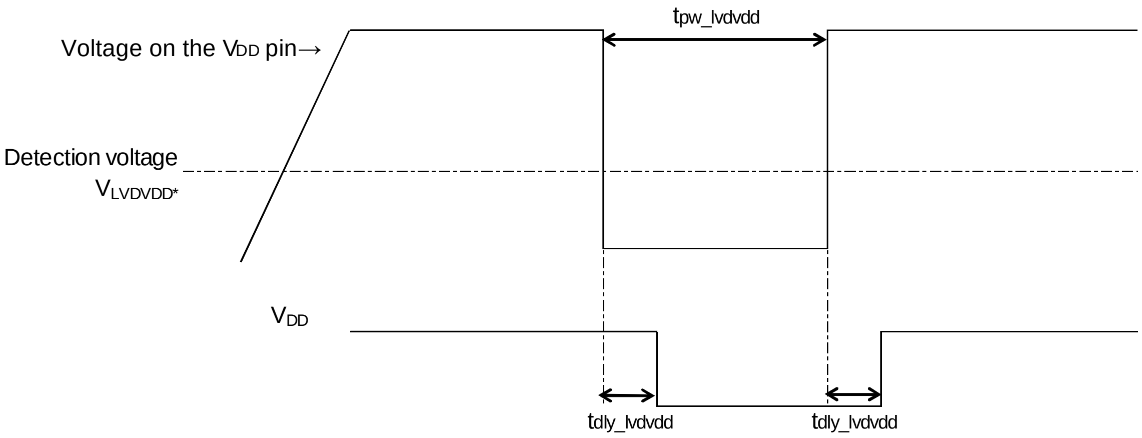
• Hold AV_{DD} ≥ 0.8 V until V_{DD} ≥ 1.47 V.

• Hold the RESET pin low until V_{DD} ≥ 1.47 V and AV_{DD} ≥ 0.8 V.

43.6.8 V_{DD} pin voltage detection characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

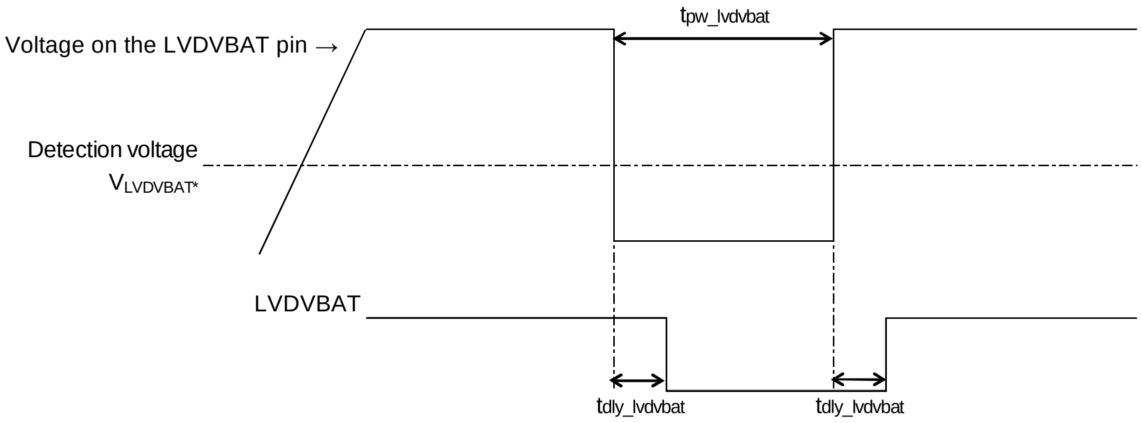
Parameter	Symbol	LVDVDD[2:0]	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{LVDVDD0}$	000	Rising	2.40	2.53	2.65	V
			Falling	2.33	2.46	2.58	V
	$V_{LVDVDD1}$	001	Rising	2.60	2.74	2.86	V
			Falling	2.53	2.67	2.79	V
	$V_{LVDVDD2}$	010	Rising	2.79	2.94	3.07	V
			Falling	2.73	2.87	2.99	V
	$V_{LVDVDD3}$	011	Rising	3.00	3.15	3.28	V
			Falling	2.93	3.08	3.21	V
	$V_{LVDVDD4}$	100	Rising	3.30	3.46	3.60	V
			Falling	3.23	3.39	3.52	V
	$V_{LVDVDD5}$	101	Rising	3.59	3.77	3.91	V
			Falling	3.53	3.70	3.84	V
Minimum pulse width	t_{pw_lvdvdd}	–	–	300			µs
Detection delay time	t_{dly_lvdvdd}	–	–			300	µs



