Preliminary User's Manual

78K0/KD1+

8-Bit Single-Chip Microcontrollers

μPD78F0122H
μPD78F0123H
μPD78F0124H
μPD78F0124HD

Document No. U16962EJ1V0UD00 (1st edition)
Date Published February 2004 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① 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, etc., 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).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

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

- **The information contained in this document is being issued in advance of the production cycle for the product. The parameters for the product may change before final production or NEC Electronics Corporation, at its own discretion, may withdraw the product prior to its production.**
- **Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics products depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M5D 02.11-1

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

[GLOBAL SUPPORT]
http://www.necel.com/en/support/support.html

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87

- **Succursale Française**
  Vélizy-Villacoublay, France
  Tel: 01-30-67 58 00

- **Filiale Italiana**
  Milano, Italy
  Tel: 02-66 75 41

- **Branch The Netherlands**
  Eindhoven, The Netherlands
  Tel: 040-244 58 45

- **Tyskland Filial**
  Taeby, Sweden
  Tel: 08-63 80 820

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

J04.1

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0/KD1+ and design and develop application systems and programs for these devices.
The target products are as follows.

78K0/KD1+: μPD78F0122H, 78F0123H, 78F0124H, 78F0124HD

Purpose

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

Organization

The 78K0/KD1+ manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

78K0/KD1+ **User's Manual** **(This Manual)**	**78K/0 Series** **User's Manual** **Instructions**
• Pin functions	• CPU functions
• Internal block functions	• Instruction set
• Interrupts	• Explanation of each instruction
• Other on-chip peripheral functions	
• Electrical specifications (target)	

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 brackets, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name:
  → Refer to **APPENDIX C REGISTER INDEX**.
- To know details of the 78K/0 Series instructions:
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

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

Differences Between 78K0/KD1+ and 78K0/KD1

Series Name Item		78K0/KD1+	78K0/KD1
Mask ROM version		None	Available
Flash memory version	Power supply	Single power supply	Two power supplies
	Self-programming function	Available	None
	Option byte	Ring-OSC can be stopped/cannot be stopped selectable	None
Power-on clear (POC) function		2.1 V ±0.1 V (fixed)[Note]	2.85 V ±0.15 V or 3.5 V ±0.2 V selectable
Regulator		None	Available
Version with on-chip debug function		Available (µPD78F0124HD)	None
Minimum instruction execution time		0.125 µs (at 16 MHz operation)	0.2 µs (at 10 MHz operation)

Note This value may change after evaluation.

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.
78K0/KD1+ User's Manual	This manual
78K0/KD1 User's Manual	U16315E
78K/0 Series Instructions User's Manual	U12326E

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
RA78K0 Assembler Package	Operation	U14445E
	Language	U14446E
	Structured Assembly Language	U11789E
CC78K0 C Compiler	Operation	U14297E
	Language	U14298E
SM78K Series System Simulator Ver. 2.30 or Later	Operation (Windows™ Based)	U15373E
	External Part User Open Interface Specifications	U15802E
ID78K Series Integrated Debugger Ver. 2.30 or Later	Operation (Windows Based)	U15185E
Project Manager Ver. 3.12 or Later (Windows Based)		U14610E

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

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
IE-78K0-NS In-Circuit Emulator	U13731E
IE-78K0-NS-A In-Circuit Emulator	U14889E
IE-78K0K1-ET In-Circuit Emulator	U16604E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.125 μs: @ 16 MHz operation with high-speed system clock) to ultra low-speed (122 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory
			Internal High-Speed RAM
μPD78F0122H	Flash memory	16 KB[Note]	512 bytes[Note]
μPD78F0123H		24 KB[Note]	1024 bytes[Note]
μPD78F0124H, 78F0124HD		32 KB[Note]	

Note The internal flash memory and internal high-speed RAM capacities can be changed using the internal memory size switching register (IMS).

- On-chip single-power-supply flash memory
- Self-programming (with boot swap function)
- On-chip debug function (μPD78F0124HD only)
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- Short startup is possible via the CPU default start using the on-chip Ring-OSC
- On-chip clock monitor function using on-chip Ring-OSC
- On-chip watchdog timer (operable with Ring-OSC clock)
- On-chip key interrupt function
- On-chip clock output controller
- I/O ports: 39 (N-ch open drain: 4)
- Timer: 7 channels
- Serial interface: 2 channels
  (UART (LIN (Local Interconnect Network)-bus supported): 1 channel, CSI/UART[Note 1]: 1 channel)
- 10-bit resolution A/D converter: 8 channels
- Supply voltage: V_{DD} = 2.7 to 5.5 V (with Ring-OSC clock or subsystem clock: V_{DD} = 2.0 to 5.5 V[Note 2])
- Operating ambient temperature: T_A = –40 to +85°C

Notes 1. Select either of the functions of these alternate-function pins.

2. Use the product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on-clear (POC) circuit is 2.1 V ±0.1 V.

Caution The operating voltage range may be changed after evaluation of the device.

1.2 Applications

- ○ Automotive equipment
  - System control for body electricals (power windows, keyless entry reception, etc.)
  - Sub-microcontrollers for control
- ○ Home audio, car audio
- ○ AV equipment
- ○ PC peripheral equipment (keyboards, etc.)
- ○ Household electrical appliances
  - Outdoor air conditioner units
  - Microwave ovens, electric rice cookers
- ○ Industrial equipment
  - Pumps
  - Vending machines
  - FA (Factory Automation)

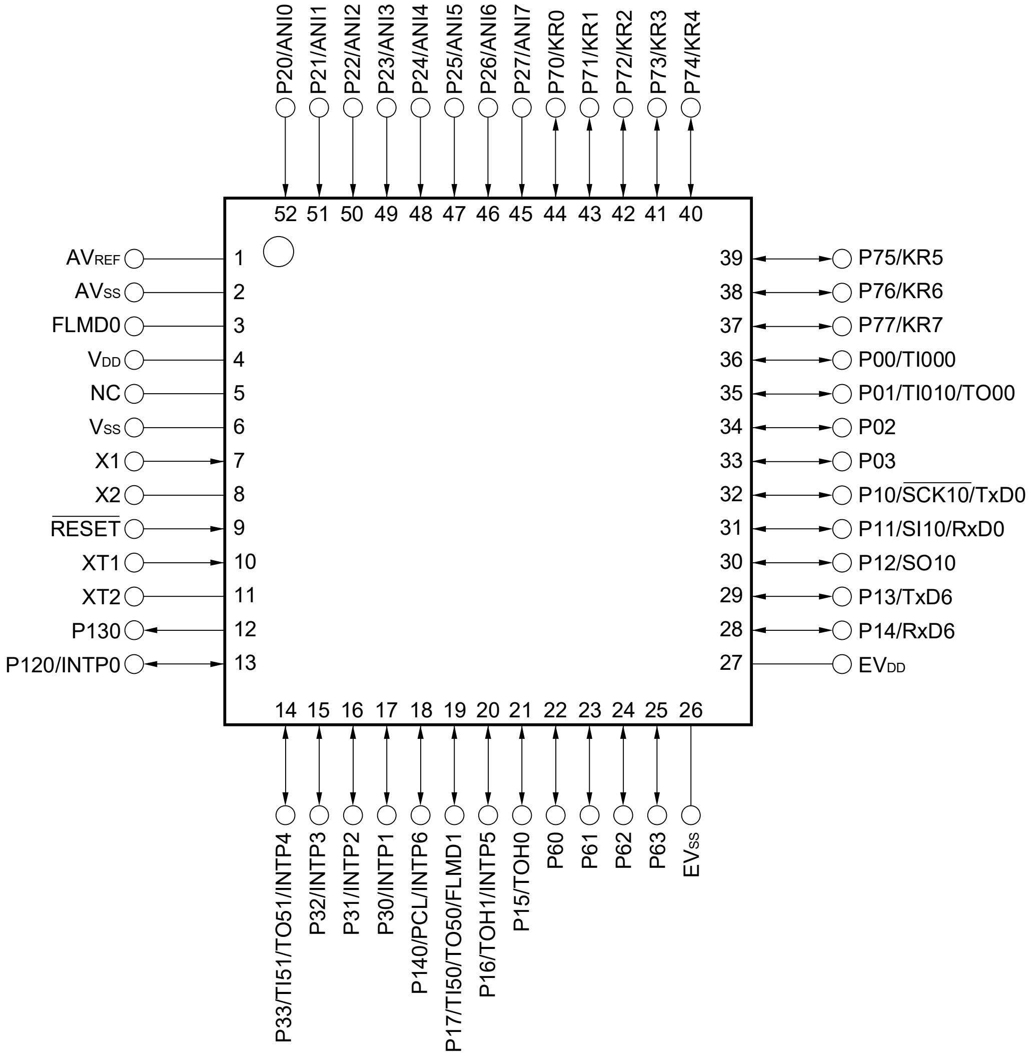
1.3 Ordering Information

- **Flash memory version**

Part Number	Package
μPD78F0122HGB-8ET	52-pin plastic LQFP (10 × 10)
μPD78F0123HGB-8ET	52-pin plastic LQFP (10 × 10)
μPD78F0124HGB-8ET	52-pin plastic LQFP (10 × 10)
μPD78F0124HDGB-8ET	52-pin plastic LQFP (10 × 10)

1.4 Pin Configuration (Top View)

- 52-pin plastic LQFP (10 × 10)



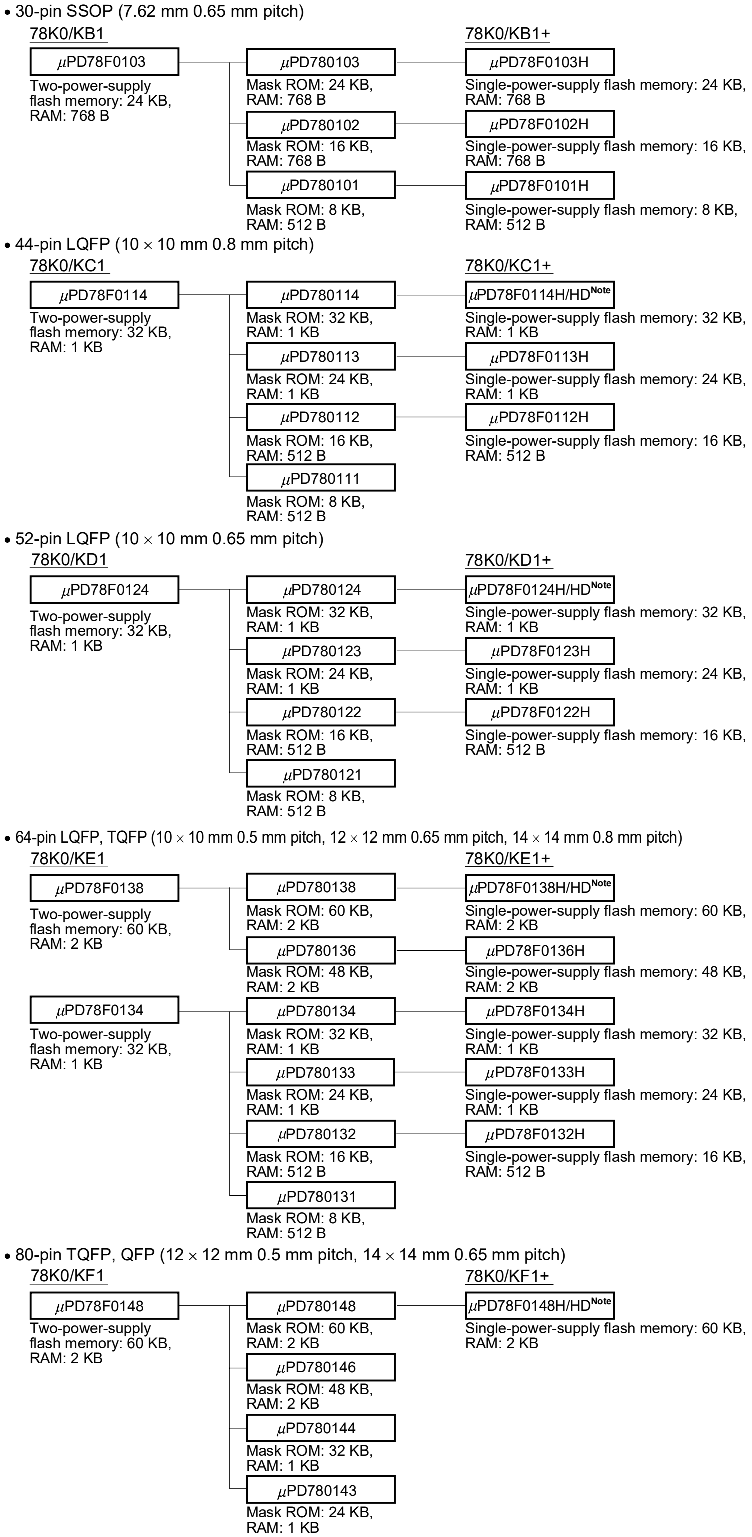
Caution Connect the AV$_{SS}$ pin to V$_{SS}$.

Pin Identification

ANI0 to ANI7:	Analog input	P130:	Port 13
AV$_{REF}$:	Analog reference voltage	P140:	Port 14
AV$_{SS}$:	Analog ground	PCL:	Programmable clock output
EV$_{DD}$:	Power supply for port	$\overline{\text{RESET}}$:	Reset
EV$_{SS}$:	Ground for port	RxD0, RxD6:	Receive data
FLMD0, FLMD1:	Flash programming mode	$\overline{\text{SCK10}}$:	Serial clock input/output
INTP0 to INTP6:	External interrupt input	SI10:	Serial data input
KR0 to KR7:	Key return	SO10:	Serial data output
NC:	Non-connection	TI000, TI010,TI50, TI51:	Timer input
P00 to P03:	Port 0	TO00, TO50, TO51,	
P10 to P17:	Port 1	TOH0, TOH1:	Timer output
P20 to P27:	Port 2	TxD0, TxD6:	Transmit data
P30 to P33:	Port 3	V$_{DD}$:	Power supply
P60 to P63:	Port 6	V$_{SS}$:	Ground
P70 to P77:	Port 7	X1, X2:	Crystal oscillator (high-speed system clock)
P120:	Port 12	XT1, XT2:	Crystal oscillator (subsystem clock)

1.5 K1 Family Lineup

1.5.1 78K0/Kx1, 78K0/Kx1+ product lineup



- 30-pin SSOP (7.62 mm 0.65 mm pitch)

78K0/KB1		78K0/KB1+
μPD78F0103 Two-power-supply flash memory: 24 KB, RAM: 768 B	μPD780103 Mask ROM: 24 KB, RAM: 768 B	μPD78F0103H Single-power-supply flash memory: 24 KB, RAM: 768 B
	μPD780102 Mask ROM: 16 KB, RAM: 768 B	μPD78F0102H Single-power-supply flash memory: 16 KB, RAM: 768 B
	μPD780101 Mask ROM: 8 KB, RAM: 512 B	μPD78F0101H Single-power-supply flash memory: 8 KB, RAM: 512 B

- 44-pin LQFP (10 × 10 mm 0.8 mm pitch)

78K0/KC1		78K0/KC1+
μPD78F0114 Two-power-supply flash memory: 32 KB, RAM: 1 KB	μPD780114 Mask ROM: 32 KB, RAM: 1 KB	μPD78F0114H/HD[Note] Single-power-supply flash memory: 32 KB, RAM: 1 KB
	μPD780113 Mask ROM: 24 KB, RAM: 1 KB	μPD78F0113H Single-power-supply flash memory: 24 KB, RAM: 1 KB
	μPD780112 Mask ROM: 16 KB, RAM: 512 B	μPD78F0112H Single-power-supply flash memory: 16 KB, RAM: 512 B
	μPD780111 Mask ROM: 8 KB, RAM: 512 B	

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

78K0/KD1		78K0/KD1+
μPD78F0124 Two-power-supply flash memory: 32 KB, RAM: 1 KB	μPD780124 Mask ROM: 32 KB, RAM: 1 KB	μPD78F0124H/HD[Note] Single-power-supply flash memory: 32 KB, RAM: 1 KB
	μPD780123 Mask ROM: 24 KB, RAM: 1 KB	μPD78F0123H Single-power-supply flash memory: 24 KB, RAM: 1 KB
	μPD780122 Mask ROM: 16 KB, RAM: 512 B	μPD78F0122H Single-power-supply flash memory: 16 KB, RAM: 512 B
	μPD780121 Mask ROM: 8 KB, RAM: 512 B	

- 64-pin LQFP, TQFP (10 × 10 mm 0.5 mm pitch, 12 × 12 mm 0.65 mm pitch, 14 × 14 mm 0.8 mm pitch)

78K0/KE1		78K0/KE1+
μPD78F0138 Two-power-supply flash memory: 60 KB, RAM: 2 KB	μPD780138 Mask ROM: 60 KB, RAM: 2 KB	μPD78F0138H/HD[Note] Single-power-supply flash memory: 60 KB, RAM: 2 KB
	μPD780136 Mask ROM: 48 KB, RAM: 2 KB	μPD78F0136H Single-power-supply flash memory: 48 KB, RAM: 2 KB
μPD78F0134 Two-power-supply flash memory: 32 KB, RAM: 1 KB	μPD780134 Mask ROM: 32 KB, RAM: 1 KB	μPD78F0134H Single-power-supply flash memory: 32 KB, RAM: 1 KB
	μPD780133 Mask ROM: 24 KB, RAM: 1 KB	μPD78F0133H Single-power-supply flash memory: 24 KB, RAM: 1 KB
	μPD780132 Mask ROM: 16 KB, RAM: 512 B	μPD78F0132H Single-power-supply flash memory: 16 KB, RAM: 512 B
	μPD780131 Mask ROM: 8 KB, RAM: 512 B	

- 80-pin TQFP, QFP (12 × 12 mm 0.5 mm pitch, 14 × 14 mm 0.65 mm pitch)

78K0/KF1		78K0/KF1+
μPD78F0148 Two-power-supply flash memory: 60 KB, RAM: 2 KB	μPD780148 Mask ROM: 60 KB, RAM: 2 KB	μPD78F0148H/HD[Note] Single-power-supply flash memory: 60 KB, RAM: 2 KB
	μPD780146 Mask ROM: 48 KB, RAM: 2 KB	
	μPD780144 Mask ROM: 32 KB, RAM: 1 KB	
	μPD780143 Mask ROM: 24 KB, RAM: 1 KB	

Note Product with on-chip debug function

The list of functions in the 78K0/Kx1 is shown below.

<table>
<tr><th colspan="2">Part Number
Item</th><th colspan="3">78K0/KB1</th><th colspan="3">78K0/KC1</th><th colspan="3">78K0/KD1</th><th colspan="5">78K0/KE1</th><th colspan="3">78K0/KF1</th></tr>
<tr><td colspan="2">Number of pins</td><td colspan="3">30 pins</td><td colspan="3">44 pins</td><td colspan="3">52 pins</td><td colspan="5">64 pins</td><td colspan="3">80 pins</td></tr>
<tr><td rowspan="3">Internal memory (bytes)</td><td>Mask ROM</td><td>8 K</td><td>16 K/
24 K</td><td>–</td><td>8 K/
16 K</td><td>24 K/
32 K</td><td>–</td><td>8 K/
16 K</td><td>24 K/
32 K</td><td>–</td><td>8 K/
16 K</td><td>24 K/
32 K</td><td>–</td><td>48 K/
60 K</td><td>–</td><td>24 K/
32 K</td><td>48 K/
60 K</td><td>–</td></tr>
<tr><td>Flash memory</td><td colspan="2">–</td><td>24 K</td><td colspan="2">–</td><td>32 K</td><td colspan="2">–</td><td>32 K</td><td colspan="2">–</td><td>32 K</td><td>–</td><td>60 K</td><td colspan="2">–</td><td>60 K</td></tr>
<tr><td>RAM</td><td>512</td><td colspan="2">768</td><td>512</td><td colspan="2">1 K</td><td>512</td><td colspan="2">1 K</td><td>512</td><td colspan="2">1 K</td><td colspan="2">2 K</td><td>1 K</td><td colspan="2">2 K</td></tr>
<tr><td colspan="2">Power supply voltage</td><td colspan="17">V_{DD} = 2.7 to 5.5 V</td></tr>
<tr><td colspan="2">Minimum instruction execution time</td><td colspan="6">0.2 μs (when 10 MHz, V_{DD} = 4.0 to 5.5 V)
0.24 μs (when 8.38 MHz, V_{DD} = 3.3 to 5.5 V)
0.4 μs (when 5 MHz, V_{DD} = 2.7 to 5.5 V)</td><td colspan="11">&lt;Connect REGC pin to V_{DD}>
0.2 μs (when 10 MHz, V_{DD} = 4.0 to 5.5 V)
0.24 μs (when 8.38 MHz, V_{DD} = 3.3 to 5.5 V)
0.4 μs (when 5 MHz, V_{DD} = 2.7 to 5.5 V)</td></tr>
<tr><td rowspan="3">Clock</td><td>X1 input</td><td colspan="17">2 to 10 MHz</td></tr>
<tr><td>Subclock</td><td colspan="3">–</td><td colspan="14">32.768 kHz</td></tr>
<tr><td>Ring-OSC</td><td colspan="17">240 kHz (TYP.)</td></tr>
<tr><td rowspan="4">Port</td><td>CMOS I/O</td><td colspan="3">17</td><td colspan="3">19</td><td colspan="3">26</td><td colspan="5">38</td><td colspan="3">54</td></tr>
<tr><td>CMOS input</td><td colspan="3">4</td><td colspan="14">8</td></tr>
<tr><td>CMOS output</td><td colspan="17">1</td></tr>
<tr><td>N-ch open-drain I/O</td><td colspan="3">–</td><td colspan="14">4</td></tr>
<tr><td rowspan="5">Timer</td><td>16 bits (TM0)</td><td colspan="10">1 ch</td><td colspan="4">2 ch</td><td>1 ch</td><td colspan="2">2 ch</td></tr>
<tr><td>8 bits (TM5)</td><td colspan="3">1 ch</td><td colspan="14">2 ch</td></tr>
<tr><td>8 bits (TMH)</td><td colspan="17">2 ch</td></tr>
<tr><td>For watch</td><td colspan="3">–</td><td colspan="14">1 ch</td></tr>
<tr><td>WDT</td><td colspan="17">1 ch</td></tr>
<tr><td rowspan="4">Serial interface</td><td>3-wire CSI[Note]</td><td colspan="10">1 ch</td><td colspan="4">2 ch</td><td>1 ch</td><td colspan="2">2 ch</td></tr>
<tr><td>Automatic transmit/ receive 3-wire CSI</td><td colspan="14">–</td><td colspan="3">1 ch</td></tr>
<tr><td>UART[Note]</td><td>–</td><td colspan="16">1 ch</td></tr>
<tr><td>UART supporting LIN-bus</td><td colspan="17">1 ch</td></tr>
<tr><td colspan="2">10-bit A/D converter</td><td colspan="3">4 ch</td><td colspan="14">8 ch</td></tr>
<tr><td rowspan="2">Interrupt</td><td>External</td><td colspan="3">6</td><td colspan="3">7</td><td colspan="3">8</td><td colspan="5">9</td><td colspan="3">9</td></tr>
<tr><td>Internal</td><td>11</td><td colspan="2">12</td><td colspan="6">15</td><td>16</td><td colspan="4">19</td><td>17</td><td colspan="2">20</td></tr>
<tr><td colspan="2">Key return input</td><td colspan="3">–</td><td colspan="3">4 ch</td><td colspan="11">8 ch</td></tr>
<tr><td rowspan="5">Reset</td><td>$\overline{\text{RESET}}$ pin</td><td colspan="17">Provided</td></tr>
<tr><td>POC</td><td colspan="17">2.85 V ±0.15 V/3.5 V ±0.20 V (selectable by option byte)</td></tr>
<tr><td>LVI</td><td colspan="17">3.1 V/3.3 V ±0.15 V/3.5 V/3.7 V/3.9 V/4.1 V/4.3 V ±0.2 V (selectable by software)</td></tr>
<tr><td>Clock monitor</td><td colspan="17">Provided</td></tr>
<tr><td>WDT</td><td colspan="17">Provided</td></tr>
<tr><td colspan="2">Clock output/buzzer output</td><td colspan="6">–</td><td colspan="3">Clock output only</td><td colspan="8">Provided</td></tr>
<tr><td colspan="2">Multiplier/divider</td><td colspan="9">–</td><td colspan="8">16 bits × 16 bits, 32 bits ÷ 16 bits</td></tr>
<tr><td colspan="2">ROM correction</td><td colspan="12">–</td><td colspan="2">Provided</td><td colspan="3">–</td></tr>
<tr><td colspan="2">Standby function</td><td colspan="17">HALT/STOP mode</td></tr>
<tr><td colspan="2">Operating ambient temperature</td><td colspan="17">Standard products, special (A) products: –40 to +85°C
Special (A1) products: –40 to +110°C (mask ROM version),
–40 to +105°C (flash memory version)
Special (A2) products: –40 to +125°C (mask ROM version)</td></tr>
</table>

Note Select either of the functions of these alternate-function pins.

The list of functions in the 78K0/Kx1+ is shown below.

Item \ Part Number		78K0/KB1+		78K0/KC1+		78K0/KD1+		78K0/KE1+			78K0/KF1+
Number of pins		30 pins		44 pins		52 pins		64 pins			80 pins
Internal memory (bytes)	Flash memory	8 K	16 K/24 K	16 K	24 K/32 K	16 K	24 K/32 K	16 K	24 K/32 K	48 K/60 K	60 K
	RAM	512	768	512	1 K	512	1 K	512	1 K	2 K	2 K
Power supply voltage		V_{DD} = 2.7 to 5.5 V (with Ring-OSC clock or subclock: V_{DD} = 2.0 to 5.5 V[Note 1])									
Minimum instruction execution time		0.125 μs (when 16 MHz, V_{DD} = 4.0 to 5.5 V) 0.24 μs (when 8.38 MHz, V_{DD} = 3.3 to 5.5 V) 0.4 μs (when 5 MHz, V_{DD} = 2.7 to 5.5 V)									
Clock	Crystal/ceramic	2 to 16 MHz									
	RC	3 to 4 MHz				–					
	Subclock	–		32.768 kHz							
	Ring-OSC	240 kHz (TYP.)									
Ports	CMOS I/O	17		19		26		38			54
	CMOS input	4		8							
	CMOS output	1									
	N-ch open-drain I/O	–		4							
Timer	16 bits (TM0)	1 ch							2 ch		
	8 bits (TM5)	2 ch									
	8 bits (TMH)	1 ch		2 ch							
	For watch	–		1 ch							
	WDT	1 ch									
Serial interface	3-wire CSI[Note 2]	1 ch							2 ch		
	Automatic transmit/ receive 3-wire CSI	–									1 ch
	UART[Note 2]	–	1 ch								
	UART supporting LIN-bus	1 ch									
10-bit A/D converter		4 ch		8 ch							
Interrupts	External	6		7		8		9			9
	Internal	11	12	15				16	19		20
Key return input		–		4 ch		8 ch					
Reset	$\overline{\text{RESET}}$ pin	Provided									
	POC	2.1 V ±0.1 V (detection voltage is fixed)									
	LVI	2.35 V/2.6 V/2.85 V/3.1 V/3.3 V ±0.15 V/3.5 V/3.7 V/3.9 V/4.1 V/4.3 V ±0.2 V (selectable by software)									
	Clock monitor	Provided									
	WDT	Provided									
Clock output/buzzer output		–				Clock output only		Provided			
External bus interface		–									Provided
Multiplier/divider		–						16 bits × 16 bits, 32 bits ÷ 16 bits			
ROM correction		–								Provided	–
Self-programming function		Provided									
Product with on-chip debug function		μPD78F0114HD, 78F0124HD, 78F0138HD, 78F0148HD									
Standby function		HALT/STOP mode									
Operating ambient temperature		T_A = –40 to +85°C									

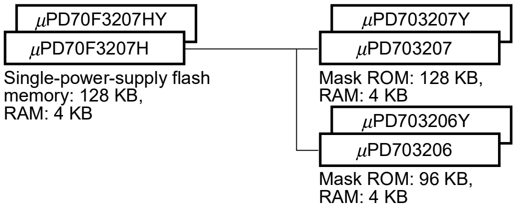
Notes 1. Use the product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on-clear (POC) circuit is 2.1 V ±0.1 V.

2. Select either of the functions of these alternate-function pins.

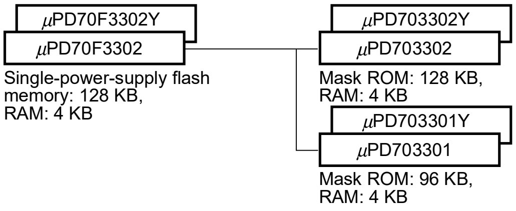
1.5.2 V850ES/Kx1, V850ES/Kx1+ product lineup

- 64-pin plastic LQFP (10 × 10 mm, 0.5 mm pitch)
- 64-pin plastic TQFP (12 × 12 mm, 0.65 mm pitch)
- 64-pin plastic LQFP (14 × 14 mm, 0.8 mm pitch)

V850ES/KE1

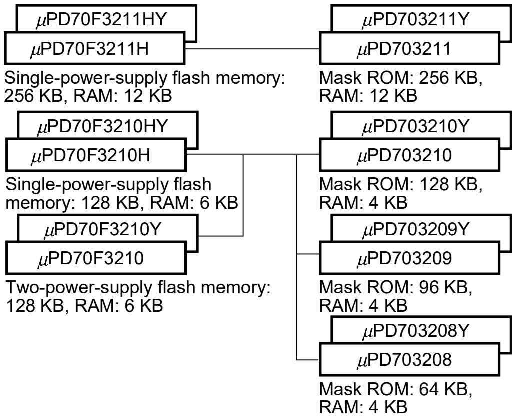

μPD70F3207HY
μPD70F3207H
Single-power-supply flash memory: 128 KB, RAM: 4 KB
μPD703207Y
μPD703207
Mask ROM: 128 KB, RAM: 4 KB
μPD703206Y
μPD703206
Mask ROM: 96 KB, RAM: 4 KB

V850ES/KE1+

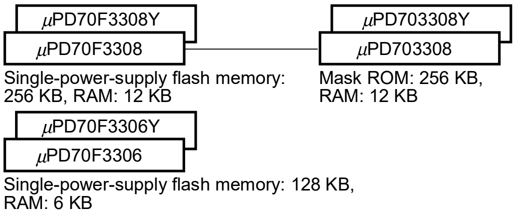

μPD70F3302Y
μPD70F3302
Single-power-supply flash memory: 128 KB, RAM: 4 KB
μPD703302Y
μPD703302
Mask ROM: 128 KB, RAM: 4 KB
μPD703301Y
μPD703301
Mask ROM: 96 KB, RAM: 4 KB

- 80-pin plastic TQFP (12 × 12 mm, 0.5 mm pitch)
- 80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)

V850ES/KF1

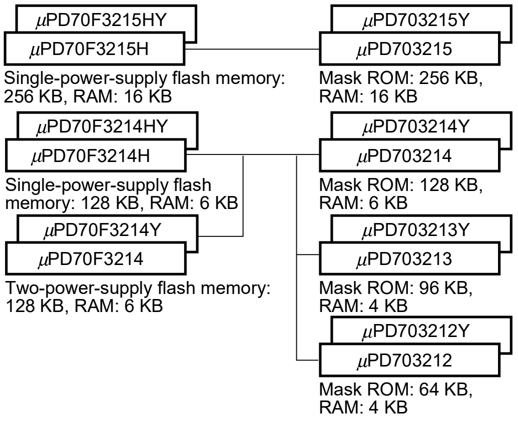

μPD70F3211HY
μPD70F3211H
Single-power-supply flash memory: 256 KB, RAM: 12 KB
μPD703211Y
μPD703211
Mask ROM: 256 KB, RAM: 12 KB
μPD70F3210HY
μPD70F3210H
Single-power-supply flash memory: 128 KB, RAM: 6 KB
μPD70F3210Y
μPD70F3210
Two-power-supply flash memory: 128 KB, RAM: 6 KB
μPD703210Y
μPD703210
Mask ROM: 128 KB, RAM: 4 KB
μPD703209Y
μPD703209
Mask ROM: 96 KB, RAM: 4 KB
μPD703208Y
μPD703208
Mask ROM: 64 KB, RAM: 4 KB

V850ES/KF1+

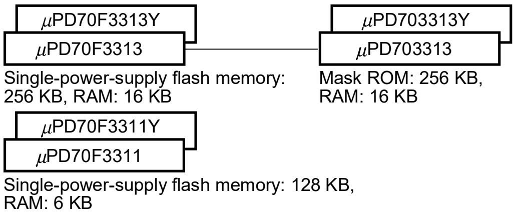

μPD70F3308Y
μPD70F3308
Single-power-supply flash memory: 256 KB, RAM: 12 KB
μPD703308Y
μPD703308
Mask ROM: 256 KB, RAM: 12 KB
μPD70F3306Y
μPD70F3306
Single-power-supply flash memory: 128 KB, RAM: 6 KB

- 100-pin plastic LQFP (14 × 14 mm, 0.5 mm pitch)
- 100-pin plastic QFP (14 × 20 mm, 0.65 mm pitch)

V850ES/KG1

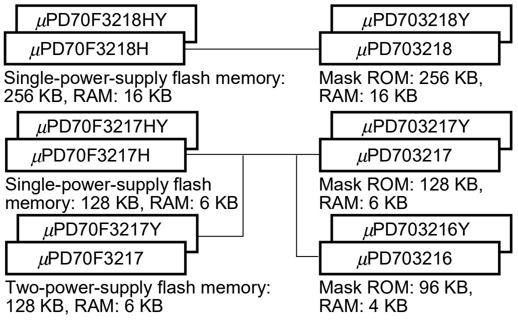

μPD70F3215HY
μPD70F3215H
Single-power-supply flash memory: 256 KB, RAM: 16 KB
μPD703215Y
μPD703215
Mask ROM: 256 KB, RAM: 16 KB
μPD70F3214HY
μPD70F3214H
Single-power-supply flash memory: 128 KB, RAM: 6 KB
μPD70F3214Y
μPD70F3214
Two-power-supply flash memory: 128 KB, RAM: 6 KB
μPD703214Y
μPD703214
Mask ROM: 128 KB, RAM: 6 KB
μPD703213Y
μPD703213
Mask ROM: 96 KB, RAM: 4 KB
μPD703212Y
μPD703212
Mask ROM: 64 KB, RAM: 4 KB

V850ES/KG1+

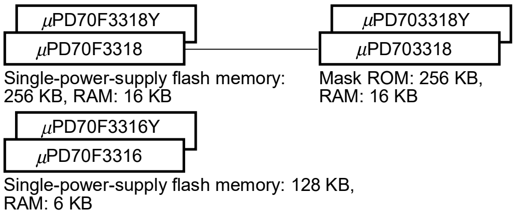

μPD70F3313Y
μPD70F3313
Single-power-supply flash memory: 256 KB, RAM: 16 KB
μPD703313Y
μPD703313
Mask ROM: 256 KB, RAM: 16 KB
μPD70F3311Y
μPD70F3311
Single-power-supply flash memory: 128 KB, RAM: 6 KB

- 144-pin plastic LQFP (20 × 20 mm, 0.5 mm pitch)

V850ES/KJ1


μPD70F3218HY
μPD70F3218H
Single-power-supply flash memory: 256 KB, RAM: 16 KB
μPD703218Y
μPD703218
Mask ROM: 256 KB, RAM: 16 KB
μPD70F3217HY
μPD70F3217H
Single-power-supply flash memory: 128 KB, RAM: 6 KB
μPD70F3217Y
μPD70F3217
Two-power-supply flash memory: 128 KB, RAM: 6 KB
μPD703217Y
μPD703217
Mask ROM: 128 KB, RAM: 6 KB
μPD703216Y
μPD703216
Mask ROM: 96 KB, RAM: 4 KB

V850ES/KJ1+


μPD70F3318Y
μPD70F3318
Single-power-supply flash memory: 256 KB, RAM: 16 KB
μPD703318Y
μPD703318
Mask ROM: 256 KB, RAM: 16 KB
μPD70F3316Y
μPD70F3316
Single-power-supply flash memory: 128 KB, RAM: 6 KB

The list of functions in the V850ES/Kx1 is shown below.

Part Number Item		V850ES/KE1		V850ES/KF1					V850ES/KG1					V850ES/KJ1			
Number of pins		64 pins		80 pins					100 pins					144 pins			
Internal memory (bytes)	Mask ROM	96/128	–	64/96	128	–	256	–	64/96	128	–	256	–	96/128	–	256	–
	Flash memory	–	128	–	–	128	–	256	–	–	128	–	256	–	128	–	256
	RAM	4		4	6		12		4	6		16		6		16	
Power supply voltage		V_{DD} = 2.7 to 5.5 V															
Minimum instruction execution time		50 ns @ 20 MHz															
Clock	X1 input	2 to 10 MHz															
	Subclock	32.768 kHz															
	Ring-OSC	–															
Ports	CMOS input	8		8					8					16			
	CMOS I/O	43		59					76					112			
	N-ch open-drain I/O	1		2					4					6			
Timer	16 bits (TMP)	1 ch		–			1 ch		–			1 ch		–		1 ch	
	16 bits (TM0)	1 ch		2 ch					4 ch					6 ch			
	8 bits (TM5)	2 ch		2 ch					2 ch					2 ch			
	8 bits (TMH)	2 ch		2 ch					2 ch					2 ch			
	Interval timer	1 ch		1 ch					1 ch					1 ch			
	For watch	1 ch		1 ch					1 ch					1 ch			
	WDT1	1 ch		1 ch					1 ch					1 ch			
	WDT2	1 ch		1 ch					1 ch					1 ch			
RTO		6 bits × 1 ch		6 bits × 1 ch					6 bits × 1 ch					6 bits × 2 ch			
Serial interface	CSI	2 ch		2 ch					2 ch					3 ch			
	Automatic transmit/ receive 3-wire CSI	–		1 ch					2 ch					2 ch			
	UART	2 ch		2 ch					2 ch					3 ch			
	UART supporting LIN-bus	–		–					–					–			
	I^{2}C[Note]	1 ch		1 ch					1 ch					2 ch			
External bus	Address space	–		128 KB					3 MB					15 MB			
	Address bus	–		16 bits					22 bits					24 bits			
	Mode	–		Multiplexed mode only					Multiplexed/separate mode								
DMA controller		–		–					–					–			
10-bit A/D converter		8 ch		8 ch					8 ch					16 ch			
8-bit D/A converter		–		–					2 ch					2 ch			
Interrupts	External	8		8					8					8			
	Internal	26		26			29		31			34		40		43	
Key return input		8 ch		8 ch					8 ch					8 ch			
Reset	$\overline{\text{RESET}}$ pin	Provided															
	POC	Not provided															
	LVI	Not provided															
	Clock monitor	Not provided															
	WDT1	Provided															
	WDT2	Provided															
ROM correction		4 points															
Regulator		Not provided		Provided													
Standby function		HALT/IDLE/STOP/sub-IDLE mode															
Operating ambient temperature		T_A = –40 to +85°C															

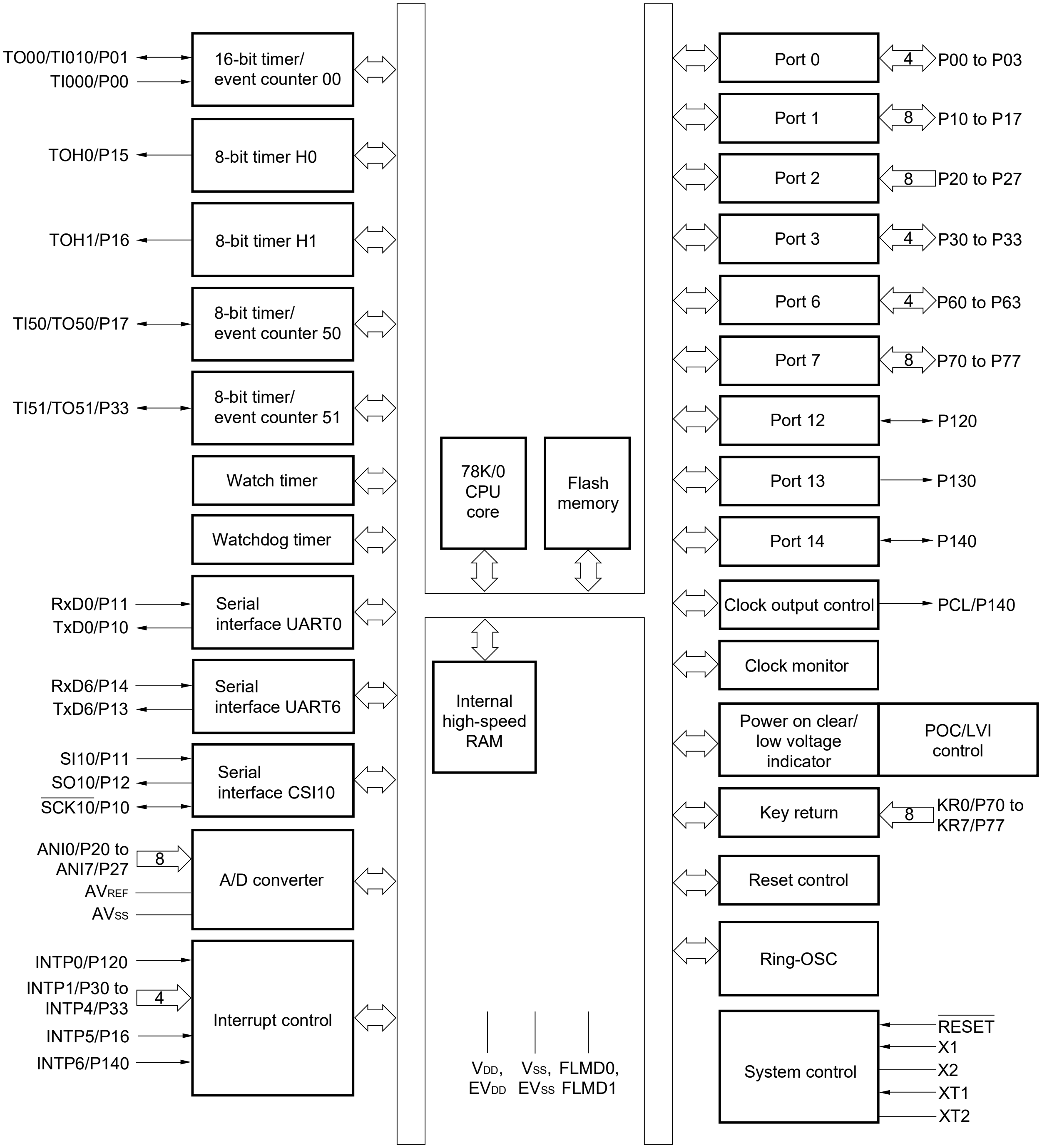
Note Provided in the Y version only.

The list of functions in the V850ES/Kx1+ is shown below.

Item / Part Number		V850ES/KE1+		V850ES/KF1+			V850ES/KG1+		V850ES/KJ1+	
Number of pins		64 pins		80 pins			100 pins		144 pins	
Internal memory (bytes)	Mask ROM	96/128	–	128	256	–	128/256	–	128/256	–
	Flash memory	–	128	–	–	256	–	256	–	256
	RAM	4		6	12		6	16	6	16
Power supply voltage		V_{DD} = 2.7 to 5.5 V								
Minimum instruction execution time		50 ns @ 20 MHz								
Clock	X1 input	2 to 10 MHz								
	Subclock	32.768 kHz								
	Ring-OSC	240 kHz (TYP.)								
Ports	CMOS input	8		8			8		16	
	CMOS I/O	43		59			76		112	
	N-ch open-drain I/O	1		2			4		6	
Timer	16 bits (TMP)	1 ch		1 ch			1 ch		1 ch	
	16 bits (TM0)	1 ch		2 ch			4 ch		6 ch	
	8 bits (TM5)	2 ch		2 ch			2 ch		2 ch	
	8 bits (TMH)	2 ch		2 ch			2 ch		2 ch	
	Interval timer	1 ch		1 ch			1 ch		1 ch	
	For watch	1 ch		1 ch			1 ch		1 ch	
	WDT1	1 ch		1 ch			1 ch		1 ch	
	WDT2	1 ch		1 ch			1 ch		1 ch	
RTO		6 bits × 1 ch		6 bits × 1 ch			6 bits × 1 ch		6 bits × 2 ch	
Serial interface	CSI	2 ch		2 ch			2 ch		3 ch	
	Automatic transmit/ receive 3-wire CSI	–		1 ch			2 ch		2 ch	
	UART	1 ch		1 ch			1 ch		2 ch	
	UART supporting LIN-bus	1 ch		1 ch			1 ch		1 ch	
	I^{2}C[Note]	1 ch		1 ch			1 ch		2 ch	
External bus	Address space	–		128 KB			3 MB		15 MB	
	Address bus	–		16 bits			22 bits		24 bits	
	Mode	–		Multiplexed mode only			Multiplexed/separate mode			
DMA controller		–		–			4 ch		4 ch	
10-bit A/D converter		8 ch		8 ch			8 ch		16 ch	
8-bit D/A converter		–		–			2 ch		2 ch	
Interrupts	External	9		9			9		9	
	Internal	27		30			42		48	
Key return input		8 ch		8 ch			8 ch		8 ch	
Reset	$\overline{\text{RESET}}$ pin	Provided								
	POC	Fixed to 2.7 V or lower								
	LVI	3.1 V/3.3 V ±0.15 V or 3.5 V/3.7 V/3.9 V/4.1 V/4.3 V ±0.2 V (selectable by software)								
	Clock monitor	Provided (monitoring by Ring-OSC)								
	WDT1	Provided								
	WDT2	Provided								
ROM correction		4 points								
Regulator		Not provided		Provided						
Standby function		HALT/IDLE/STOP/sub-IDLE mode								
Operating ambient temperature		T_A = –40 to +85°C								

Note Provided in the Y version only.

1.6 Block Diagram



1.7 Outline of Functions

(1/2)

Item		μPD78F0122H	μPD78F0123H	μPD78F0124H	μPD78F0124HD
Internal memory (bytes)	Flash memory (self programming supported)[Note 1]	16 KB	24 KB	32 KB	
	High-speed RAM[Note 1]	512 bytes	1 KB		
Memory space		64 KB			
High-speed system clock (oscillation frequency)		Crystal/ceramic/external clock oscillation (2 to 16 MHz: V_{DD} = 4.0 to 5.5 V, 2 to 8.38 MHz: V_{DD} = 3.3 to 5.5 V, 2 to 5 MHz: V_{DD} = 2.7 to 5.5 V)			
Ring-OSC clock (oscillation frequency)		On-chip Ring oscillation (240 kHz (TYP.): V_{DD} = 2.2 to 5.5 V)			
Subsystem clock (oscillation frequency)		Crystal/external clock oscillation (32.768 kHz: V_{DD} = 2.2 to 5.5 V)			
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.125 μs/0.25 μs/0.5 μs/1.0 μs/2.0 μs (high-speed system clock: @ f_{XP} = 16 MHz operation)			
		8.3 μs/16.6 μs/33.2 μs/66.4 μs/132.8 μs (TYP.) (Ring-OSC clock: @ f_R = 240 kHz (TYP.) operation)			
		122 μs (subsystem clock: @ f_{XT} = 32.768 kHz operation)			
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.			
I/O ports		Total: 39 CMOS I/O 26 CMOS input 8 CMOS output 1 N-ch open-drain I/O 4			
Timers		• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • 8-bit timer: 2 channels • Watch timer 1 channel • Watchdog timer: 1 channel			
	Timer outputs	5 (PWM output: 3)			
Clock output		• 78.125 kHz, 156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (high-speed system clock: 10 MHz) • 32.768 kHz (subsystem clock: 32.768 kHz)			
A/D converter		10-bit resolution × 8 channels			
Serial interface		• UART mode supporting LIN-bus: 1 channel • 3-wire serial I/O mode/UART mode[Note 2]: 1 channel			

Notes 1. The internal flash memory capacity and internal high-speed RAM capacity can be changed using the internal memory size switching register (IMS).

2. Select either of the functions of these alternate-function pins.

(2/2)

Item		μPD78F0122H	μPD78F0123H	μPD78F0124H	μPD78F0124HD
Vectored interrupt sources	Internal	15			
	External	8			
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of key input pins (KR0 to KR7).			
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by clock monitor • Internal reset by power-on-clear • Internal reset by low-voltage detector			
On-chip debug fucntion		–			Provided
Supply voltage		V_{DD} = 2.7 to 5.5 V (with Ring-OSC clock or subsystem clock: V_{DD} = 2.0 to 5.5 V[Note])			
Operating ambient temperature		T_A = –40 to +85°C			
Package		52-pin plastic LQFP (10 × 10)			

Note Use the product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the power-on clear (POC) circuit is 2.1 V ±0.1 V.

Caution The operating voltage range may be changed after evaluation of the device.

An outline of the timer is shown below.

		16-Bit Timer/ Event Counter 00	8-Bit Timer/ Event Counters 50 and 51		8-Bit Timers H0 and H1		Watch Timer	Watchdog Timer
		TM00	TM50	TM51	TMH0	TMH1		
Operation mode	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel[Note]	1 channel
	External event counter	1 channel	1 channel	1 channel	–	–	–	–
Function	Timer output	1 output	1 output	1 output	1 output	1 output	–	–
	PPG output	1 output	–	–	–	–	–	–
	PWM output	–	1 output	1 output	1 output	1 output	–	–
	Pulse width measurement	2 inputs	–	–	–	–	–	–
	Square-wave output	1 output	1 output	1 output	1 output	1 output	–	–
	Interrupt source	2	1	1	1	1	1	–

Note The watch timer function and interval timer function can be used simultaneously.

Remark TM51 and TMH1 can be used in combination as a carrier generator mode.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

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

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
EV_{DD}	Port pins other than P20 to P27
V_{DD}	Pins other than port pins

(1) Port pins

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 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.	Input	TI000
P01				TI010/TO00
P02				–
P03				–
P10	I/O	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.	Input	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50/FLMD1
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30 to P32	I/O	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.	Input	INTP1 to INTP3
P33				INTP4/TI51/TO51
P60 to P63	I/O	Port 6. 4-bit I/O port (N-ch open drain). Input/output can be specified in 1-bit units.	Input	–
P70 to P77	I/O	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.	Input	KR0 to KR7
P120	I/O	Port 12. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0
P130	Output	Port 13. 1-bit output-only port.	Output	–
P140	I/O	Port 14. 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.	Input	PCL/INTP6

(2) Non-port pins

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P120
INTP1 to INTP3				P30 to P32
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
INTP6				P140/PCL
SI10	Input	Serial data input to serial interface	Input	P11/RxD0
SO10	Output	Serial data output from serial interface	Input	P12
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	Input	P10/TxD0
RxD0	Input	Serial data input to asynchronous serial interface	Input	P11/SI10
RxD6				P14
TxD0	Output	Serial data output from asynchronous serial interface	Input	P10/$\overline{\text{SCK10}}$
TxD6				P13
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input	P00
TI010		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P01/TO00
TO00	Output	16-bit timer/event counter 00 output	Input	P01/TI010
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P17/TO50/FLMD1
TI51		External count clock input to 8-bit timer/event counter 51		P33/TO51/INTP4
TO50	Output	8-bit timer/event counter 50 output	Input	P17/TI50/FLMD1
TO51		8-bit timer/event counter 51 output		P33/TI51/INTP4
TOH0		8-bit timer H0 output		P15
TOH1		8-bit timer H1 output		P16/INTP5
PCL	Output	Clock output (for trimming of high-speed system clock, subsystem clock)	Input	P140/INTP6
ANI0 to ANI7	Input	A/D converter analog input	Input	P20 to P27
AV$_{\text{REF}}$	Input	A/D converter reference voltage input and positive power supply for port 2	–	–
AV$_{\text{SS}}$	–	A/D converter ground potential. Make the same potential as EV$_{\text{SS}}$ or V$_{\text{SS}}$.	–	–
KR0 to KR7	Input	Key interrupt input	Input	P70 to P77
$\overline{\text{RESET}}$	Input	System reset input	–	–
X1	Input	Connecting resonator for high-speed system clock	–	–
X2	–		–	–
XT1	Input	Connecting resonator for subsystem clock	–	–
XT2	–		–	–
V$_{\text{DD}}$	–	Positive power supply (except for ports)	–	–
EV$_{\text{DD}}$	–	Positive power supply for ports	–	–
V$_{\text{SS}}$	–	Ground potential (except for ports)	–	–
EV$_{\text{SS}}$	–	Ground potential for ports	–	–
FLMD0	–	Flash memory programming mode setting.	–	–
FLMD1			Input	P17/TI50/TO50
NC	–	Not internally connected. Leave open.	–	–

2.2 Description of Pin Functions

2.2.1 P00 to P03 (port 0)

P00 to P03 function as a 4-bit I/O port. These pins also function as timer I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P00 to P03 function as a 4-bit I/O port. P00 to P03 can be set to input or output in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

(2) Control mode

P00 to P03 function as timer I/O.