43.6.9 LVDVBAT pin voltage detection characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

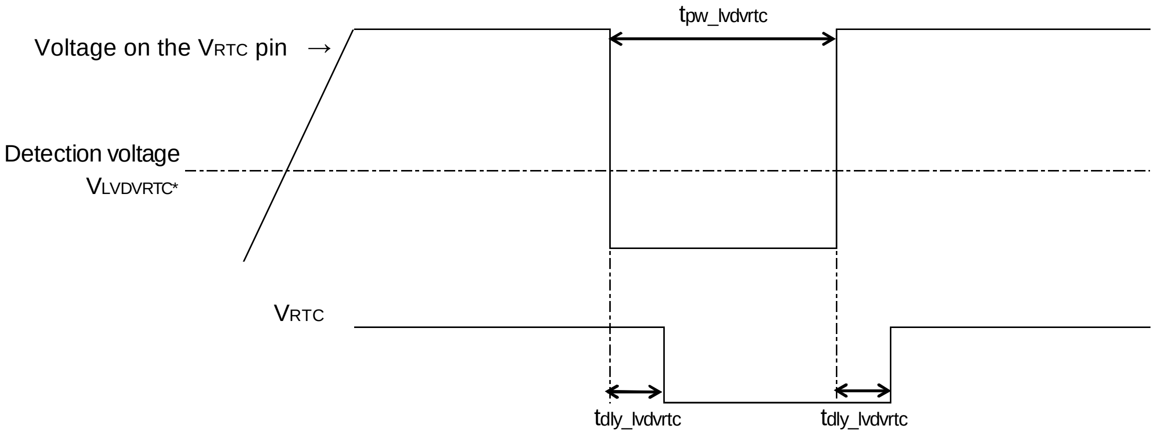
Parameter	Symbol	LVDVBAT[2:0]	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{LVDVBAT0}$	000	Rising	2.11	2.23	2.34	V
			Falling	2.06	2.17	2.28	V
	$V_{LVDVBAT1}$	001	Rising	2.31	2.43	2.54	V
			Falling	2.25	2.37	2.48	V
	$V_{LVDVBAT2}$	010	Rising	2.51	2.63	2.74	V
			Falling	2.45	2.57	2.68	V
	$V_{LVDVBAT3}$	011	Rising	2.60	2.73	2.84	V
			Falling	2.54	2.67	2.78	V
	$V_{LVDVBAT4}$	100	Rising	2.69	2.83	2.95	V
			Falling	2.64	2.77	2.89	V
	$V_{LVDVBAT5}$	101	Rising	2.79	2.93	3.05	V
			Falling	2.73	2.87	2.99	V
	$V_{LVDVBAT6}$	110	Rising	2.99	3.13	3.26	V
			Falling	2.93	3.07	3.19	V
Minimum pulse width	$t_{pw\_lvdvbat}$	–	–	300			µs
Detection delay time	$t_{dly\_lvdvbat}$	–	–			300	µs
Pin resistor	$r_{in\_lvdvbat}$	–	LVDVBATEN = 1		109		MΩ