(a) TI000

This is the pin for inputting an external count clock to 16-bit timer/event counter 00 and is also for inputting a capture trigger signal to the capture registers (CR000, CR010) of 16-bit timer/event counter 00.

(b) TI010

This is the pin for inputting a capture trigger signal to the capture register (CR000) of 16-bit timer/event counter 00.

(c) TO00

This is a timer output pin.

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and flash memory programming mode setting.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and flash memory programming mode setting.

(a) SI10

This is a serial interface serial data input pin.

(b) SO10

This is a serial interface serial data output pin.

(c) $\overline{\textbf{SCK10}}$

This is a serial interface serial clock I/O pin.

(d) RxD0, RxD6

These are serial data input pins of the asynchronous serial interface.

(e) TxD0, TxD6

These are serial data output pins of the asynchronous serial interface.

(f) TI50

This is a pin for inputting an external count clock to 8-bit timer/event counter 50.

(g) TO50, TOH0, and TOH1

These are timer output pins.

(h) INTP5

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(i) FLMD1

This is the pin for setting the flash memory programming mode.

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an 8-bit input-only port. These pins also function as pins for A/D converter analog input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P20 to P27 function as an 8-bit input-only port.

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, refer to **(5) ANI0/P20 to ANI7/P27** in **12.6 Cautions for A/D Converter.**

2.2.4 P30 to P33 (port 3)

P30 to P33 function as a 4-bit I/O port. These pins also function as pins for external interrupt request input and timer I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P30 to P33 function as a 4-bit I/O port. P30 to P33 can be set to input or output in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P33 function as external interrupt request input pins and timer I/O pins.

(a) INTP1 to INTP4

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) TI51

This is an external count clock input pin to 8-bit timer/event counter 51.

(c) TO51

This is a timer output pin.

Caution In the μPD78F0124HD, be sure to pull the P31 pin down after reset to prevent malfunction.

Remark P31/INTP2 and P32/INTP3 of the μPD78F0124HD can be used as on-chip debug mode setting pins when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION (μPD78F0124HD ONLY)**.

2.2.5 P60 to P63 (port 6)

P60 to P63 function as a 4-bit I/O port. P60 to P63 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

P60 to P63 are N-ch open-drain pins.

2.2.6 P70 to P77 (port 7)

P70 to P77 function as an 8-bit I/O port. These pins also function as key interrupt input pins.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P70 to P77 function as an 8-bit I/O port. P70 to P77 can be set to input or output in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

(2) Control mode

P70 to P77 function as key interrupt input pins.

2.2.7 P120 (port 12)

P120 functions as a 1-bit I/O port. This pin also functions as a pin for external interrupt request input.

The following operation modes can be specified.

(1) Port mode

P120 functions as a 1-bit I/O port. P120 can be set to input or output using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

(2) Control mode

P120 functions as an external interrupt request input pin (INTP0) for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

2.2.8 P130 (port 13)

P130 functions as a 1-bit output-only port.

2.2.9 P140 (port 14)

P140 functions as a 1-bit I/O port. This pin also functions as a pin for external interrupt request input and clock output.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P140 functions as a 1-bit I/O port. P140 can be set to input or output in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

(2) Control mode

P140 functions as external interrupt request input and clock output.

(a) INTP6

This is the external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) PCL

This is a clock output pin.

2.2.10 $\mathrm{AV_{REF}}$

This is the A/D converter reference voltage input pin.

When A/D converter is not used, connect this pin to $\mathrm{EV_{DD}}$ or $\mathrm{V_{DD}}$[Note].

Note Connect port 2 directly to $\mathrm{EV_{DD}}$ when it is used as a digital port.

2.2.11 $\mathrm{AV_{SS}}$

This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the $\mathrm{EV_{SS}}$ pin or $\mathrm{V_{SS}}$ pin.

2.2.12 $\overline{\mathrm{RESET}}$

This is the active-low system reset input pin.

2.2.13 X1 and X2

These are the pins for connecting a resonator for high-speed system clock oscillation.

When supplying an external clock, input a signal to the X1 pin and input the inverse signal to the X2 pin.

Remark X1 and X2 of the μPD78F0124HD can be used as on-chip debug mode setting pins when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION (μPD78F0124HD ONLY)**.

2.2.14 XT1 and XT2

These are the pins for connecting a resonator for subsystem clock oscillation.

When supplying an external clock, input a signal to the XT1 pin and input the inverse signal to the XT2 pin.

2.2.15 $\mathrm{V_{DD}}$ and $\mathrm{EV_{DD}}$

$\mathrm{V_{DD}}$ is the positive power supply pin for other than ports.

$\mathrm{EV_{DD}}$ is the positive power supply pin for ports.

2.2.16 $\mathrm{V_{SS}}$ and $\mathrm{EV_{SS}}$

$\mathrm{V_{SS}}$ is the ground potential pin for other than ports.

$\mathrm{EV_{SS}}$ is the ground potential pin for ports.

2.2.17 FLMD0 and FLMD1

These pins set the flash memory programming mode.

Connect FLMD0 to $\mathrm{EV_{SS}}$ or $\mathrm{V_{SS}}$ in the normal operation mode (FLMD1 is not used in the normal operation mode).

In flash memory programming mode, be sure to connect these pins to the flash programmer.

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

Table 2-2 shows the types of pin I/O circuits and the recommended connections of unused pins.
Refer to **Figure 2-1** for the configuration of the I/O circuit of each type.

Table 2-2. Pin I/O Circuit Types

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI000	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TI010/TO00			
P02			
P03			
P10/$\overline{\text{SCK10}}$/TxD0			
P11/SI10/RxD0			
P12/SO10	5-A		
P13/TxD6			
P14/RxD6	8-A		
P15/TOH0	5-A		
P16/TOH1/INTP5	8-A		
P17/TI50/TO50/FLMD1			
P20/ANI0 to P27/ANI7	9-C	Input	Connect to EV_{DD} or EV_{SS}.
P30/INTP1	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/INTP2 (except μPD78F0124HD)			
P31/INTP2 (μPD78F0124HD)			Connect to EV_{SS} via a resistor.
P32/INTP3			Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P33/TI51/TO51/INTP4			
P60, P61	13-R		Input: Connect to EV_{SS}. Output: Leave this pin open at low-level output after clearing the output latch of the port to 0.
P62, P63	13-W		
P70/KR0 to P77/KR7	8-A		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P120/INTP0			
P130	3-C	Output	Leave open.
P140/PCL/INTP6	8-A	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
$\overline{\text{RESET}}$	2	Input	–
XT1	16		Connect directly to EV_{SS} or V_{SS}[Note 1].
XT2		–	Leave open.
AV_{REF}	–		Connect directly to EV_{DD} or V_{DD}[Note 2].
AV_{SS}			Connect directly to EV_{SS} or V_{SS}.
FLMD0			Connect directly to EV_{SS} or V_{SS}.

Notes 1. Bit 6 (FRC) of the processor clock control register (PCC) must be set to 1 after reset mode is released.

2. Connect port 2 directly to EV_{DD} when it is used as a digital port.

Figure 2-1. Pin I/O Circuit List (1/2)

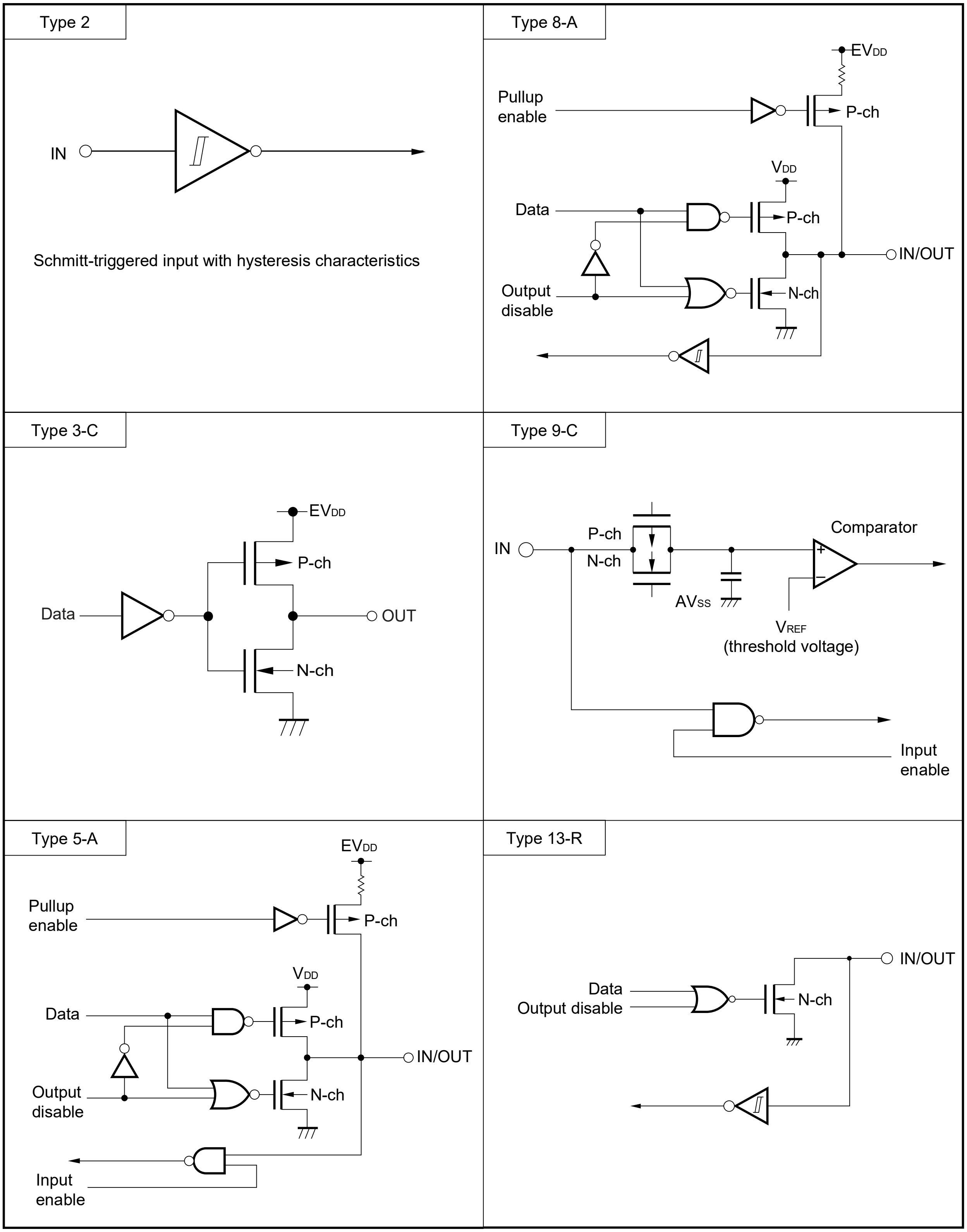
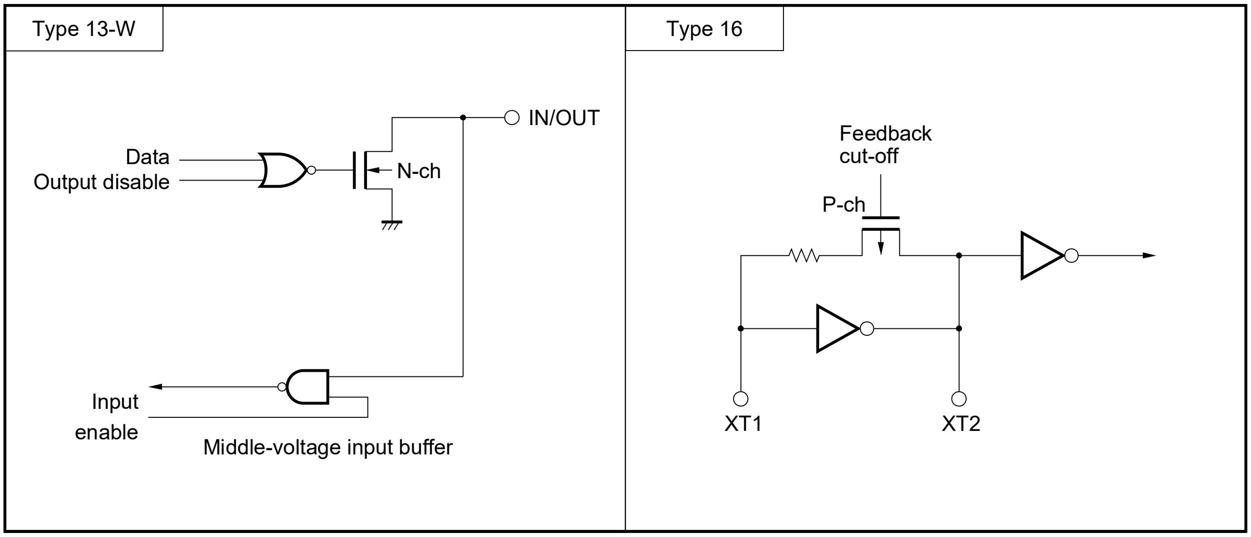


Figure 2-1. Pin I/O Circuit List (2/2)



CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

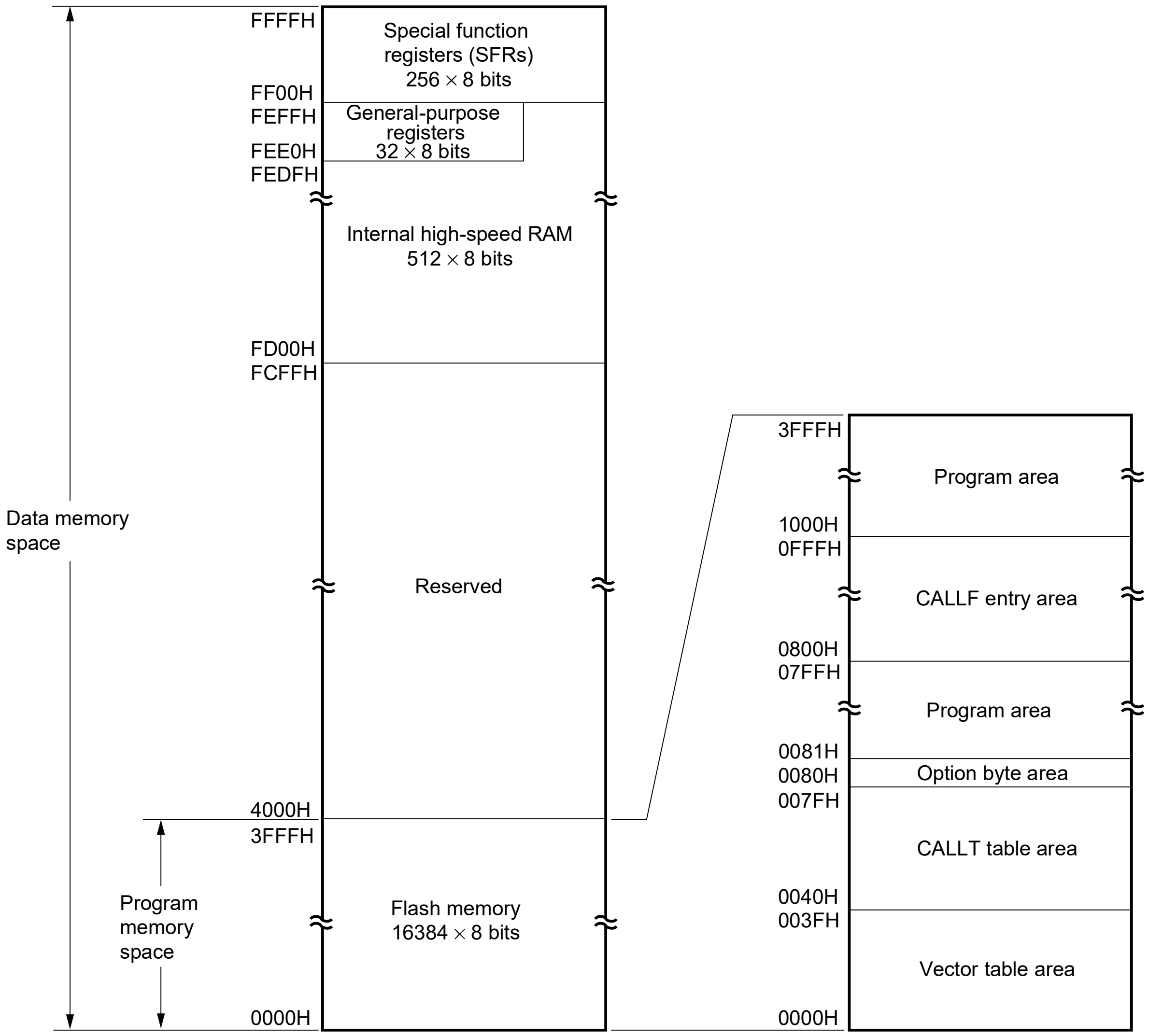
Products in the 78K0/KD1+ can each access a 64 KB memory space. Figures 3-1 to 3-4 show the memory maps.

Caution Regardless of the internal memory capacity, the initial value of the internal memory size switching register (IMS) of all products in the 78K0/KD1+ is fixed (IMS = CFH). Therefore, set the value corresponding to each product as indicated below. In addition, set the following values to the internal memory size switching register (IMS) when using the 78K0/KD1+ to evaluate the program of a mask ROM version of the 78K0/KD1.

Table 3-1. Set Values of Internal Memory Size Switching Register (IMS)

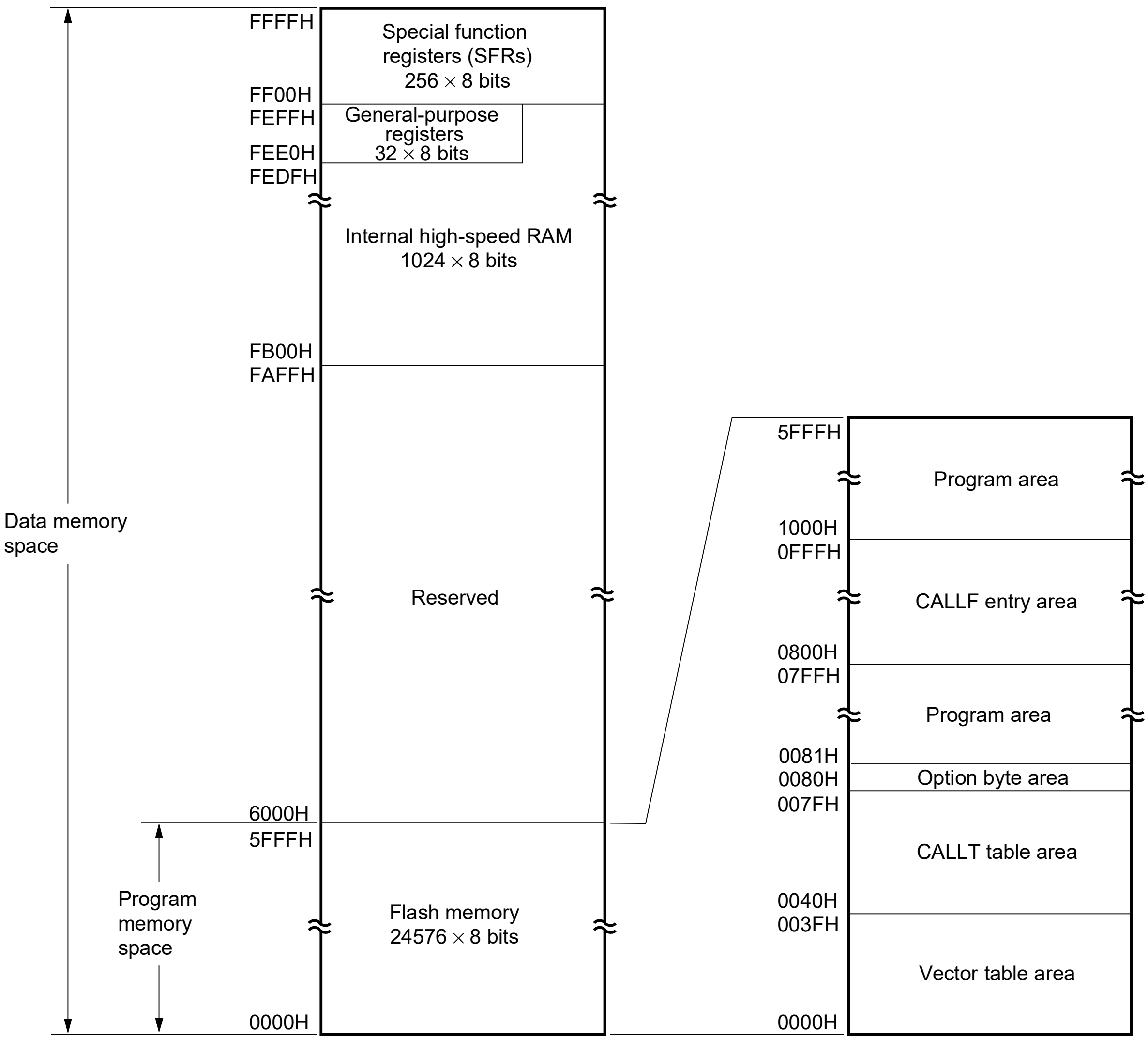
Flash Memory Version (78K0/KD1+)	Target Mask ROM Version (78K0/KD1)	Internal Memory Size Switching Register (IMS)
−	μPD780121	42H
μPD78F0122H	μPD780122	44H
μPD78F0123H	μPD780123	C6H
μPD78F0124H, 78F0124HD	μPD780124	C8H

Figure 3-1. Memory Map (μPD78F0122H)



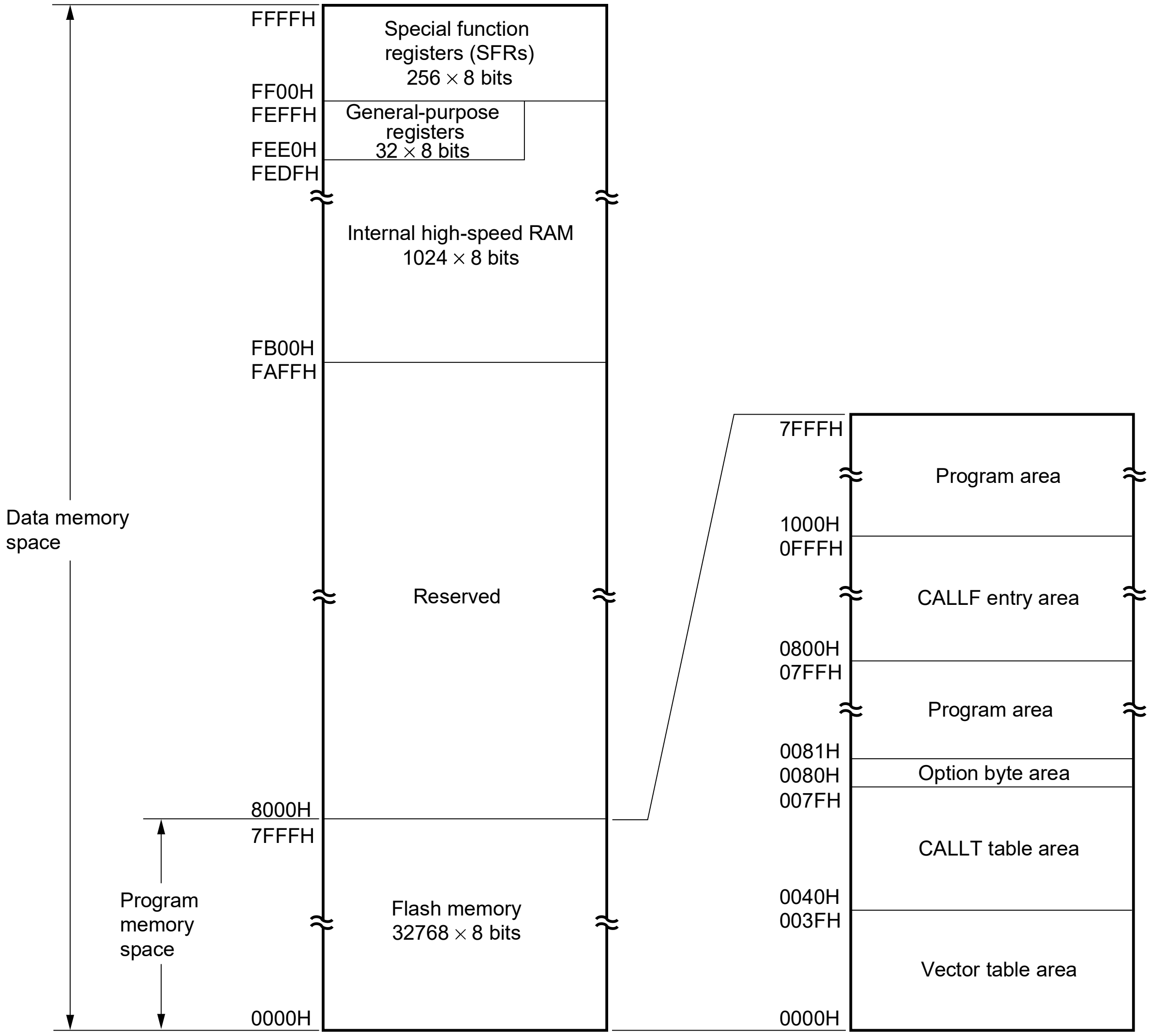
Caution When replacing the μPD78F0122H with the μPD78F0124HD, note that the area from 0081H to 0083H in the μPD78F0124HD cannot be used.