43.6.10 V_{RTC} pin voltage detection characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

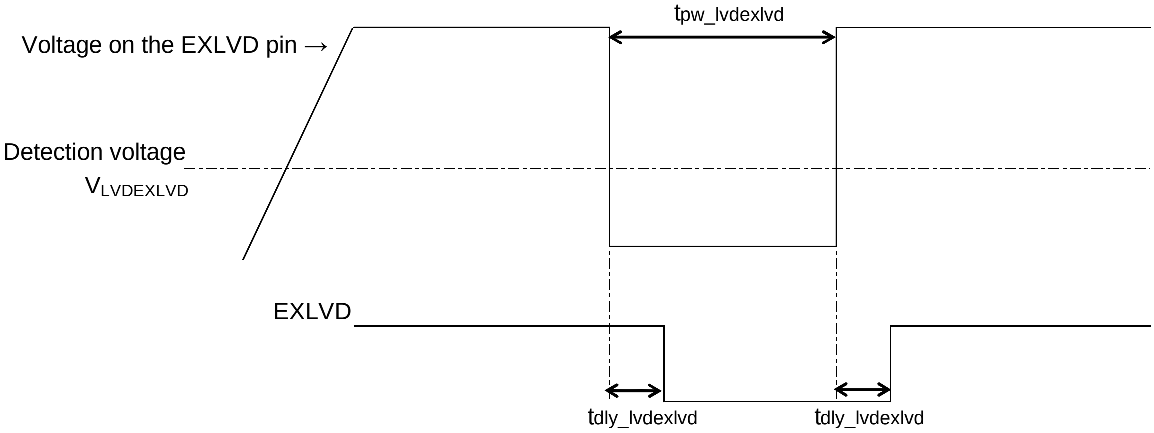
Parameter	Symbol	LVDVRTC[1:0]	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{LVDVRTC0}$	00	Rising	2.16	2.22	2.28	V
			Falling	2.10	2.16	2.22	V
	$V_{LVDVRTC1}$	01	Rising	2.36	2.43	2.50	V
			Falling	2.30	2.37	2.44	V
	$V_{LVDVRTC2}$	10	Rising	2.56	2.63	2.70	V
			Falling	2.50	2.57	2.64	V
	$V_{LVDVRTC3}$	11	Rising	2.76	2.84	2.92	V
			Falling	2.70	2.78	2.86	V
Minimum pulse width	$t_{pw\_lvdvrtc}$	–	–	300			µs
Detection delay time	$t_{dly\_lvdvrtc}$	–	–			300	µs



43.6.11 EXLVD pin voltage detection

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{LVDEXLVD}$	Rising	1.25	1.33	1.41	V
		Falling	1.20	1.28	1.36	V
Minimum pulse width	$t_{pw\_lvdexlvd}$	–	300			µs
Detection delay time	$t_{dly\_lvdexlvd}$	–			300	µs
Pin resistor	r_{in_exlvd}	LVDEXLVDEN = 1		34		MΩ



43.7 LCD Characteristics

43.7.1 Resistance division method

(1) Static display mode

(T_A = –40 to +85°C, V_{L4} (MIN.) ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	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 +85°C, V_{L4} (MIN.) ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.7		V_{DD}	V

(3) 1/3 bias method

(T_A = –40 to +85°C, V_{L4} (MIN.) ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.5		V_{DD}	V

43.7.2 Internal voltage boosting method

(1) 1/3 bias method

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C4^{Note 1} = 0.47 µF	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
			VLCD = 0BH	1.25	1.35	1.43	V
			VLCD = 0CH	1.30	1.40	1.48	V
			VLCD = 0DH	1.35	1.45	1.53	V
			VLCD = 0EH	1.40	1.50	1.58	V
			VLCD = 0FH	1.45	1.55	1.63	V
			VLCD = 10H	1.50	1.60	1.68	V
			VLCD = 11H	1.55	1.65	1.73	V
			VLCD = 12H	1.60	1.70	1.78	V
			VLCD = 13H Note 4	1.65	1.75	1.83	V
Doubler output voltage	V_{L2}	C1 to C4^{Note 1} = 0.47 µF		2 V_{L1}–0.10	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L4}	C1 to C4^{Note 1} = 0.47 µF		3 V_{L1}–0.15	3 V_{L1}	3 V_{L1}	V
Reference voltage setup time^{Note 2}	t_{VWAIT1}			5			ms
Voltage boost wait time^{Note 3}	t_{VWAIT2}	C1 to C4^{Note 1} = 0.47 µF		500			ms

Notes 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 µF±30%

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 (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