Figure 3-2. Memory Map (μPD78F0123H)



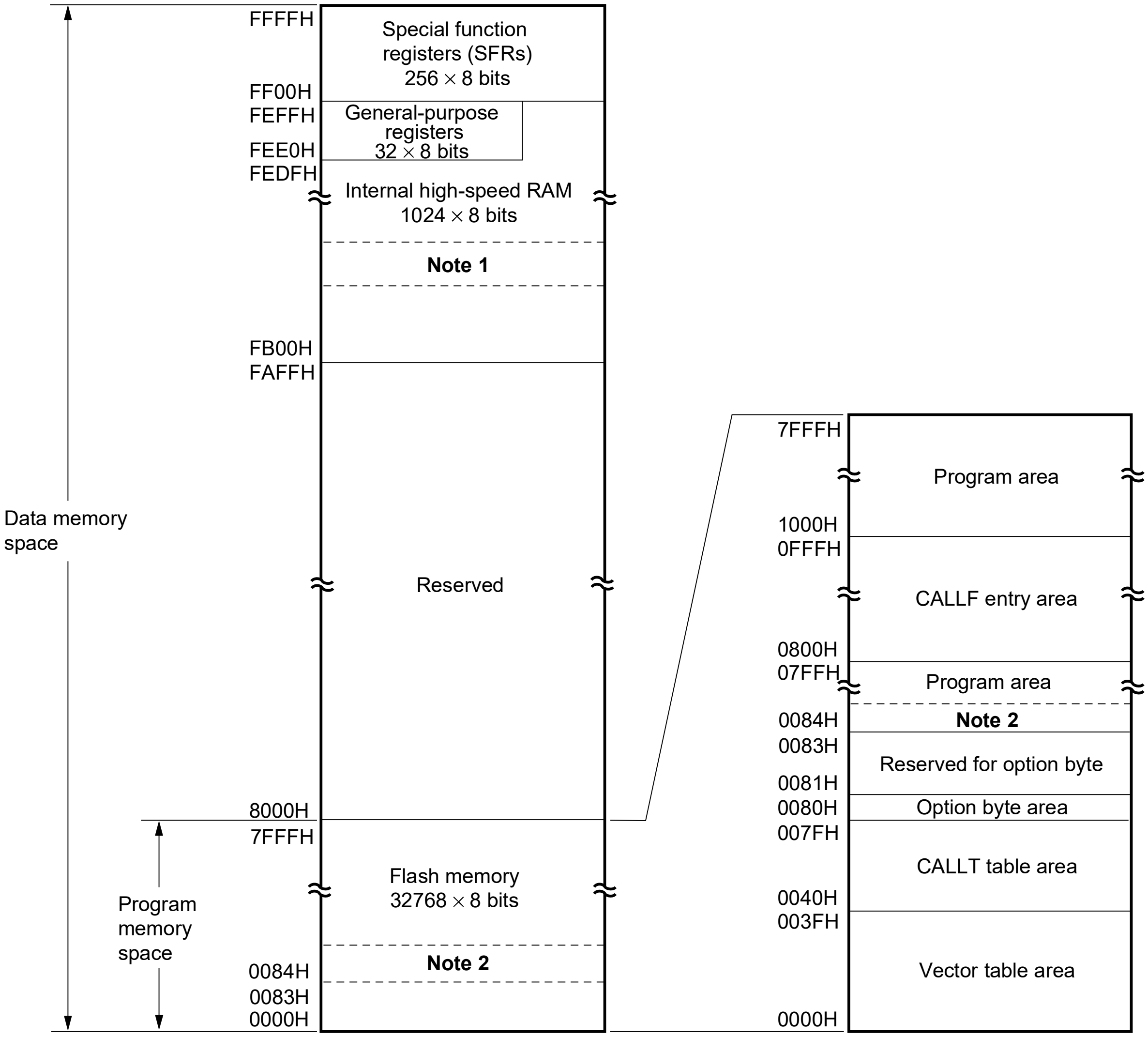
Caution When replacing the μPD78F0123H with the μPD78F0124HD, note that the area from 0081H to 0083H in the μPD78F0124HD cannot be used.

Figure 3-3. Memory Map (μPD78F0124H)



Caution When replacing the μPD78F0124H with the μPD78F0124HD, note that the area from 0081H to 0083H in the μPD78F0124HD cannot be used.

Figure 3-4. Memory Map (μPD78F0124HD)



Notes 1. During on-chip debugging, 9 bytes (planned) of this area are used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (256 bytes to 1 KB).

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

78K0/KD1+ products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0122H	Flash memory	16384 × 8 bits (0000H to 3FFFH)
μPD78F0123H		24576 × 8 bits (0000H to 5FFFH)
μPD78F0124H, 78F0124HD		32768 × 8 bits (0000H to 7FFFH)

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

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon reset signal input or generation of each interrupt request are stored in the vector table area.

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

Table 3-3. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, clock monitor, WDT	001AH	INTTMH1
		001CH	INTTMH0
0004H	INTLVI	001EH	INTTM50
0006H	INTP0	0020H	INTTM000
0008H	INTP1	0022H	INTTM010
000AH	INTP2	0024H	INTAD
000CH	INTP3	0026H	INTSR0
000EH	INTP4	0028H	INTWTI
0010H	INTP5	002AH	INTTM51
0012H	INTSRE6	002CH	INTKR
0014H	INTSR6	002EH	INTWT
0016H	INTST6	0030H	INTP6
0018H	INTCSI10/INTST0		

(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

The option byte area is assigned to the 1-byte area of 0080H. Refer to **CHAPTER 23 OPTION BYTE** for details.

(4) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

78K0/KD1+ products incorporate the following internal high-speed RAMs.

Table 3-4. Internal High-Speed RAM Capacity

Part Number	Internal High-Speed RAM
μPD78F0122H	512 × 8 bits (FD00H to FEFFH)
μPD78F0123H	1024 × 8 bits (FB00H to FEFFH)
μPD78F0124H, 78F0124HD	

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per one bank.

This area cannot be used as a program area in which instructions are written and executed.

The internal high-speed RAM can also be used as a stack memory.

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (refer to **Table 3-5 Special Function Register List** in **3.2.3 Special Function Registers (SFRs)**).

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

3.1.4 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 78K0/KD1+, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-5 to 3-8 show correspondence between data memory and addressing. For details of each addressing mode, refer to **3.4 Operand Address Addressing**.

Figure 3-5. Correspondence Between Data Memory and Addressing (μPD78F0122H)

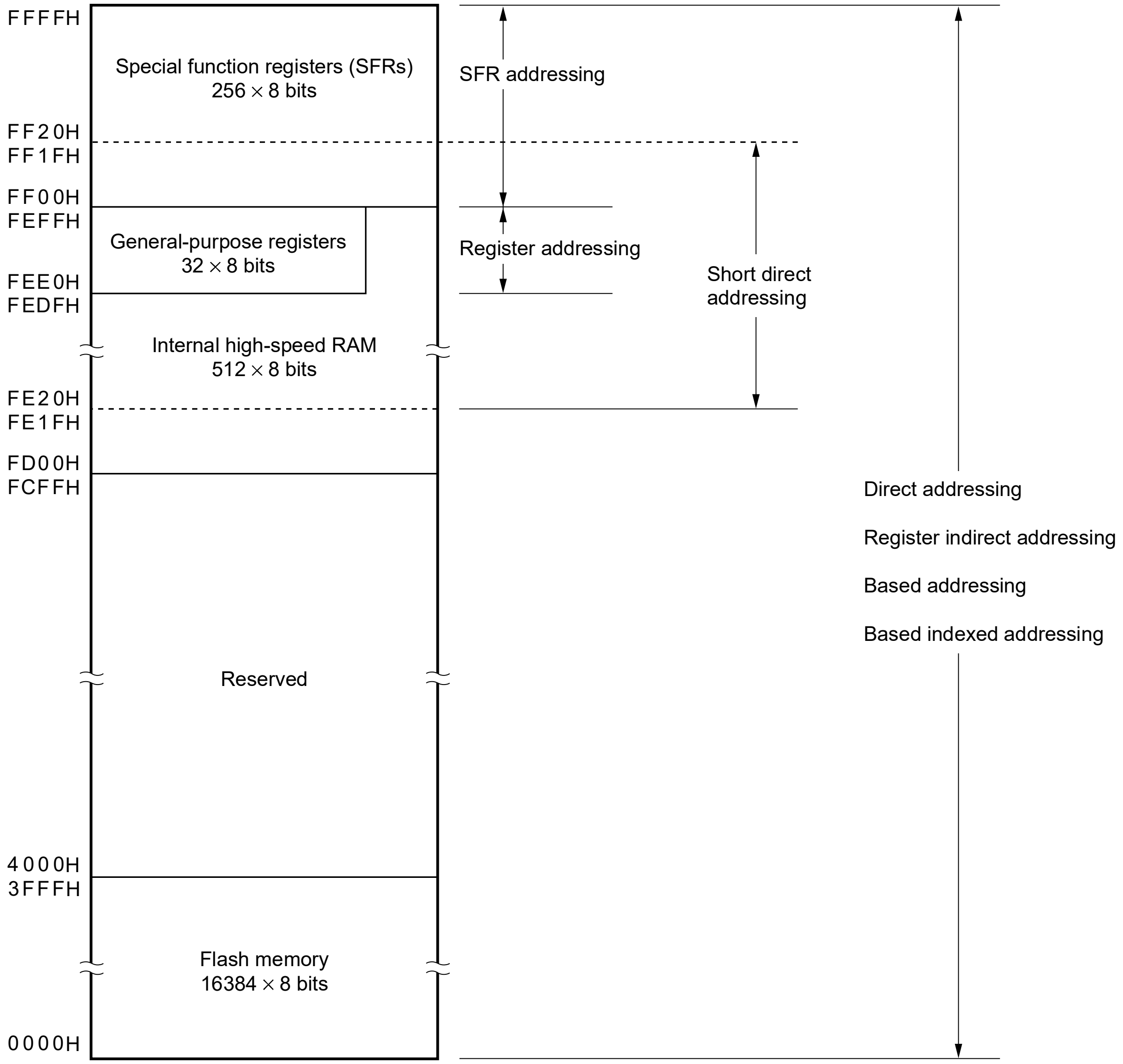


Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F0123H)

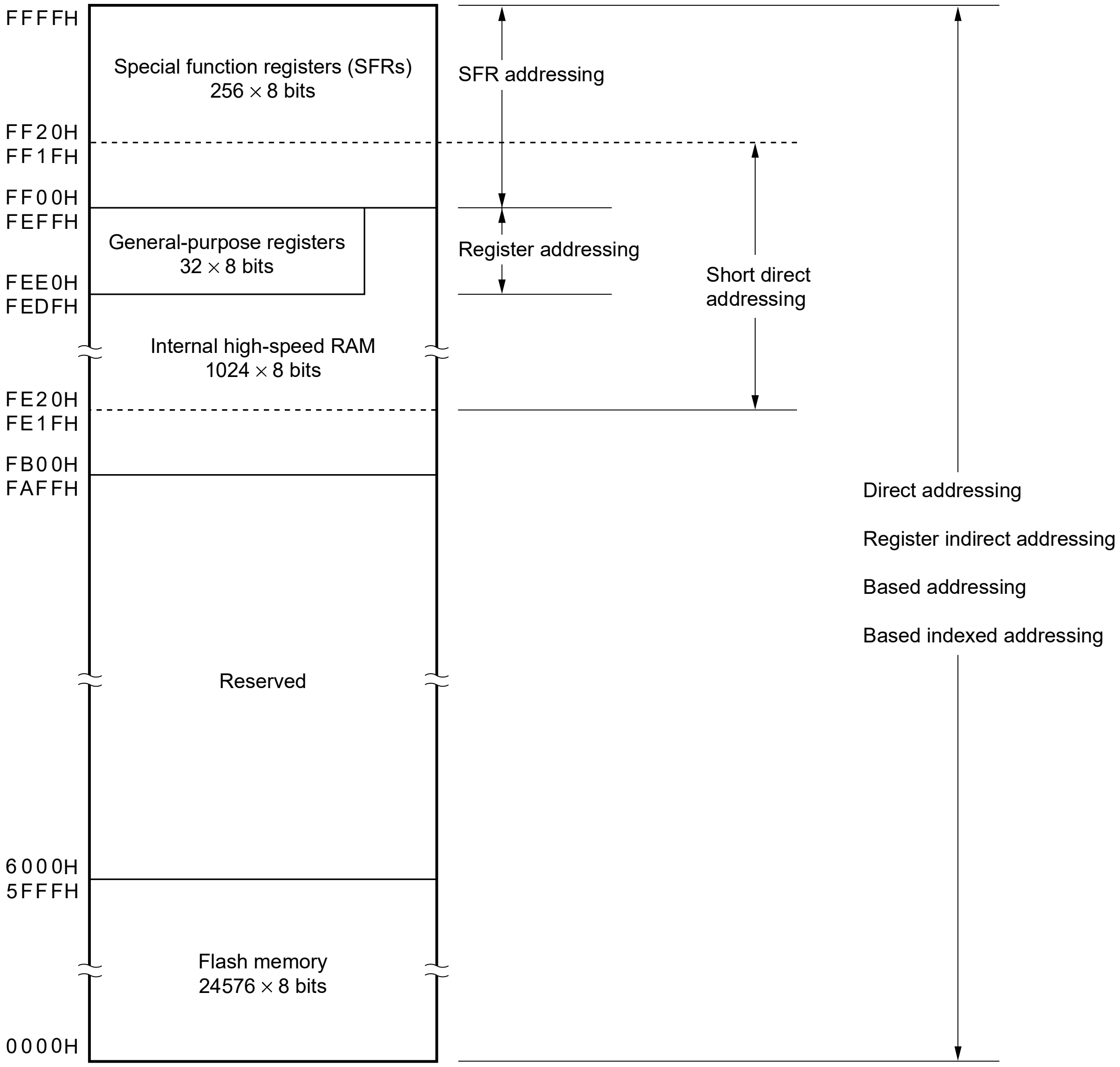


Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F0124H)

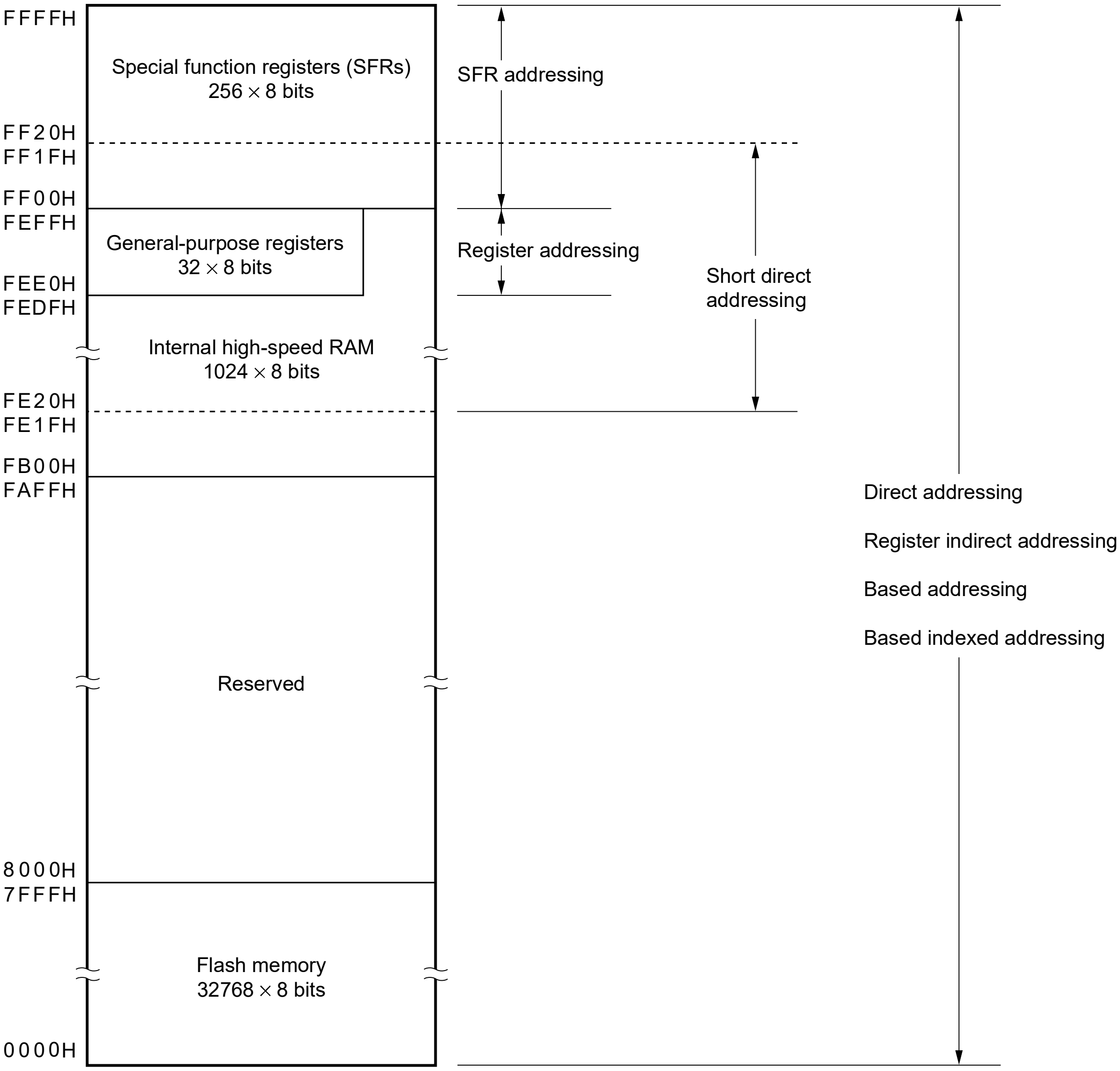
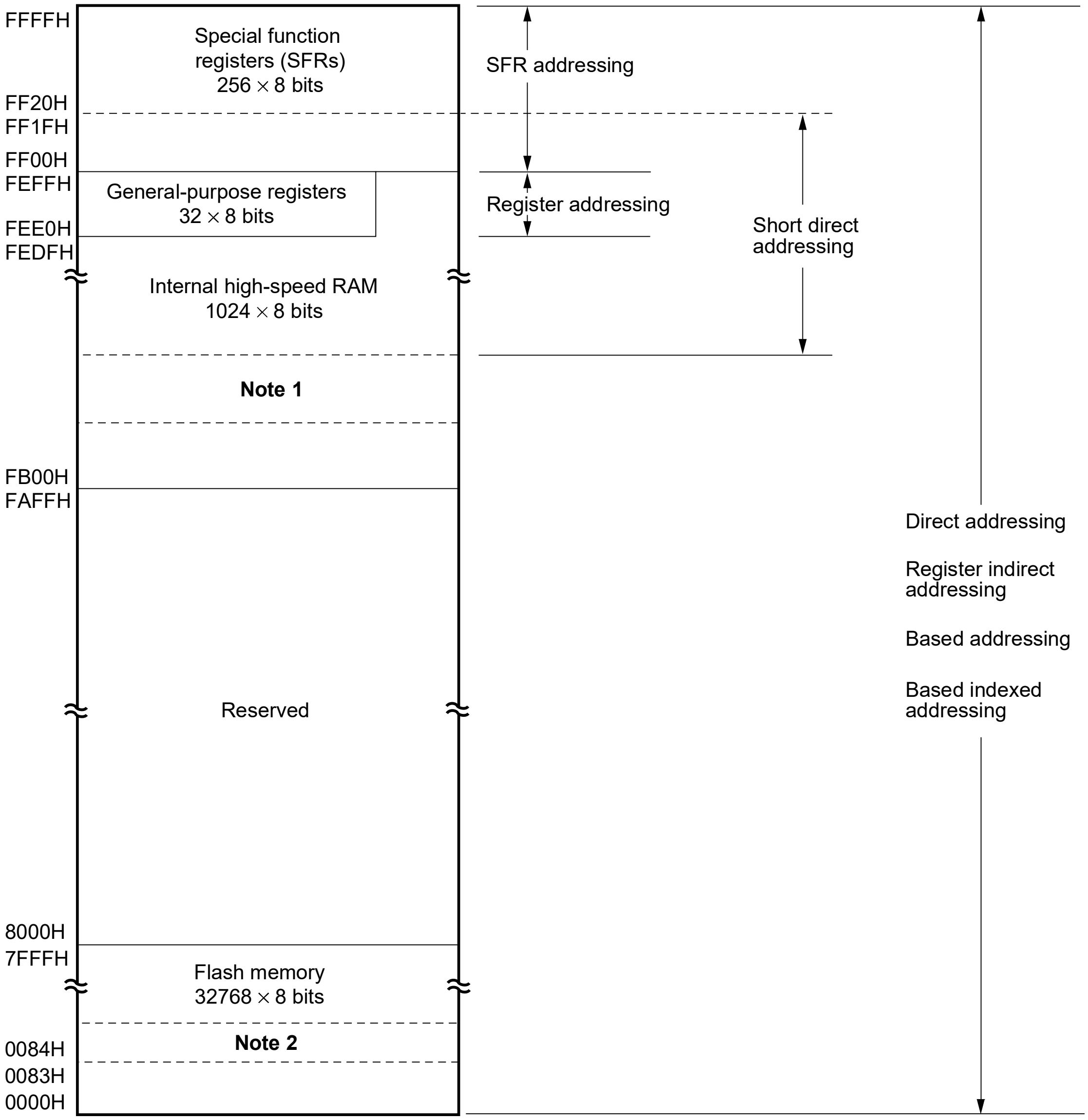


Figure 3-8. Correspondence Between Data Memory and Addressing (μPD78F0124HD)



Notes 1. During on-chip debugging, 9 bytes (planned) of this area are used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (256 bytes to 1 KB).

3.2 Processor Registers

The 78K0/KD1+ 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 16-bit register that holds the address information of the next program to be executed.

In normal operation, the 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.

$\overline{\text{RESET}}$ input sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-9. Format of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(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 automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 3-10. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	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 interrupts are disabled. Other interrupt requests are all disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgement is controlled with an in-service priority flag (ISP), 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 acknowledgement and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

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

(c) Register bank select flags (RBS0 and 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 flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (refer to **16.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L)**) can not be acknowledged. Actual request acknowledgement is controlled by the interrupt enable flag (IE).

(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 high-speed RAM area can be set as the stack area.

Figure 3-11. Format of Stack Pointer

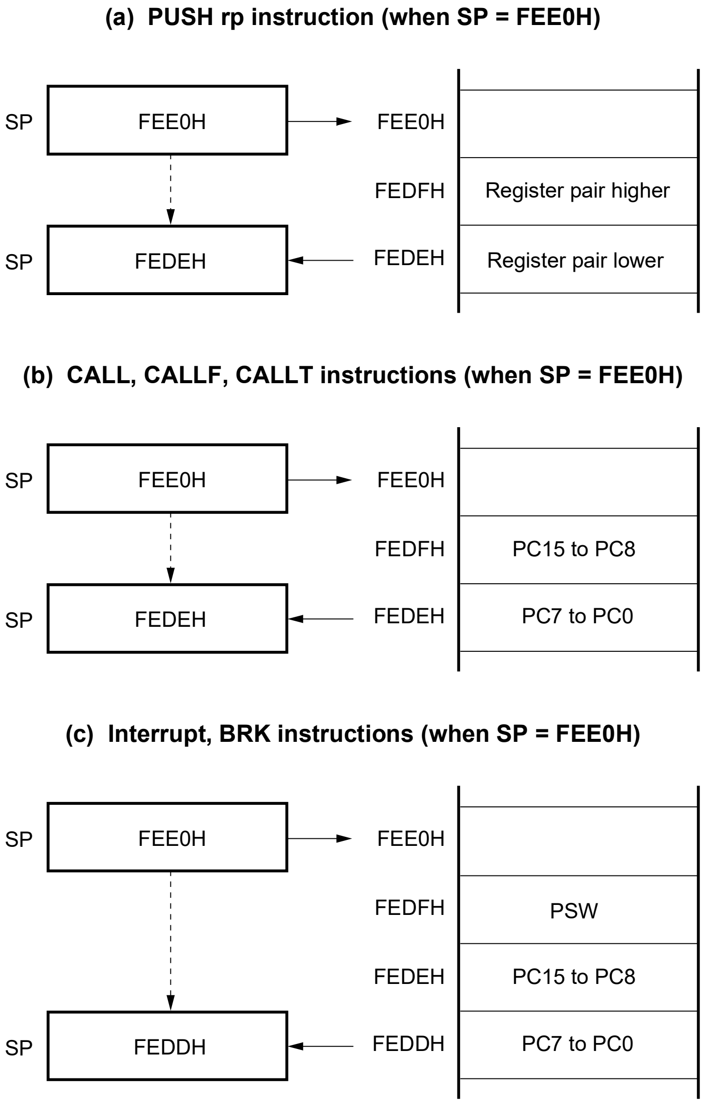
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-12 and 3-13.