4. This setting is only available when V_{DD} ≥ V_{L1}.

(2) 1/4 bias method

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1}	C1 to C5[Note 1] = 0.47 µF	VLCD = 04H	0.90	1.00	1.08	V
			VLCD = 05H	0.95	1.05	1.13	V
			VLCD = 06H	1.00	1.10	1.18	V
			VLCD = 07H	1.05	1.15	1.23	V
			VLCD = 08H	1.10	1.20	1.28	V
			VLCD = 09H	1.15	1.25	1.33	V
			VLCD = 0AH	1.20	1.30	1.38	V
Doubler output voltage	V_{L2}	C1 to C5[Note 1] = 0.47 µF		2 V_{L1}–0.08	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C5[Note 1] = 0.47 µF		3 V_{L1}–0.12	3 V_{L1}	3 V_{L1}	V
Quadruply output voltage	V_{L4}	C1 to C5[Note 1] = 0.47 µF		4 V_{L1}–0.16	4 V_{L1}	4 V_{L1}	V
Reference voltage setup time[Note 2]	t_{VWAIT1}			5			ms
Voltage boost wait time[Note 3]	t_{VWAIT2}	C1 to C5[Note 1] = 0.47 µF		500			ms

Notes 1. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L3} and GND
C5: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = C5 = 0.47 µF±30%

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 (by setting the MDSET1 and MDSET0 bits of the LCDM0 register to 01B) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

43.7.3 Capacitor split method

(1) 1/3 bias method

(T_A = –40 to +85°C, 2.2 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{L4} voltage	V_{L4}	C1 to C4 = 0.47 μF[Note 2]		V_{DD}		V
V_{L2} voltage	V_{L2}	C1 to C4 = 0.47 μF[Note 2]	2/3 V_{L4}– 0.1	2/3 V_{L4}	2/3 V_{L4} + 0.1	V
V_{L1} voltage	V_{L1}	C1 to C4 = 0.47 μF[Note 2]	1/3 V_{L4}– 0.1	1/3 V_{L4}	1/3 V_{L4} + 0.1	V
Capacitor split wait time[Note 1]	t_{VWAIT}		100			ms

Notes 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

2. This is a capacitor that is connected between voltage pins used to drive the LCD.

C1: A capacitor connected between CAPH and CAPL

C2: A capacitor connected between V_{L1} and GND

C3: A capacitor connected between V_{L2} and GND

C4: A capacitor connected between V_{L4} and GND

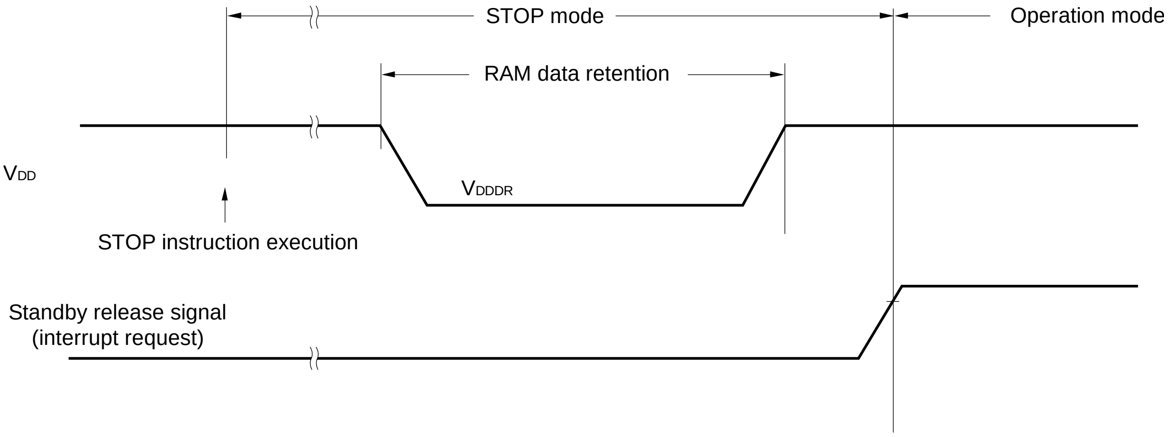
C1 = C2 = C3 = C4 = 0.47 μF±30%

43.8 RAM Data Retention Characteristics

(T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46^Note		5.5	V

Note The value 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.