Caution Since $\overline{\text{RESET}}$ input makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-12. Data to Be Saved to Stack Memory

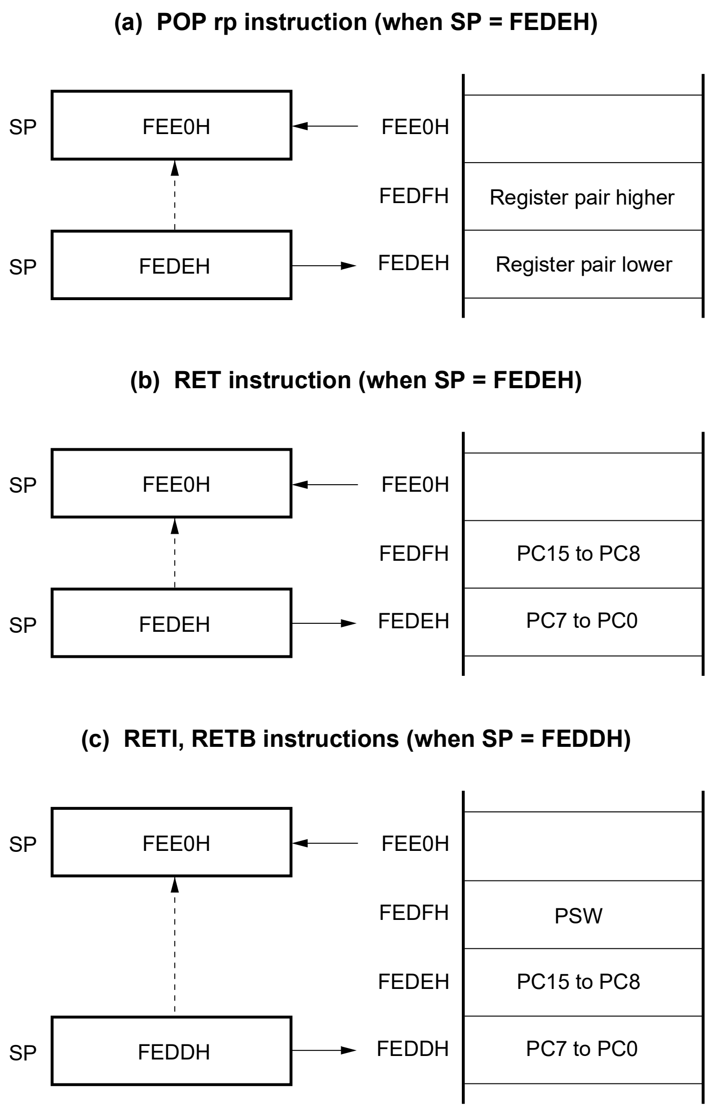


(a) PUSH rp instruction (when SP = FEE0H)

(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)

(c) Interrupt, BRK instructions (when SP = FEE0H)

Figure 3-13. Data to Be Restored from Stack Memory



(a) POP rp instruction (when SP = FEDEH)

(b) RET instruction (when SP = FEDEH)

(c) RETI, RETB instructions (when SP = FEDDH)

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) 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).

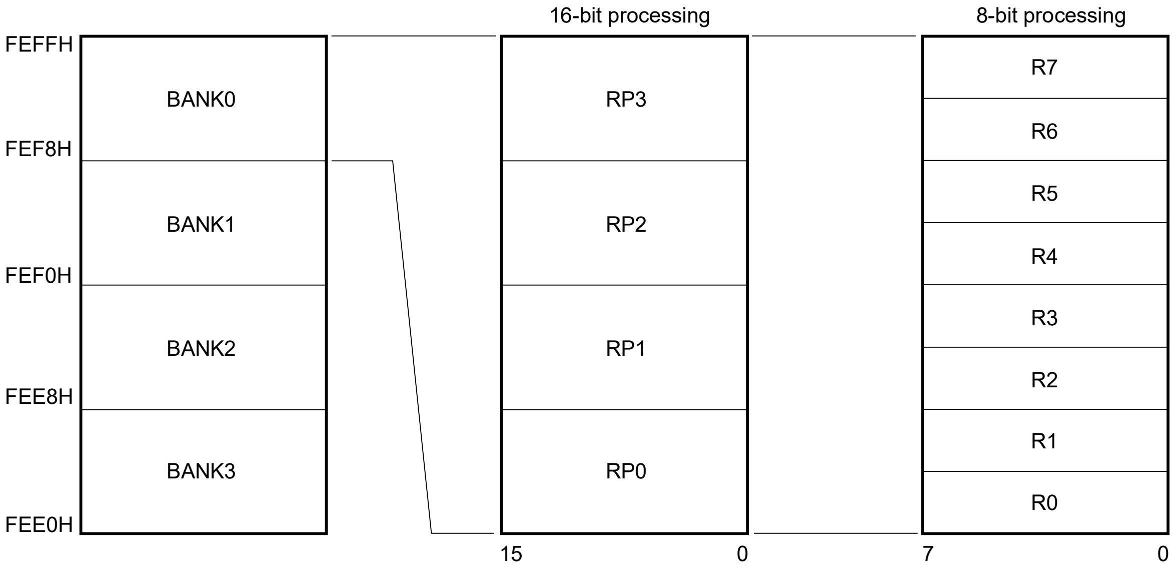
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

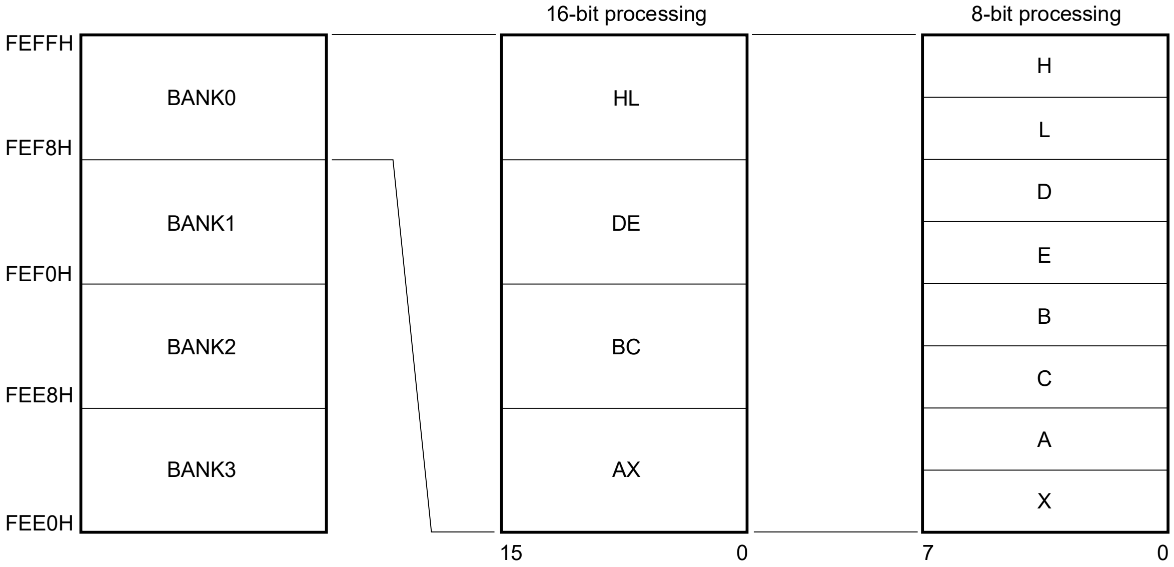
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-14. Configuration of General-Purpose Registers

(a) Absolute name



(b) Function name



3.2.3 Special function registers (SFRs)

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

SFRs are allocated to the FF00H to FFFFH area.

Special function registers can be manipulated like general-purpose registers, using operation, transfer and bit manipulation instructions. The manipulatable bit units, 1, 8, and 16, depend on the special function register type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved 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 reserved 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 special function registers. 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 RA78K0, and is defined by the header file “sfrbit.h” in the CC78K0. When using the RA78K0, ID78K0-NS, ID78K0, or SM78K0, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding special function register can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulatable bit units
  Indicates the manipulatable bit unit (1, 8, or 16). “–” indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 3-5. Special Function Register List (1/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF00H	Port register 0	P0	R/W	√	√	–	00H
FF01H	Port register 1	P1	R/W	√	√	–	00H
FF02H	Port register 2	P2	R	√	√	–	Undefined
FF03H	Port register 3	P3	R/W	√	√	–	00H
FF06H	Port register 6	P6	R/W	√	√	–	00H
FF07H	Port register 7	P7	R/W	√	√	–	00H
FF08H	A/D conversion result register	ADCR	R	–	–	√	Undefined
FF09H							
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH
FF0CH	Port register 12	P12	R/W	√	√	–	00H
FF0DH	Port register 13	P13	R/W	√	√	–	00H
FF0EH	Port register 14	P14	R/W	√	√	–	00H
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H
FF18H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H
FF19H	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH
FF27H	Port mode register 7	PM7	R/W	√	√	–	FFH
FF28H	A/D converter mode register	ADM	R/W	√	√	–	00H
FF29H	Analog input channel specification register	ADS	R/W	√	√	–	00H
FF2AH	Power-fail comparison mode register	PFM	R/W	√	√	–	00H
FF2BH	Power-fail comparison threshold register	PFT	R/W	–	√	–	00H
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH
FF2EH	Port mode register 14	PM14	R/W	√	√	–	FFH
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H

Table 3-5. Special Function Register List (2/3)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
FF3EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H
FF50H	Asynchronous serial interface operation mode register 6	ASIM6	R/W	√	√	–	01H
FF53H	Asynchronous serial interface reception error status register 6	ASIS6	R	–	√	–	00H
FF55H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H
FF56H	Clock selection register 6	CKSR6	R/W	–	√	–	00H
FF57H	Baud rate generator control register 6	BRGC6	R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 6	ASICL6	R/W	√	√	–	16H
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50	R/W	–	√	–	00H
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H
FF6DH	8-bit timer H carrier control register 1	TMCYC1	R/W	√	√	–	00H
FF6EH	Key return mode register	KRM	R/W	√	√	–	00H
FF6FH	Watch timer operation mode register	WTM	R/W	√	√	–	00H
FF70H	Asynchronous serial interface operation mode register 0	ASIM0	R/W	√	√	–	01H
FF71H	Baud rate generator control register 0	BRGC0	R/W	–	√	–	1FH
FF72H	Receive buffer register 0	RXB0	R	–	√	–	FFH
FF73H	Asynchronous serial interface reception error status register 0	ASIS0	R	–	√	–	00H
FF74H	Transmit shift register 0	TXS0	W	–	√	–	FFH
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	Undefined
FF8CH	Timer clock selection register 51	TCL51	R/W	–	√	–	00H
FF98H	Watchdog timer mode register	WDTM	R/W	–	√	–	67H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	9AH
FFA0H	Ring-OSC mode register	RCM	R/W	√	√	–	00H
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC	R/W	√	√	–	00H
FFA3H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	05H
FFA9H	Clock monitor mode register	CLM	R/W	√	√	–	00H

Table 3-5. Special Function Register List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FFACH	Reset control flag register	RESF		R	–	√	–	00H[Note 1]
FFBAH	16-bit timer mode control register 00	TMC00		R/W	√	√	–	00H
FFBBH	Prescaler mode register 00	PRM00		R/W	√	√	–	00H
FFBCH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H
FFBDH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H
FFBFH	Low-voltage detection level selection register	LVIS		R/W	–	√	–	00H
FFC0H	Flash protect command register	PFCMD		W	×	√	×	Undefined
FFC2H	Flash status register	PFS		R/W	√	√	×	00H
FFC4H	Flash programming mode control register	FLPMC		R/W	√	√	×	0XH[Note 2]
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
FFE2H	Interrupt request flag register 1L	IF1L		R/W	√	√	–	00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
FFE6H	Interrupt mask flag register 1L	MK1L		R/W	√	√	–	FFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
FFEAH	Priority specification flag register 1L	PR1L		R/W	√	√	–	FFH
FFF0H	Internal memory size switching register[Note 3]	IMS		R/W	–	√	–	CFH
FFFBH	Processor clock control register	PCC		R/W	√	√	–	00H
FFFDH	System wait control register	VSWC		R/W	√	√	×	00H[Note 4]

Notes 1. This value varies depending on the reset source.

2. This value differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

3. Regardless of the internal memory capacity, the initial value of the internal memory size switching register (IMS) of all products in the 78K0/KD1+ is fixed (IMS = CFH). Therefore, set the value corresponding to each product as indicated below. In addition, set the following values to the internal memory size switching register (IMS) when using the 78K0/KD1+ to evaluate the program of a mask ROM version of the 78K0/KD1.

Flash Memory Version (78K0/KD1+)	Target Mask ROM Version (78K0/KD1)	Internal Memory Size Switching Register (IMS)
–	μPD780121	42H
μPD78F0122H	μPD780122	44H
μPD78F0123H	μPD780123	C6H
μPD78F0124H, 78F0124HD	μPD780124	C8H

4. Do not access VSWC in the current IE environment (IE-78K0-NS, IE-78K0-NS-A, and IE-78K0K1-ET).

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (for details of instructions, refer to **78K/0 Series Instructions User's Manual (U12326E)**).

3.3.1 Relative addressing

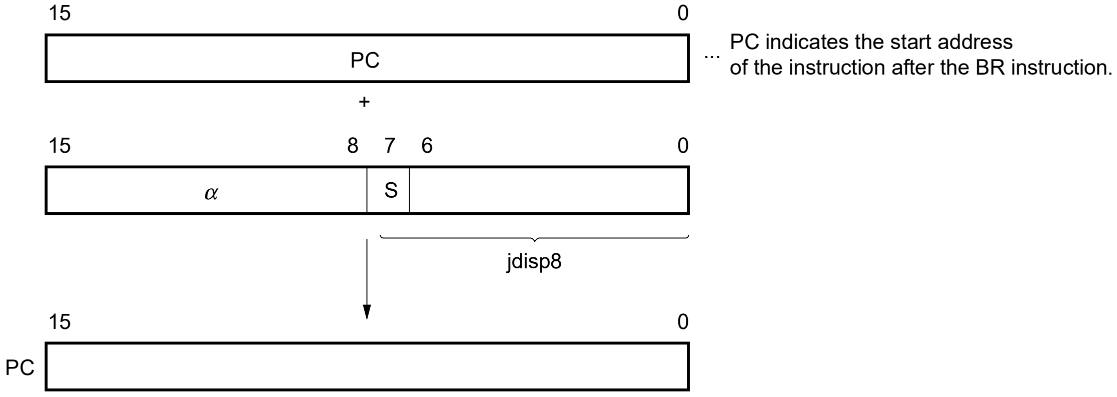
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit.

In other words, relative addressing consists of relative branching from the start address of the following instruction to the –128 to +127 range.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]


15
0
PC
... PC indicates the start address of the instruction after the BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

When S = 0, all bits of α are 0.
When S = 1, all bits of α are 1.

3.3.2 Immediate addressing

[Function]

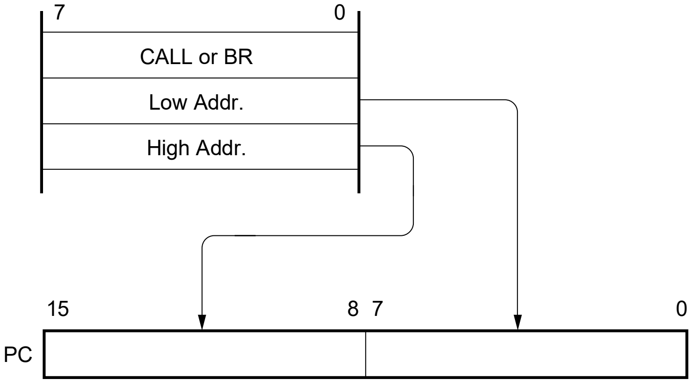
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

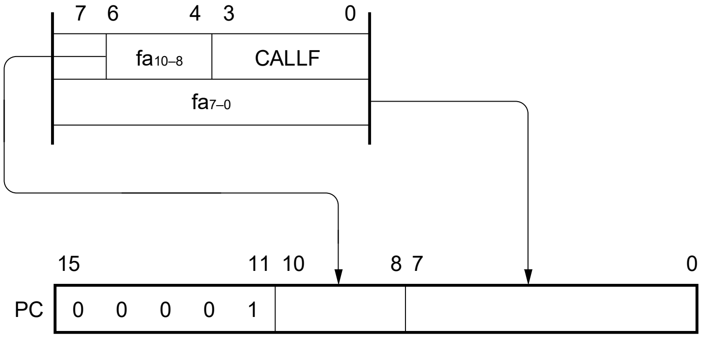
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



3.3.3 Table indirect addressing

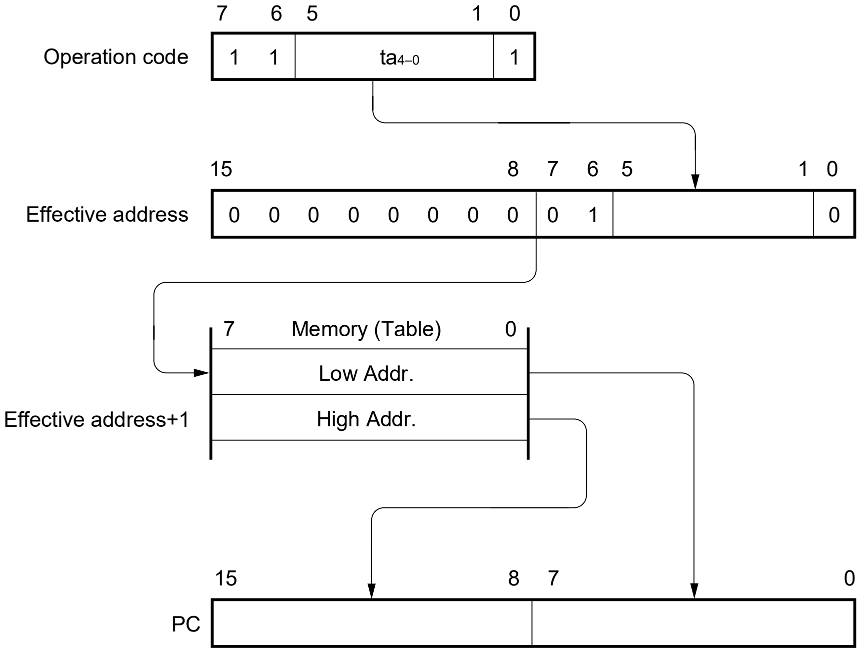
[Function]

Table contents (branch destination address) of the particular location to be addressed by bits 1 to 5 of the immediate data of an operation code are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address stored in the memory table from 40H to 7FH, and allows branching to the entire memory space.

[Illustration]



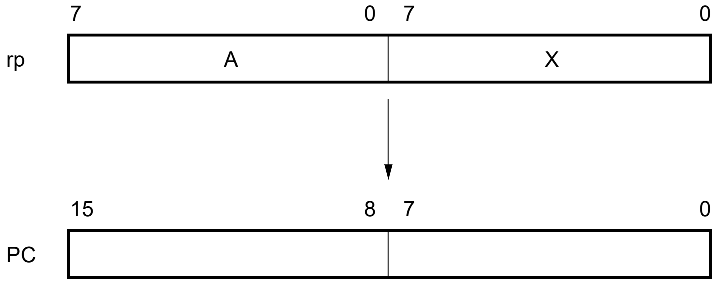
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

The register that functions as an accumulator (A and AX) among the general-purpose registers is automatically (implicitly) addressed.

Of the 78K0/KD1+ instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage
DIVUW	AX register for dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values that become decimal correction targets
ROR4/ROL4	A register for storage of digit data that undergoes digit rotation

[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

[Description example]

In the case of MULU X

With an 8-bit × 8-bit multiply instruction, the product of A register and X register is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes (Rn and RPn) of an operation code.

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

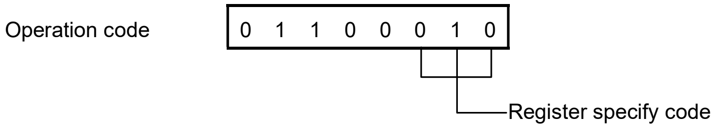
[Operand format]

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

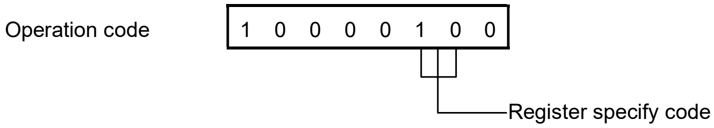
‘r’ and ‘rp’ can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

[Operand format]

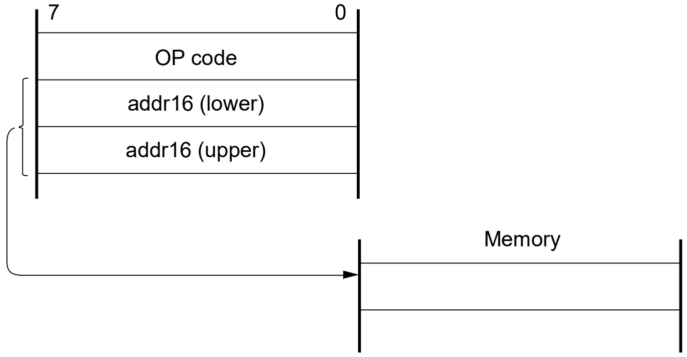
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.
This addressing is applied to the 256-byte space FE20H to FF1FH. Internal RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.
The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.
When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]**.

[Operand format]

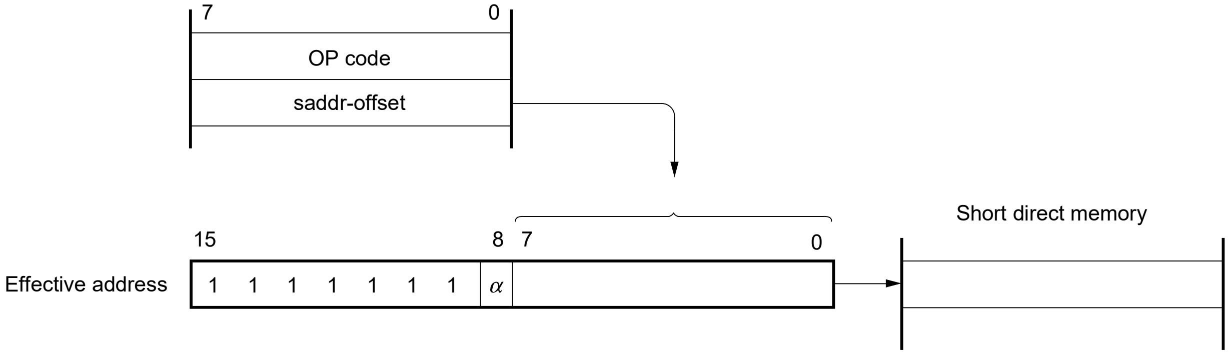
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1FH (even address only)

[Description example]

MOV 0FE30H, A; when transferring value of A register to saddr (FE30H)

Operation code	1 1 1 1 0 0 1 0	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)

[Illustration]



When 8-bit immediate data is 20H to FFH, α = 0
When 8-bit immediate data is 00H to 1FH, α = 1

3.4.5 Special function register (SFR) addressing

[Function]

A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

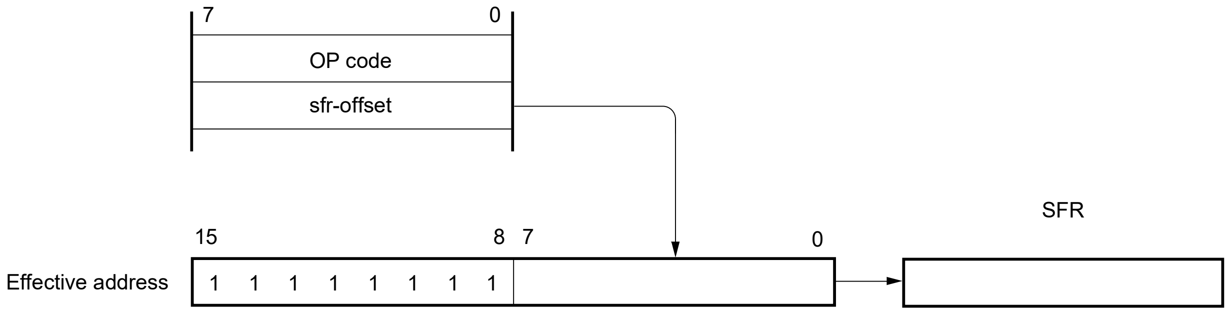
Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

[Description example]

MOV PM0, A; when selecting PM0 (FF20H) as sfr

Operation code	1 1 1 1 0 1 1 0	OP code
	0 0 1 0 0 0 0 0	20H (sfr-offset)

[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an instruction word and by a register bank select flag (RBS0 and RBS1) serve as an operand address for addressing the memory. This addressing can be carried out for all the memory spaces.

[Operand format]

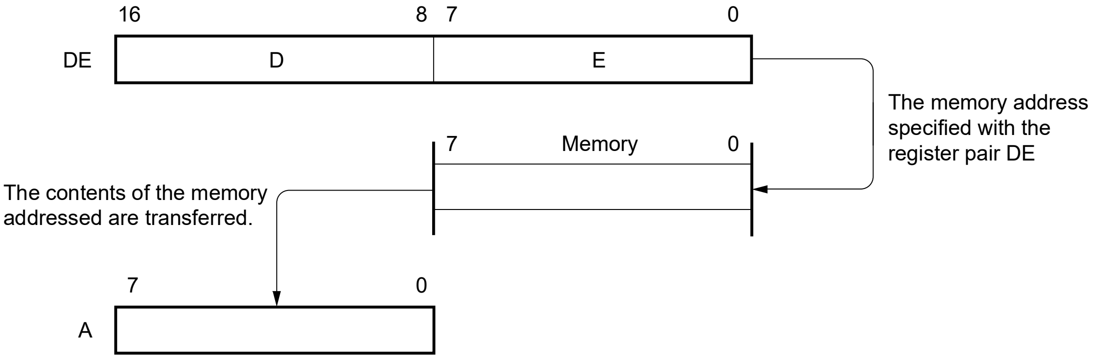
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; when selecting [DE] as register pair

Operation code 1 0 0 0 0 1 0 1

[Illustration]


16
8
7
0
DE
D
E
The memory address specified with the register pair DE
7
Memory
0
The contents of the memory addressed are transferred.
7
0
A

3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

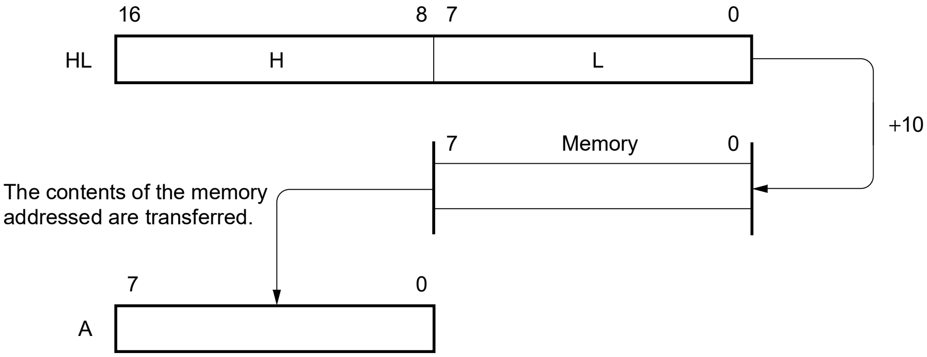
Identifier	Description
–	[HL + byte]

[Description example]

MOV A, [HL + 10H]; when setting byte to 10H

Operation code	1	0	1	0	1	1	1	0
	0	0	0	1	0	0	0	0

[Illustration]



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

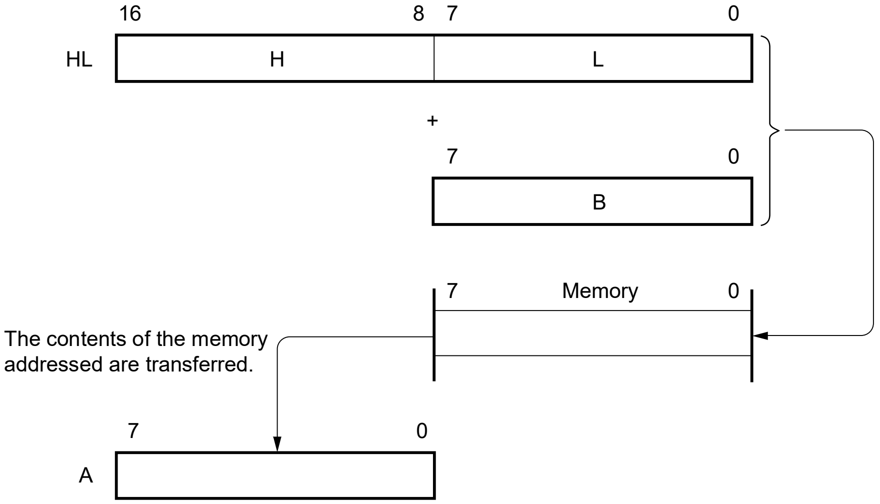
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

In the case of MOV A, [HL + B] (selecting B register)

Operation code 1 0 1 0 1 0 1 1

[Illustration]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.

This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.

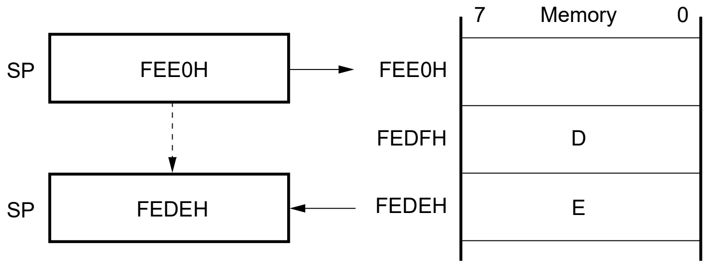
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

In the case of PUSH DE (saving DE register)

Operation code	1	0	1	1	0	1	0	1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are two types of pin I/O buffer power supplies: AV_{REF} and EV_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
EV_{DD}	Port pins other than P20 to P27

78K0/KD1+ products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, refer to **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

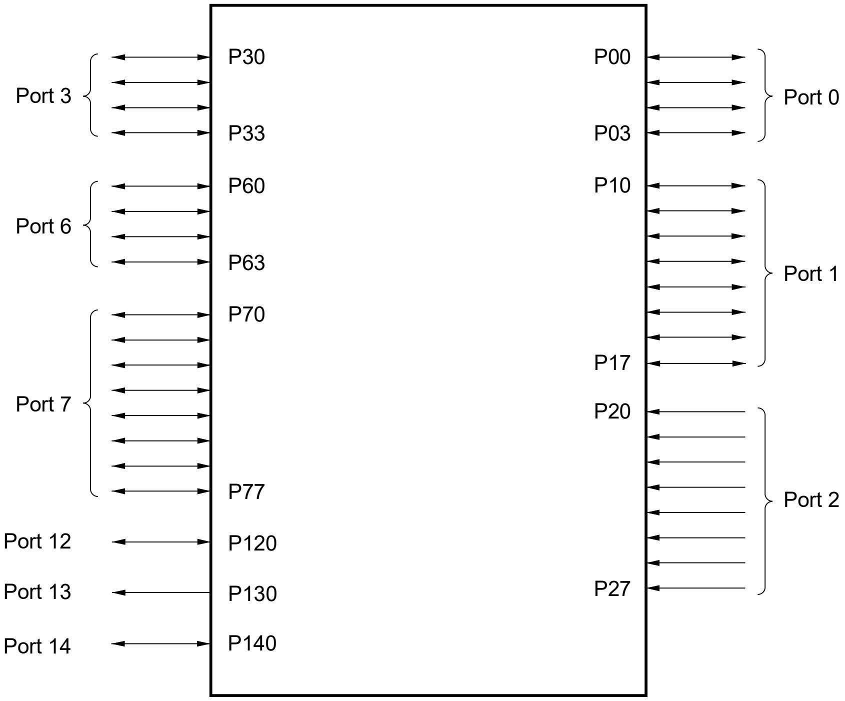


Table 4-2. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 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.	Input	TI000
P01				TI010/TO00
P02				–
P03				–
P10	I/O	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.	Input	$\overline{\text{SCK10}}$/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50/FLMD1
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30 to P32	I/O	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.	Input	INTP1 to INTP3
P33				INTP4/TI51/TO51
P60 to P63	I/O	Port 6. 4-bit I/O port (N-ch open drain). Input/output can be specified in 1-bit units.	Input	–
P70 to P77	I/O	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.	Input	KR0 to KR7
P120	I/O	Port 12. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0
P130	Output	Port 13. 1-bit output-only port.	Output	–
P140	I/O	Port 14. 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.	Input	PCL/INTP6

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode register (PM0, PM1, PM3, PM6, PM7, PM12, PM14) Port register (P0 to P3, P6, P7, P12 to P14) Pull-up resistor option register (PU0, PU1, PU3, PU7, PU12, PU14)
Port	Total: 39 (CMOS I/O: 26, CMOS input: 8, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	Total: 26

4.2.1 Port 0

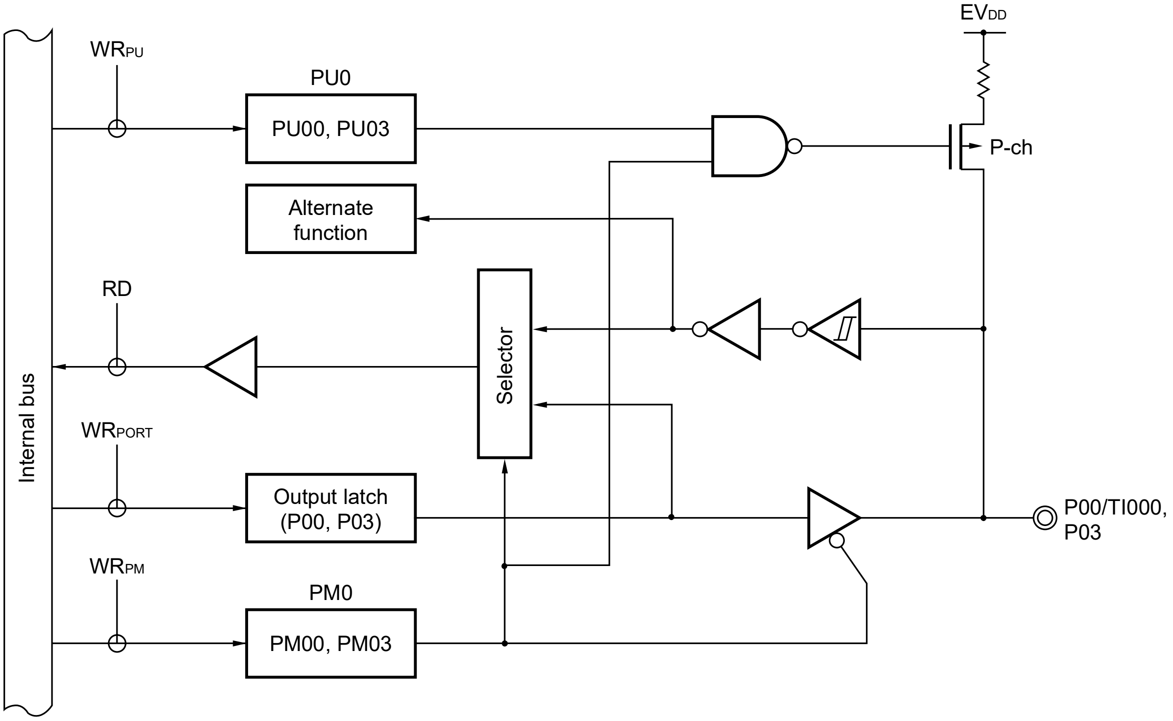
Port 0 is a 4-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

This port can also be used for timer I/O.

$\overline{\text{RESET}}$ input sets port 0 to input mode.

Figures 4-2 to 4-4 show block diagrams of port 0.

Figure 4-2. Block Diagram of P00 and P03



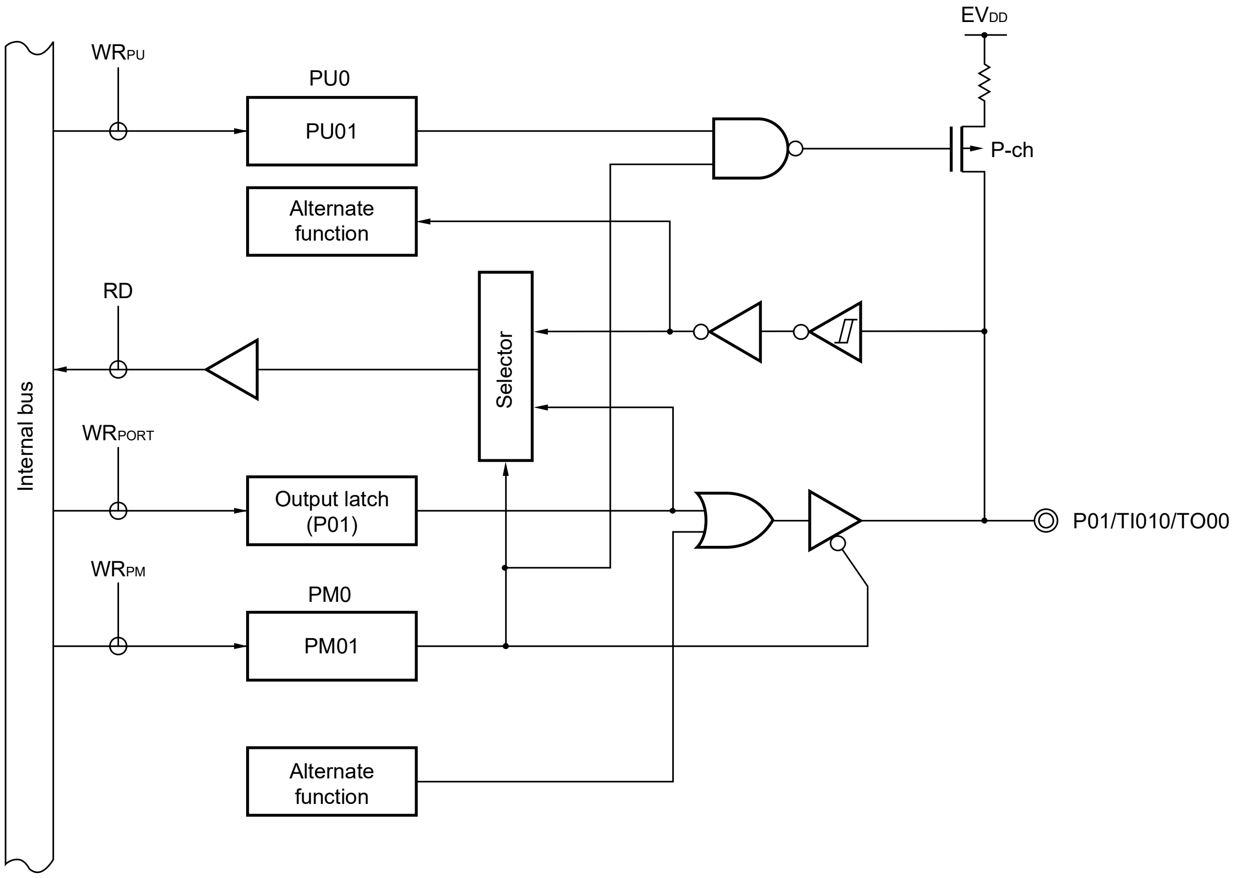
PU0: Pull-up resistor option register 0

PM0: Port mode register 0

RD: Read signal

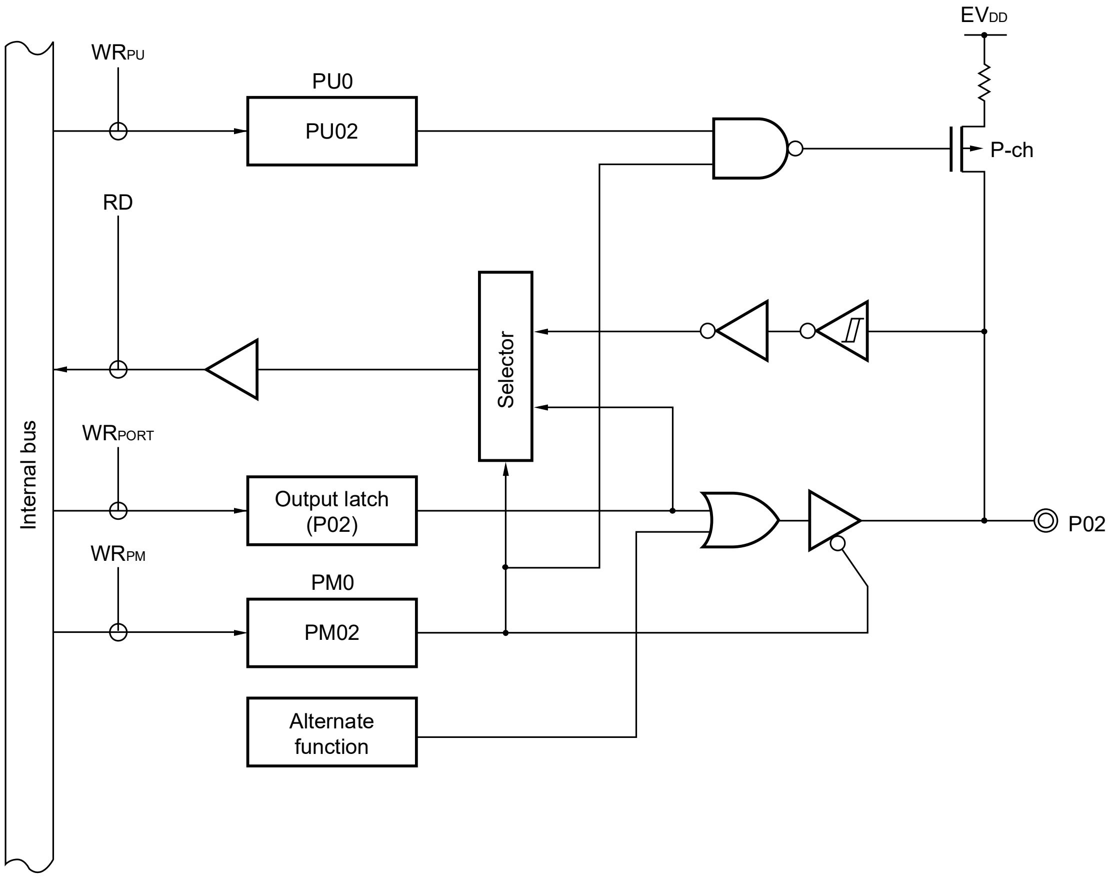
WR××: Write signal

Figure 4-3. Block Diagram of P01



PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P02



PU0:	Pull-up resistor option register 0
PM0:	Port mode register 0
RD:	Read signal
WR××:	Write signal

4.2.2 Port 1

Port 1 is an 8-bit 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 1-bit units by pull-up resistor option register 1 (PU1).

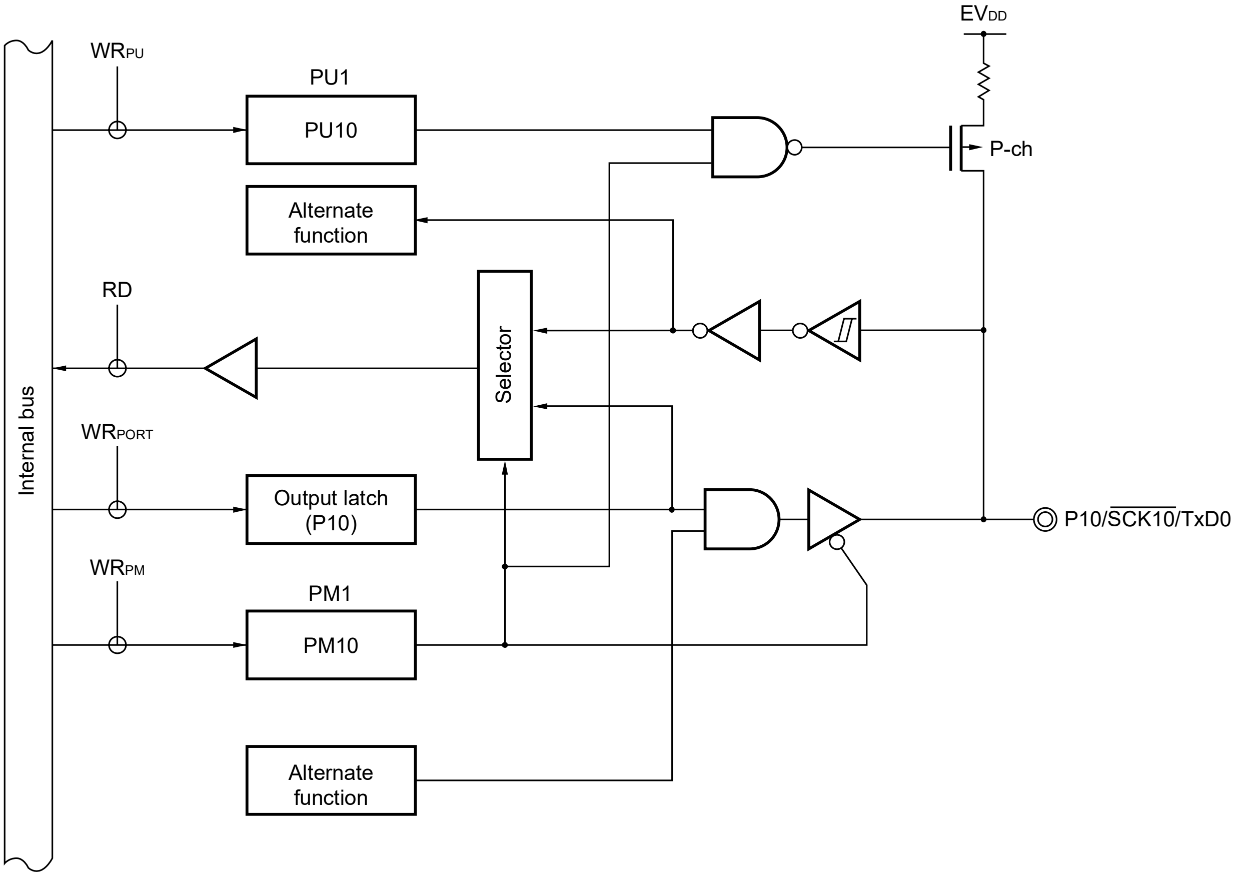
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and flash memory programming mode setting.

$\overline{\text{RESET}}$ input sets port 1 to input mode.

Figures 4-5 to 4-9 show block diagrams of port 1.

Caution When P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 are used as general-purpose ports, do not write to serial clock selection register 10 (CSIC10).

Figure 4-5. Block Diagram of P10



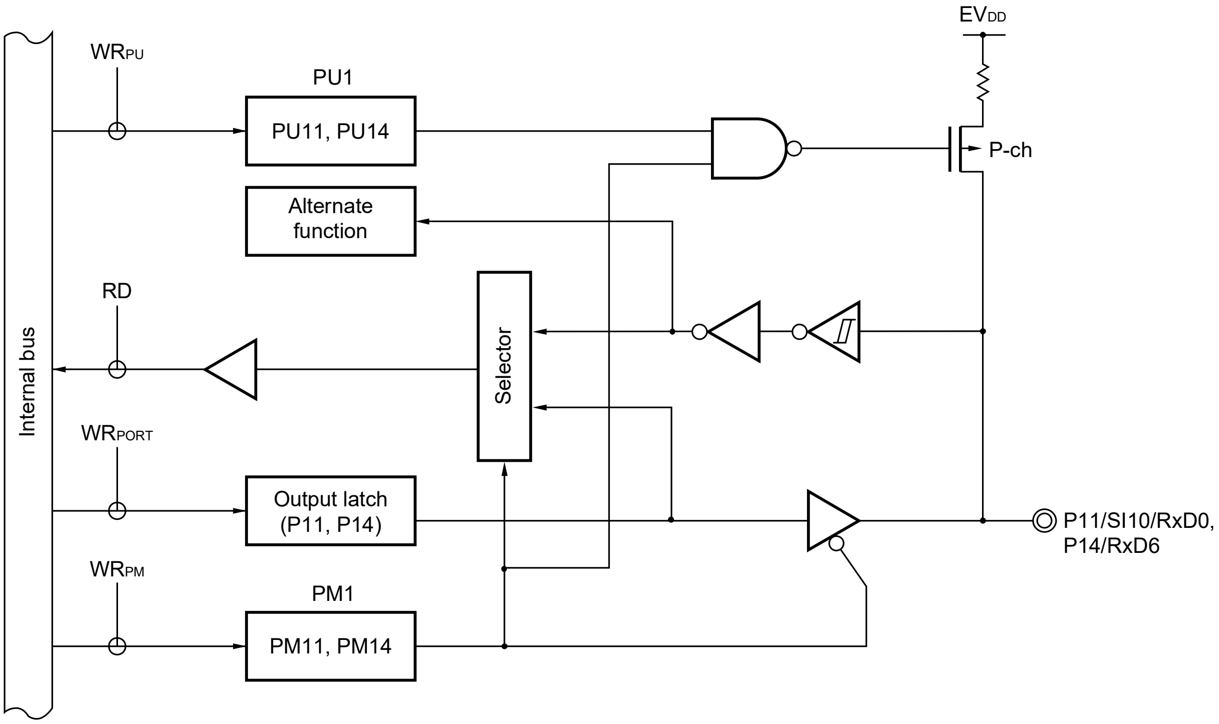
PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

WR××: Write signal

Figure 4-6. Block Diagram of P11 and P14



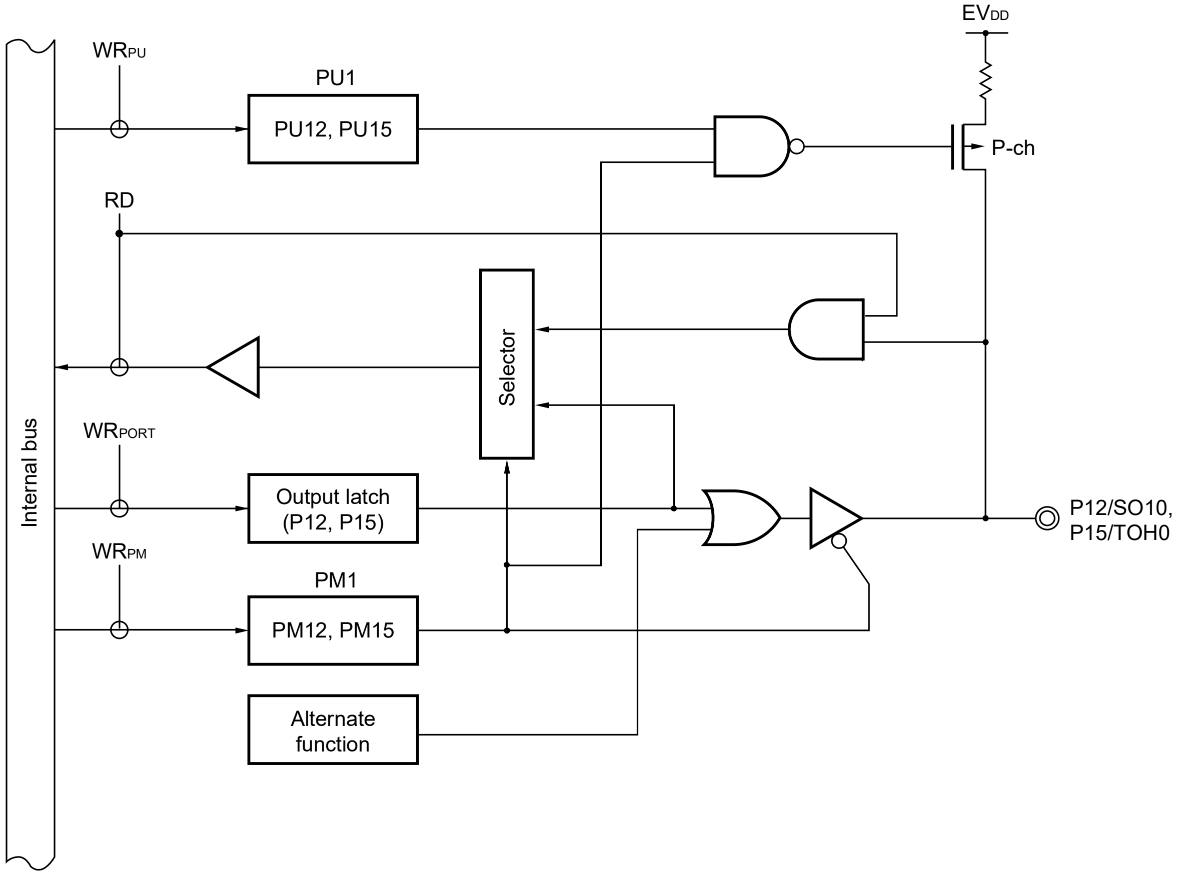
PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