43.9 Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V[Note 4], AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	1.8 V ≤ V_{DD} ≤ 5.5 V[Note 4]	1		32	MHz
Number of code flash rewrites[Notes 1, 2, 3]	C_{erwr}	Retained for 20 years T_A = 85°C	1,000			Times
Number of data flash rewrites[Notes 1, 2, 3]		Retained for 1 year T_A = 25°C		1,000,000		
		Retained for 5 years T_A = 85°C	100,000			
		Retained for 20 years T_A = 85°C	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. This characteristic indicates the flash memory characteristic and based on Renesas Electronics reliability test.

4. Execute bank programming while the following conditions are satisfied.
(1) 2.7 V ≤ V_{DD} ≤ 5.5 V
(2) Operation is in HS (high-speed main) mode or LS (low-speed main) mode.
Bank programming is prohibited in LP (low-power main) mode and LV (low-voltage main) mode.
Self-programming is possible when 1.8 V ≤ V_{DD} ≤ 5.5 V.

43.10 Dedicated Flash Memory Programmer Communication (UART)

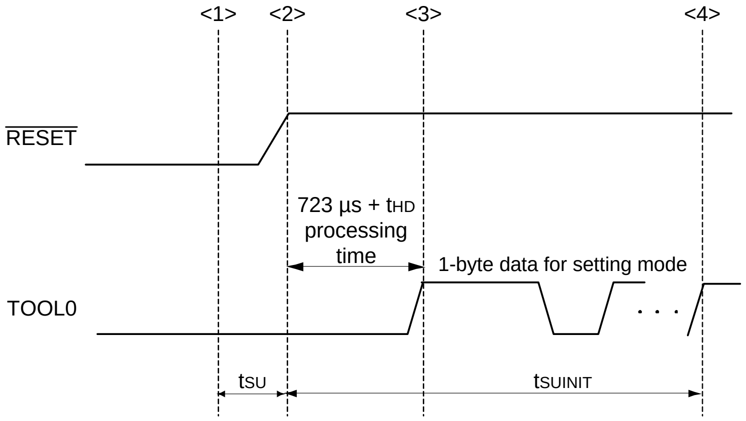
(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

43.11 Timing of Entry to Flash Memory Programming Modes

(T_A = –40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} = EV_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	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	t_{SU}	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 (excluding the processing time of the firmware to control the flash memory)	t_{HD}	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 (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 segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level.