WR××: Write signal

Figure 4-7. Block Diagram of P12 and P15



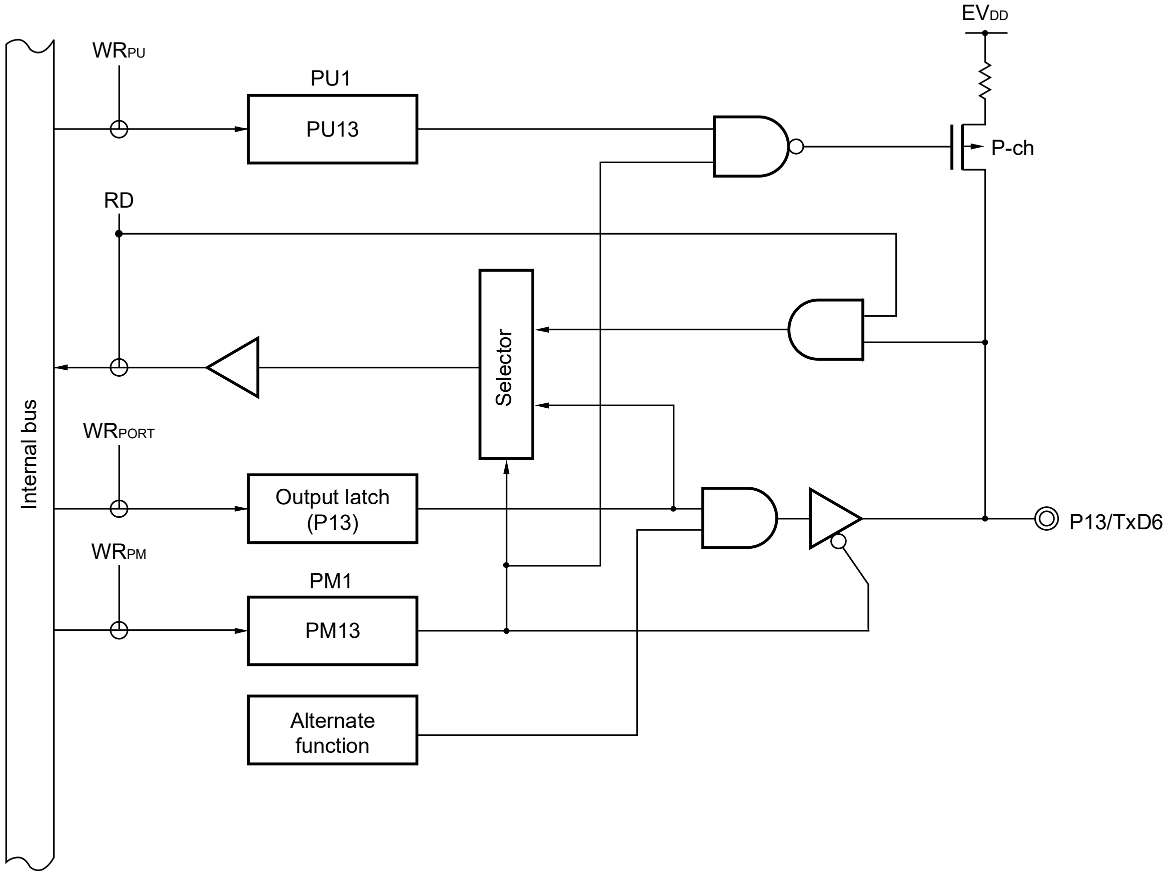
PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

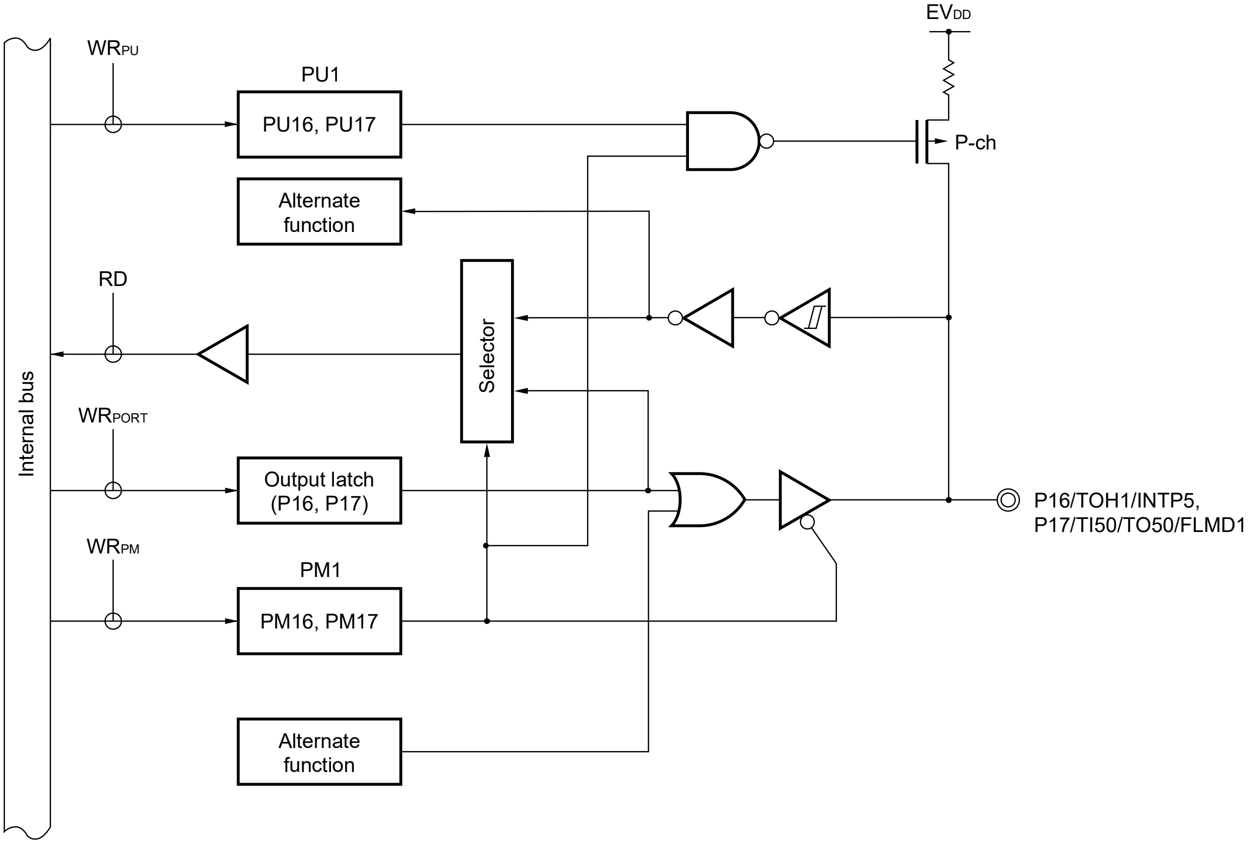
WR××: Write signal

Figure 4-8. Block Diagram of P13



PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-9. Block Diagram of P16 and P17



PU1:	Pull-up resistor option register 1
PM1:	Port mode register 1
RD:	Read signal
WR××:	Write signal

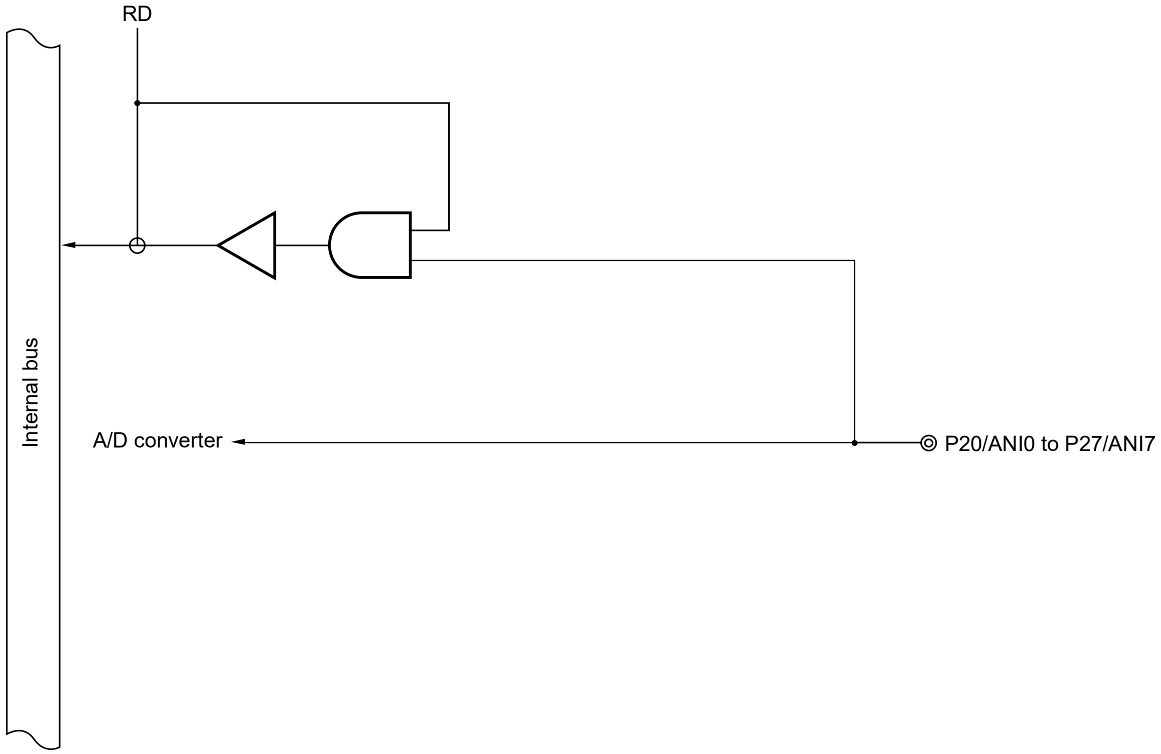
4.2.3 Port 2

Port 2 is an 8-bit input-only port.

This port can also be used for A/D converter analog input.

Figure 4-10 shows a block diagram of port 2.

Figure 4-10. Block Diagram of P20 to P27



RD: Read signal

4.2.4 Port 3

Port 3 is a 4-bit 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 used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input and timer I/O.

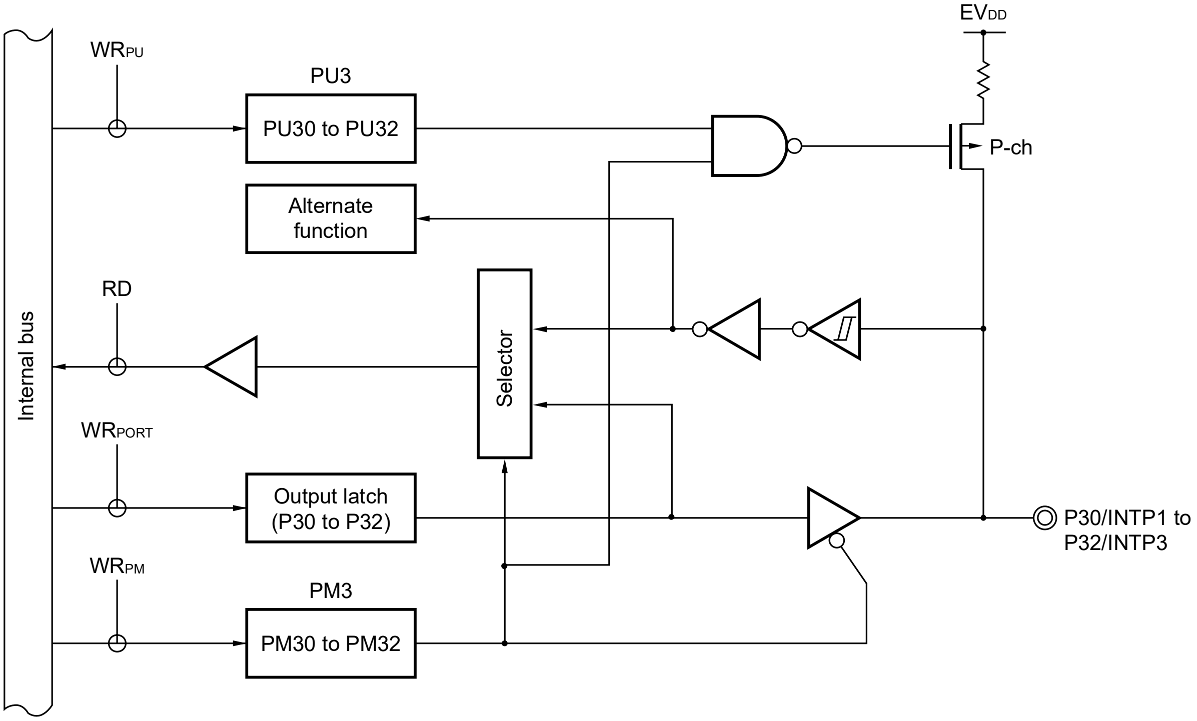
$\overline{\text{RESET}}$ input sets port 3 to input mode.

Figures 4-11 and 4-12 show block diagrams of port 3.

Caution In the μPD78F0124HD, be sure to pull the P31 pin down after reset to prevent malfunction.

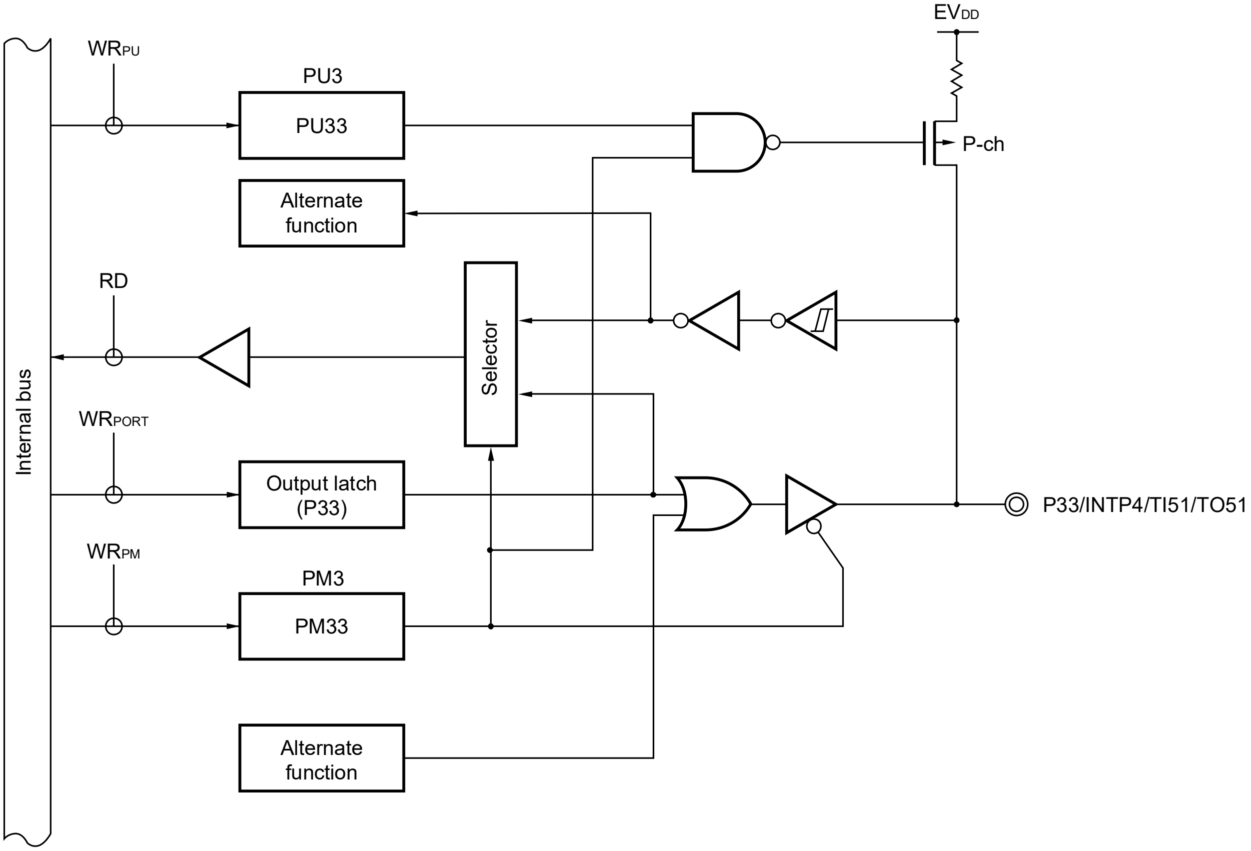
Remark P31/INTP2 and P32/INTP3 of the μPD78F0124HD can be used as on-chip debug mode setting pins when the on-chip debug function is used. For details, refer to **CHAPTER 25 ON-CHIP DEBUG FUNCTION (μPD78F0124HD ONLY)**.

Figure 4-11. Block Diagram of P30 to P32



PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

Figure 4-12. Block Diagram of P33



PU3: Pull-up resistor option register 3

PM3: Port mode register 3

RD: Read signal

WR××: Write signal

4.2.5 Port 6

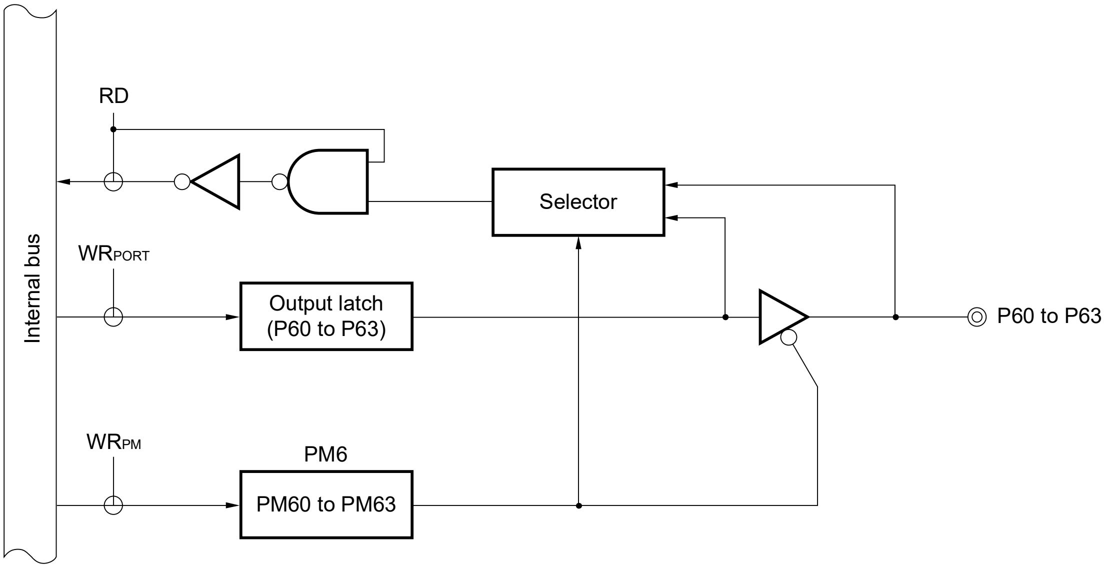
Port 6 is a 4-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The P60 to P63 pins are N-ch open-drain pins.

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figure 4-13 shows a block diagram of port 6.

Figure 4-13. Block Diagram of P60 to P63



PM6: Port mode register 6
RD: Read signal
WR××: Write signal

4.2.6 Port 7

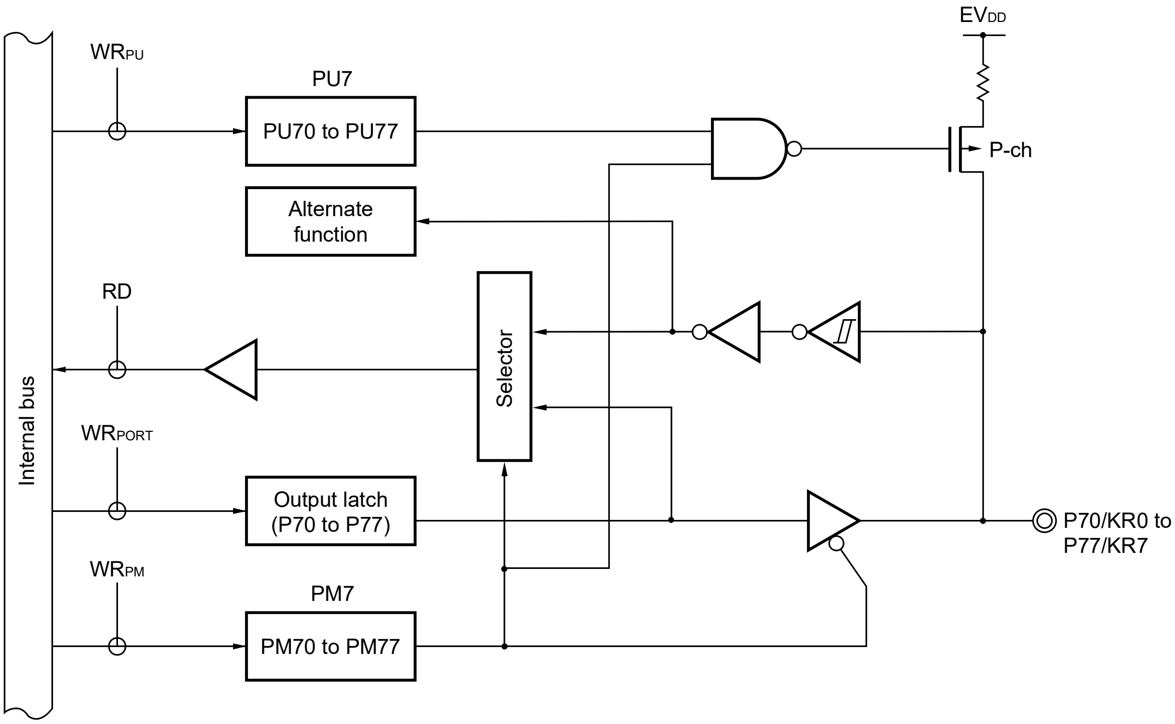
Port 7 is an 8-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for key return input.

$\overline{\text{RESET}}$ input sets port 7 to input mode.

Figure 4-14 shows a block diagram of port 7.

Figure 4-14. Block Diagram of P70 to P77



PU7:	Pull-up resistor option register 7
PM7:	Port mode register 7
RD:	Read signal
WR××:	Write signal

4.2.7 Port 12

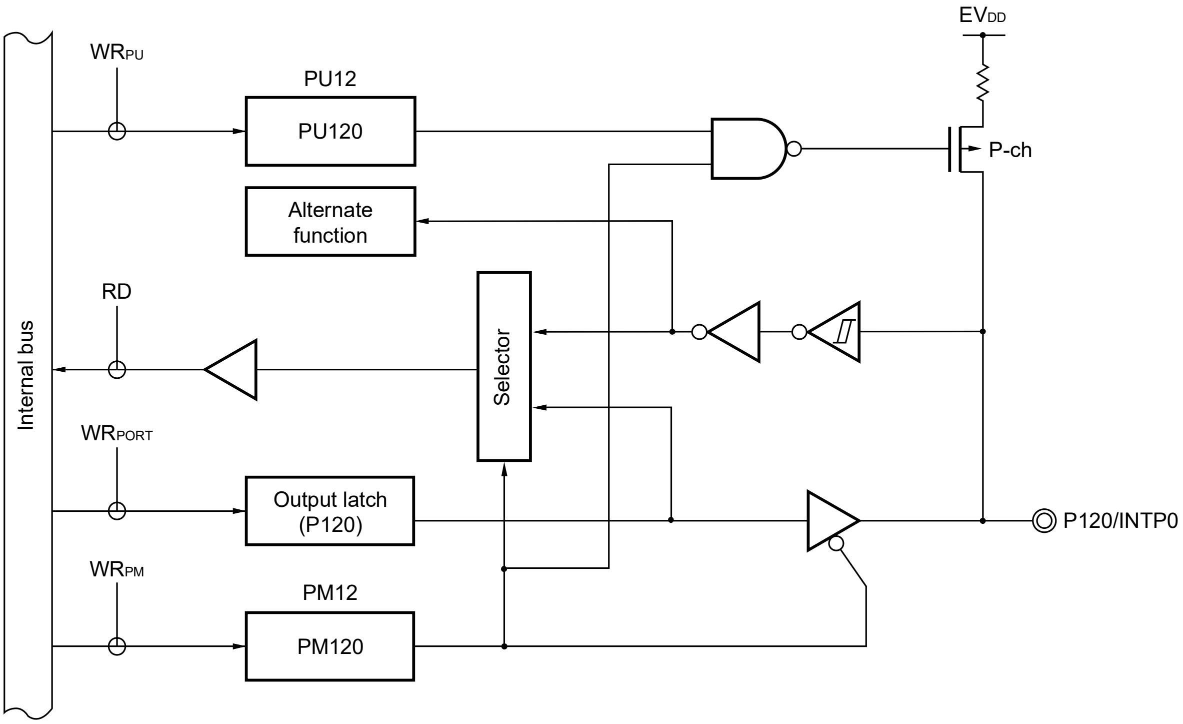
Port 12 is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

This port can also be used for external interrupt input.

$\overline{\text{RESET}}$ input sets port 12 to input mode.

Figure 4-15 shows a block diagram of port 12.

Figure 4-15. Block Diagram of P120



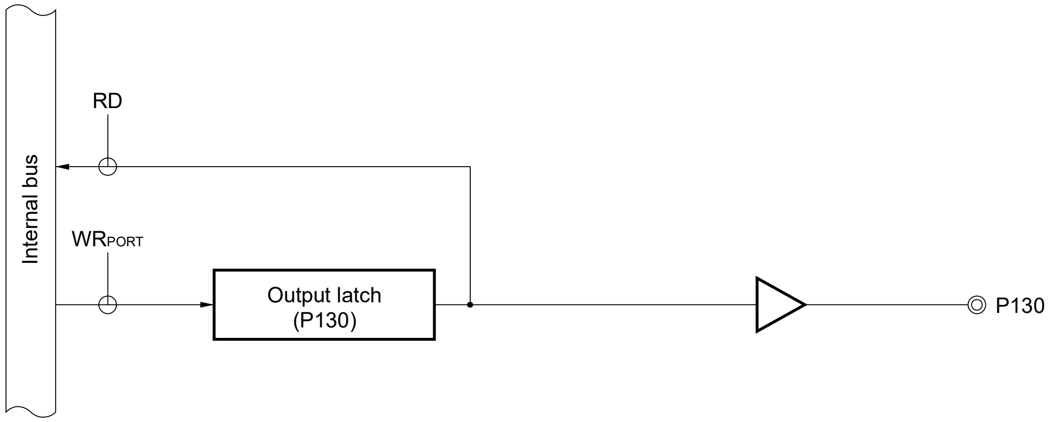
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

4.2.8 Port 13

Port 13 is a 1-bit output-only port.