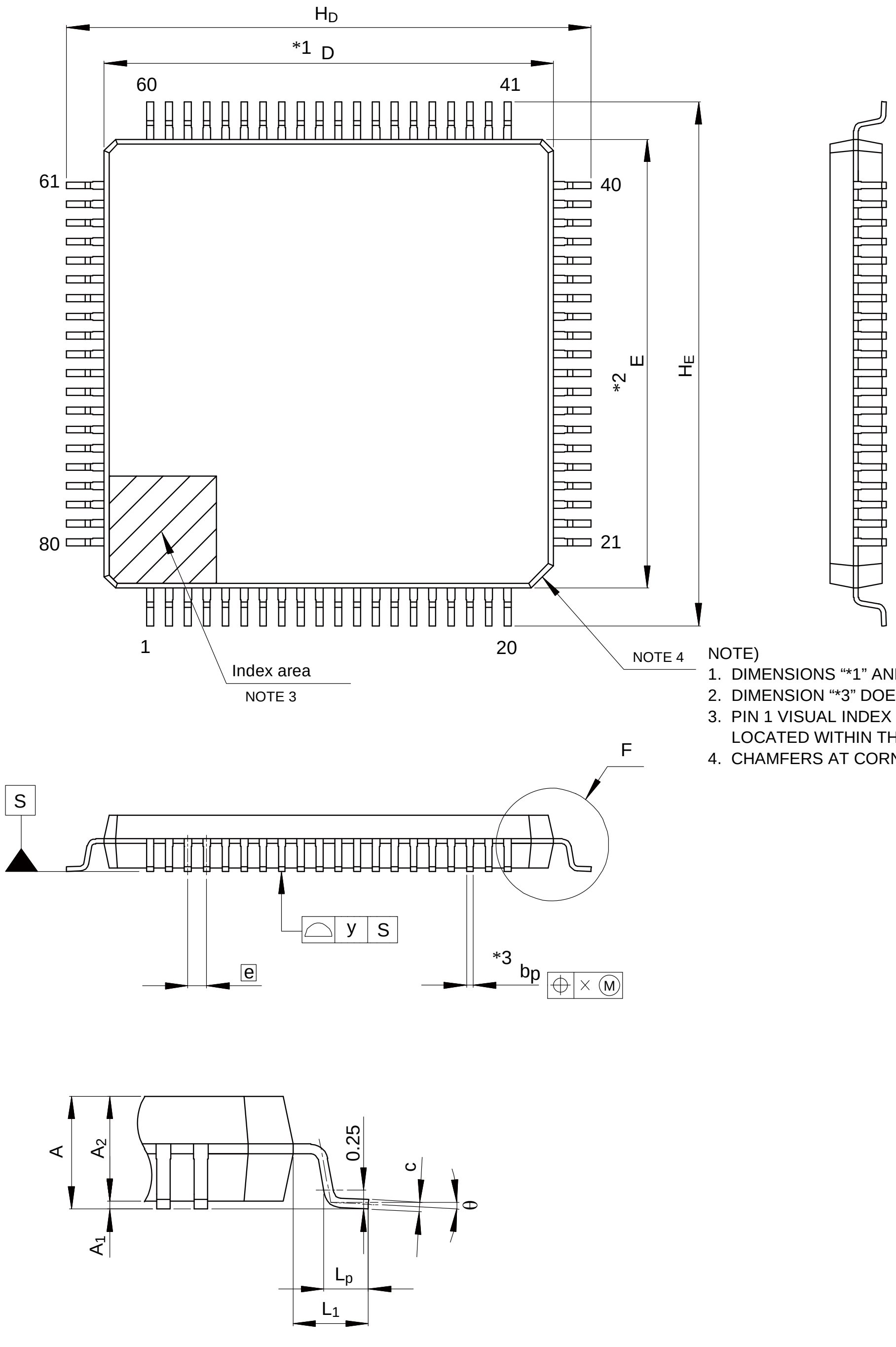
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

CHAPTER 44 PACKAGE DRAWINGS

44.1 80-pin Product

R5F10NMLDFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5



Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

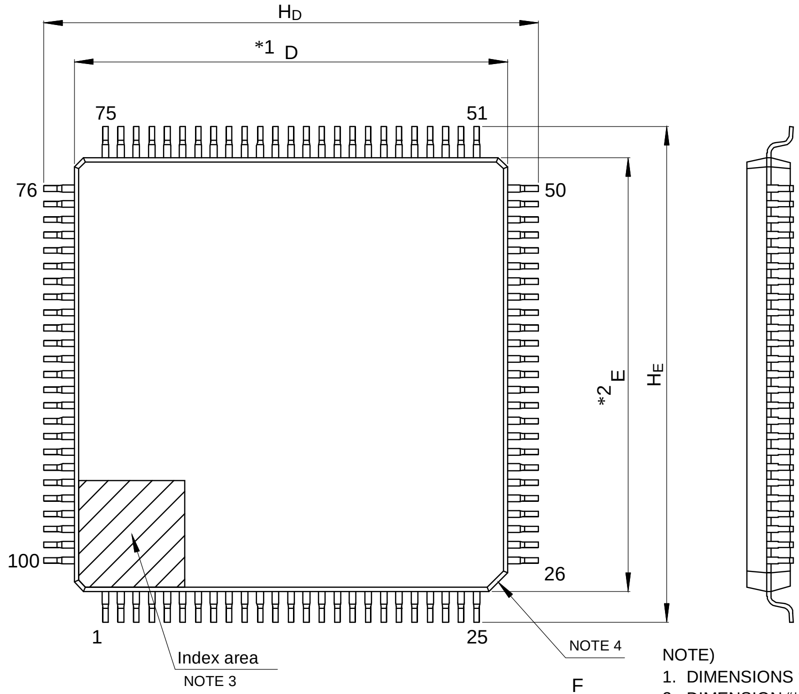
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

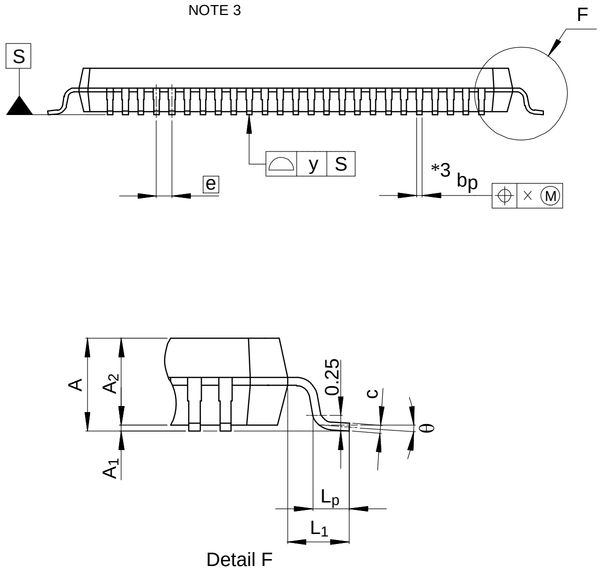
44.2 100-pin Product

R5F10NPLDFB

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

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	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	0.15	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	—

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.4	Modification of Figure 1-1. Part Number, Memory Size, and Package of RL78/I1C (512 KB)	(d)
p.4	Modification of Table 1-1. List of Ordering Part Numbers	(d)
p.5	Addition of Caution 3 in 1.3.1 80-pin product	(c)
p.6	Addition of Caution 5 in 1.3.2 100-pin product	(c)
CHAPTER 2 PIN FUNCTIONS		
p.27	Modification of Table 2-3. Connection of Unused Pins (2/2)	(c)
p.44	Addition of Caution1, 2 and Remark of Figure 2-17. Pin Block Diagram for Pin Type 12-1-6	(c)
CHAPTER 4 PORT FUNCTIONS		
p.107	Modification of description and deletion of Note in 4.2.13 Port 15	(c)
p.143	Modification of Table 4-8. Setting Examples of Registers and Output Latches When Using Alternate Function (12/12)	(c)
p.148	Addition of 4.6.3 Point to Note on the P150 to P152	(c)
CHAPTER 34 SAFETY FUNCTIONS		
p.1107	Modification of description in 34.1 Overview of Safety Functions	(c)
p.1112	Modification of description in 34.3.2 CRC operation function (general-purpose CRC)	(c)
p.1116	Modification of description in 34.3.4 RAM guard function	(c)
p.1117	Modification of description in 34.3.5 SFR guard function	(c)
CHAPTER 43 ELECTRICAL SPECIFICATIONS		
p.1225	Addition of description and note in the table of 43.3.1 Pin characteristics	(b)

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. Chapter indicates the chapter of each edition.

Edition	Description	Chapter
Rev.1.00	First edition issued	**All**
Rev.1.01	The module name for 3-wire serial I/O and 3-wire serial were changed to simplified SPI.	**All**
	The module name for CSI was changed to simplified SPI.	
	“Wait” was modified to “clock stretch”.	
	Addition of Note 1.1 Features	**CHAPTER 1 OUTLINE**
	Addition of Note in 4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers	**CHAPTER 4 PORT FUNCTIONS**
	Addition of Note in CHAPTER 20 SERIAL ARRAY UNIT	**CHAPTER 20 SERIAL ARRAY UNIT**
Rev.1.10	Modification of Notes 1 and 4 in 43.3.2 Supply current characteristics	**CHAPTER 43 ELECTRICAL SPECIFICATIONS**
	Modification of Note 9 to Note 5 in 43.3.2 Supply current characteristics	
	Modification of Note 5 to Note 6 in 43.3.2 Supply current characteristics	
	Deletion of Note 6 in 43.3.2 Supply current characteristics	
	Modification of Notes 1 and 5 and delete Notes 6 in 43.3.2 Supply current characteristics	
	Addition of Note 5 in 43.3.2 Supply current characteristics	
	Modification of Note 5 to Note 6 in 43.3.2 Supply current characteristics	

RL78/I1C (512 KB) User’s Manual: Hardware

Publication Date: Rev. 1.00 Dec 25, 2020
Rev. 1.11 Mar 22, 2024

Published by: Renesas Electronics Corporation

RL78/I1C (512 KB)

Renesas Electronics Corporation