Figure 4-16 shows a block diagram of port 13.

Figure 4-16. Block Diagram of P130



RD: Read signal

WD××: Write signal

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the reset signal to the CPU.

4.2.9 Port 14

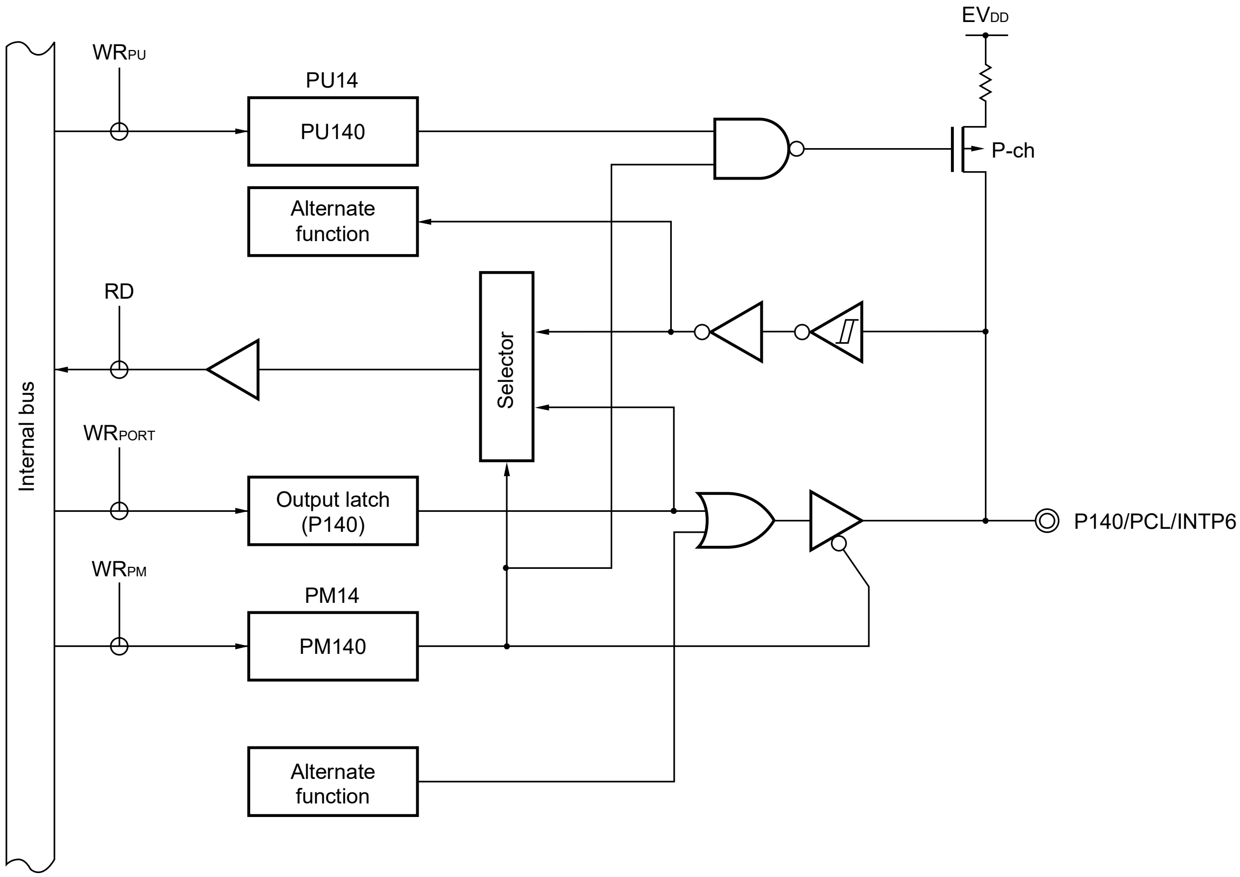
Port 14 is a 1-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode using port mode register 14 (PM14). When the P140 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for external interrupt request input and clock output.

$\overline{\text{RESET}}$ input sets port 14 to input mode.

Figure 4-17 shows a block diagram of port 14.

Figure 4-17. Block Diagram of P140



PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following three types of registers.

- Port mode registers (PM0, PM1, PM3, PM6, PM7, PM12, PM14)
- Port registers (P0 to P3, P6, P7, P12 to P14)
- Pull-up resistor option registers (PU0, PU1, PU3, PU7, PU12, PU14)

(1) Port mode registers (PM0, PM1, PM3, PM6, PM7, PM12, and PM14)

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.

$\overline{\text{RESET}}$ input sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register and output latch as shown in Table 4-4.

Figure 4-18. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FF26H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W

	7	6	5	4	3	2	1	0			
PM12	1	1	1	1	1	1	1	PM120	FF2CH	FFH	R/W

	7	6	5	4	3	2	1	0			
PM14	1	1	1	1	1	1	1	PM140	FF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 6, 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Table 4-4. Settings of Port Mode Register and Output Latch When Using Alternate Function

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI000	Input	1	×
P01	TI010	Input	1	×
	TO00	Output	0	0
P10	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
	TxD0	Output	0	1
P11	SI10	Input	1	×
	RxD0	Input	1	×
P12	SO10	Output	0	0
P13	TxD6	Output	0	1
P14	RxD6	Input	1	×
P15	TOH0	Output	0	0
P16	TOH1	Output	0	0
	INTP5	Input	1	×
P17	TI50	Input	1	×
	TO50	Output	0	0
P30 to P32	INTP1 to INTP3	Input	1	×
P33	INTP4	Input	1	×
	TI51	Input	1	×
	TO51	Output	0	0
P70 to P77	KR0 to KR7	Input	1	×
P120	INTP0	Input	1	×
P140	PCL	Output	0	0
	INTP6	Input	1	×

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

(2) Port registers (P0 to P3, P6, P7, P12 to P14)

These registers write the data that is output from the chip when data is output from 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 value of the output latch is read.

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

$\overline{\text{RESET}}$ input clears these registers to 00H (but P2 is undefined).

Figure 4-19. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	P03	P02	P01	P00	FF00H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	Undefined	R
	7	6	5	4	3	2	1	0			
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P6	0	0	0	0	P63	P62	P61	P60	FF06H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P7	P77	P76	P75	P74	P73	P72	P71	P70	FF07H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P12	0	0	0	0	0	0	0	P120	FF0CH	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P13	0	0	0	0	0	0	0	P130	FF0DH	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P14	0	0	0	0	0	0	0	P140	FF0EH	00H (output latch)	R/W

Pmn	m = 0 to 3, 6, 7, 12 to 14; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

(3) Pull-up resistor option registers (PU0, PU1, PU3, PU7, PU12, and PU14)

These registers specify whether the on-chip pull-up resistors of P00 to P03, P10 to P17, P30 to P33, P70 to P77, P120, or P140 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in PU0, PU1, PU3, PU7, PU12, and PU14. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, PU3, PU7, PU12, and PU14.

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

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 4-20. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	FF30H	00H	R/W
	7	6	5	4	3	2	1	0			
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
	7	6	5	4	3	2	1	0			
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
	7	6	5	4	3	2	1	0			
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
	7	6	5	4	3	2	1	0			
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W
	7	6	5	4	3	2	1	0			
PU14	0	0	0	0	0	0	0	PU140	FF3EH	00H	R/W

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

4.4 Port Function Operations

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

Caution In the case of a 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed as an 8-bit unit. Therefore, on a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined, even for bits other than the manipulated bit.

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 by reset.

(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.

Once data is written to the output latch, it is retained until data is written to the output latch again.

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 by reset.

(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.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three system clock oscillators are available.

- High-speed system clock oscillator
  The high-speed system clock oscillator oscillates a clock of f_{XP} = 2.0 to 16.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the main OSC control register (MOC) and processor clock control register (PCC).
- Ring-OSC oscillator
  The Ring-OSC oscillator oscillates a clock of f_R = 240 kHz (TYP.). Oscillation can be stopped by setting the Ring-OSC mode register (RCM) when “Can be stopped by software” is set by the option byte and the high-speed system clock is used as the CPU clock.
- Subsystem clock oscillator
  The subsystem clock oscillator oscillates a clock of f_{XT} = 32.768 kHz. Oscillation cannot be stopped. When subsystem clock oscillator is not used, setting not to use the on-chip feedback resistor is possible using the processor clock control register (PCC), and the operating current can be reduced in the STOP mode.

Remarks 1. f_{XP}: High-speed system clock oscillation frequency
2. f_R: Ring-OSC clock oscillation frequency
3. f_{XT}: Subsystem clock oscillation frequency

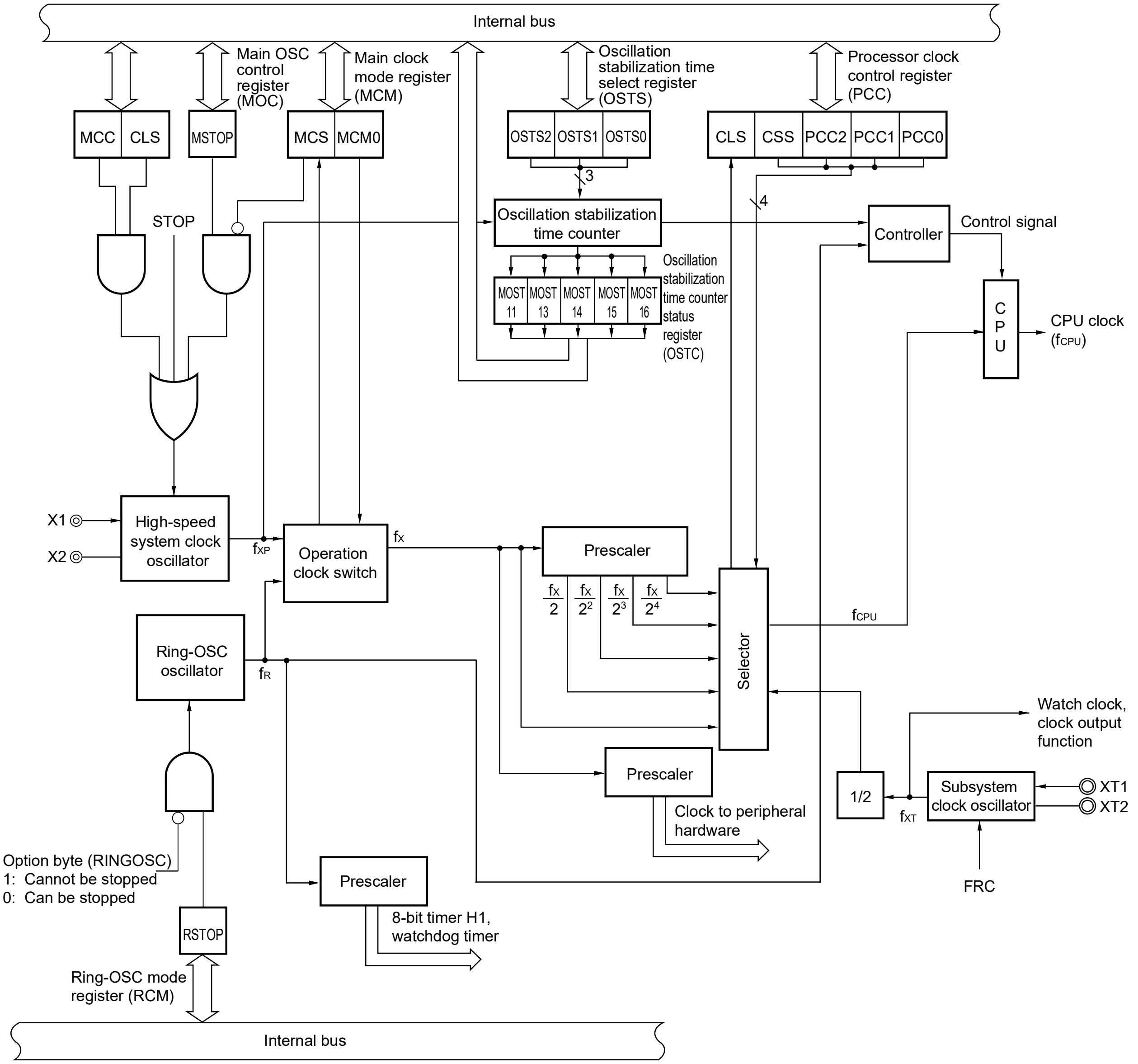
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Ring-OSC mode register (RCM) Main clock mode register (MCM) Main OSC control register (MOC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System wait control register (VSWC)
Oscillators	High-speed system clock oscillator Ring-OSC oscillator Subsystem clock oscillator

Figure 5-1. Block Diagram of Clock Generator


Internal bus
Main OSC control register (MOC)
Main clock mode register (MCM)
Oscillation stabilization time select register (OSTS)
Processor clock control register (PCC)
MCC
CLS
MSTOP
MCS
MCM0
OSTS2
OSTS1
OSTS0
CLS
CSS
PCC2
PCC1
PCC0
3
4
STOP
Oscillation stabilization time counter
Controller
Control signal
MOST 11
MOST 13
MOST 14
MOST 15
MOST 16
Oscillation stabilization time counter status register (OSTC)
C P U
CPU clock (f_{CPU})
X1
X2
High-speed system clock oscillator
f_{XP}
Operation clock switch
f_X
Prescaler
$\frac{f_X}{2}$
$\frac{f_X}{2^2}$
$\frac{f_X}{2^3}$
$\frac{f_X}{2^4}$
Selector
f_{CPU}
Ring-OSC oscillator
f_R
Watch clock, clock output function
Prescaler
Clock to peripheral hardware
1/2
f_{XT}
Subsystem clock oscillator
XT1
XT2
FRC
Option byte (RINGOSC)
1: Cannot be stopped
0: Can be stopped
Prescaler
8-bit timer H1, watchdog timer
RSTOP
Ring-OSC mode register (RCM)
Internal bus

5.3 Registers Controlling Clock Generator

The following seven registers are used to control the clock generator.

- Processor clock control register (PCC)
- Ring-OSC mode register (RCM)
- Main clock mode register (MCM)
- Main OSC control register (MOC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System wait control register (VSWC)

(1) Processor clock control register (PCC)

The PCC register is used to select the CPU clock, the division ratio, main system clock oscillator operation/stop and whether to use the on-chip feedback resistor[Note] of the subsystem clock oscillator.
The PCC can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears PCC to 00H.

Note The feedback resistor is required to control the bias point of the oscillation waveform so that the bias point is in the middle of the power supply voltage (refer to **Figure 5-12 Subsystem Clock Feedback Resistor**).

Figure 5-2. Format of Processor Clock Control Register (PCC)

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

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
PCC	MCC	FRC	CLS	CSS	0	PCC2	PCC1	PCC0

MCC	Control of high-speed system clock oscillator operation[Note 2]
0	Oscillation possible
1	Oscillation stopped

FRC	Subsystem clock feedback resistor selection[Note 3]
0	On-chip feedback resistor used
1	On-chip feedback resistor not used

CLS	CPU clock status
0	High-speed system clock or Ring-OSC clock
1	Subsystem clock

CSS[Note 4]	PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection		
					MCM0 = 0	MCM0 = 1
0	0	0	0	f_X	f_R	f_{XP}
	0	0	1	$f_X/2$	$f_R/2$	$f_{XP}/2$
	0	1	0	$f_X/2^2$	Setting prohibited	$f_{XP}/2^2$
	0	1	1	$f_X/2^3$	Setting prohibited	$f_{XP}/2^3$
	1	0	0	$f_X/2^4$	Setting prohibited	$f_{XP}/2^4$
1	0	0	0	$f_{XT}/2$		
	0	0	1			
	0	1	0			
	0	1	1			
	1	0	0			
Other than above				Setting prohibited		

Notes 1. Bit 5 is read-only.

2. When the CPU is operating on the subsystem clock, MCC should be used to stop the high-speed system clock oscillator operation. When the CPU is operating on the Ring-OSC clock, use bit 7 (MSTOP) of the main OSC control register (MOC) to stop the high-speed system clock oscillator operation (this cannot be set by MCC). A STOP instruction should not be used.

3. Clear this bit to 0 when the subsystem clock is used, and set it to 1 when the subsystem clock is not used.

4. Be sure to switch CSS from 1 to 0 when bits 1 (MCS) and 0 (MCM0) of the main clock mode register (MCM) are 1.

Caution Be sure to clear bit 3 to 0.

Remarks 1. MCM0: Bit 0 of the main clock mode register (MCM)

2. f_X: Main system clock oscillation frequency (high-speed system clock oscillation frequency or Ring-OSC clock oscillation frequency)

3. f_R: Ring-OSC clock oscillation frequency

4. f_{XP}: High-speed system clock oscillation frequency

5. f_{XT}: Subsystem clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the 78K0/KD1+. Therefore, the relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in the Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: 2/f_{CPU}			
	High-Speed System Clock[Note]		Ring-OSC Clock[Note]	Subsystem Clock
	At 10 MHz Operation	At 16 MHz Operation	At 240 kHz (TYP.) Operation	At 32.768 kHz Operation
f_X	0.2 µs	0.125 µs	8.3 µs (TYP.)	–
$f_X/2$	0.4 µs	0.25 µs	16.6 µs (TYP.)	–
$f_X/2^2$	0.8 µs	0.5 µs	Setting prohibited	–
$f_X/2^3$	1.6 µs	1.0 µs	Setting prohibited	–
$f_X/2^4$	3.2 µs	2.0 µs	Setting prohibited	–
$f_{XT}/2$	–		–	122.1 µs

Note The main clock mode register (MCM) is used to set the CPU clock (high-speed system clock/Ring-OSC clock) (refer to **Figure 5-4**).

(2) Ring-OSC mode register (RCM)

This register sets the operation mode of Ring-OSC.

This register is valid when "Can be stopped by software" is set for Ring-OSC by the option byte, and the high-speed system clock or subsystem clock is selected as the CPU clock. If "Cannot be stopped" is selected for Ring-OSC by the option byte, settings for this register are invalid.

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

$\overline{\text{RESET}}$ input clears RCM to 00H.

Figure 5-3. Format of Ring-OSC Mode Register (RCM)

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	Ring-OSC oscillating/stopped
0	Ring-OSC oscillating
1	Ring-OSC stopped

Caution Make sure that the bit 1 (MCS) of the main clock mode register (MCM) is 1 before setting RSTOP.

(3) Main clock mode register (MCM)

This register sets the CPU clock (high-speed system clock/Ring-OSC clock).

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

$\overline{\text{RESET}}$ input clears MCM to 00H.

Figure 5-4. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	<1>	<0>
MCM	0	0	0	0	0	0	MCS	MCM0

MCS	CPU clock status
0	Operates with Ring-OSC clock
1	Operates with high-speed system clock

MCM0	Selection of clock supplied to CPU
0	Ring-OSC clock
1	High-speed system clock

Note Bit 1 is read-only.

Cautions 1. When Ring-OSC clock is selected as the clock to be supplied to the CPU, the divided clock of the Ring-OSC oscillator output (f_X) is supplied to the peripheral hardware (f_X = 240 kHz (TYP.)).

Operation of the peripheral hardware with Ring-OSC clock cannot be guaranteed. Therefore, when Ring-OSC clock is selected as the clock supplied to the CPU, do not use peripheral hardware. In addition, stop the peripheral hardware before switching the clock supplied to the CPU from the high-speed system clock to the Ring-OSC clock. Note, however, that the following peripheral hardware can be used when the CPU operates on the Ring-OSC clock.

- **Watchdog timer**
- **Clock monitor**
- **8-bit timer H1 when $f_R/2^7$ is selected as count clock**
- **Peripheral hardware selecting external clock as the clock source (Except when external count clock of TM0n (n = 0, 1) is selected (TI00n valid edge))**

2. Always switch subsystem clock operation to high-speed system clock operation (bit 4 (CSS) of the processor clock control register (PCC) is changed from 1 to 0) with MCS = 1 and MCM0 = 1.

(4) Main OSC control register (MOC)

This register selects the operation mode of the high-speed system clock.

This register is used to stop the high-speed system clock oscillator operation when the CPU is operating with the Ring-OSC clock. Therefore, this register is valid only when the CPU is operating with the Ring-OSC clock.

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

$\overline{\text{RESET}}$ input clears MOC to 00H.

Figure 5-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of high-speed system clock oscillator operation
0	High-speed system clock oscillator operating
1	High-speed system clock oscillator stopped

Cautions 1. Make sure that bit 1 (MCS) of the main clock mode register (MCM) is 1 before setting MSTOP.

2. To stop high-speed system clock oscillation when the CPU is operating on the subsystem clock, set bit 7 (MCC) of the processor clock control register (PCC) to 1 (setting by MSTOP is not possible).

(5) Oscillation stabilization time counter status register (OSTC)

This is the status register of the high-speed system clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the high-speed system clock oscillation stabilization time can be checked.

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

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT), the STOP instruction, MSTOP = 1, and MCC = 1 clear OSTC to 00H.

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

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_{XP} = 10 MHz	f_{XP} = 16 MHz
1	0	0	0	0	$2^{11}/f_{XP}$ min.	204.8 μs min.	128 μs min.
1	1	0	0	0	$2^{13}/f_{XP}$ min.	819.2 μs min.	512 μs min.
1	1	1	0	0	$2^{14}/f_{XP}$ min.	1.64 ms min.	1.02 ms min.
1	1	1	1	0	$2^{15}/f_{XP}$ min.	3.27 ms min.	2.04 ms min.
1	1	1	1	1	$2^{16}/f_{XP}$ min.	6.55 ms min.	4.09 ms min.

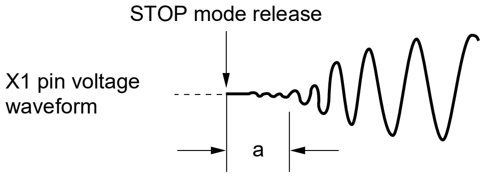
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the Ring-OSC is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

3. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts ("a" below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: High-speed system clock oscillation frequency

(6) Oscillation stabilization time select register (OSTS)

This register is used to select the high-speed system clock oscillation stabilization wait time when STOP mode is released.

The wait time set by OSTS is valid only after STOP mode is released with the high-speed system clock selected as CPU clock. After STOP mode is released with Ring-OSC selected as CPU clock, the oscillation stabilization time must be confirmed by OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 05H.

Figure 5-7. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

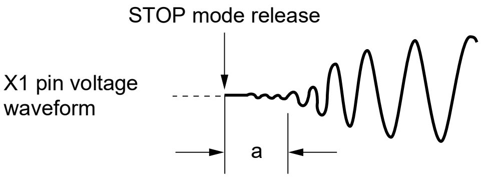
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_{XP} = 10 MHz	f_{XP} = 16 MHz
0	0	1	$2^{11}/f_{XP}$	204.8 μs	128 μs
0	1	0	$2^{13}/f_{XP}$	819.2 μs	512 μs
0	1	1	$2^{14}/f_{XP}$	1.64 ms	1.02 ms
1	0	0	$2^{15}/f_{XP}$	3.27 ms	2.04 ms
1	0	1	$2^{16}/f_{XP}$	6.55 ms	4.09 ms
Other than above			Setting prohibited		

Cautions 1. If the STOP mode is entered and then released while the Ring-OSC is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

2. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts ("a" below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: High-speed system clock oscillation frequency

(7) System wait control register (VSWC)

This register is used to control wait states when a high-speed CPU and a low-speed peripheral I/O are connected.

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

$\overline{\text{RESET}}$ input clears VSWC to 00H.

Figure 5-8. Format of System Wait Control Register (VSWC)

Address: FFFDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
VSWC	0	0	0	PAW0	0	0	0	PDW0

PAW0	Control of system clock address wait
0	No wait
1	One wait state inserted

PDW0	Control of system clock data wait
0	No wait
1	One wait state inserted

Cautions 1. Be sure to insert one wait state if the minimum instruction execution time is 0.2 μs or less (f_{XP} = 10 MHz or more).

2. Do not access VSWC in the current IE environment (IE-78K0-NS, IE-78K0-NS-A, and IE-78K0K1-ET).

3. Be sure to clear bits 1 to 3 and 5 to 7 to 0.

5.4 System Clock Oscillator

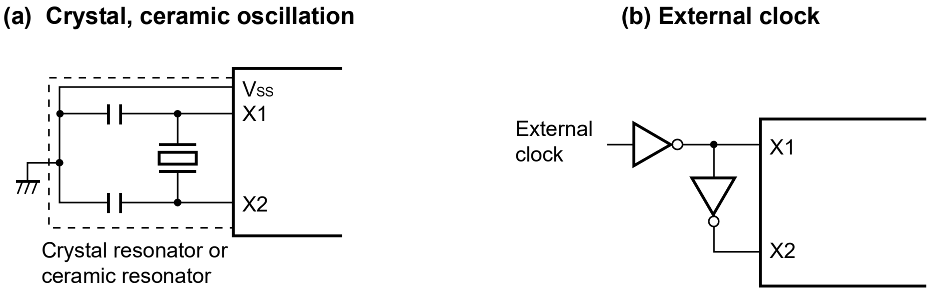
5.4.1 High-speed system clock oscillator

The high-speed system clock oscillator oscillates with a crystal resonator or ceramic resonator (Standard: 16 MHz) connected to the X1 and X2 pins.

An external clock can be input to the high-speed system clock oscillator. In this case, input the clock signal to the X1 pin and input the inverse signal to the X2 pin.

Figure 5-9 shows examples of the external circuit of the high-speed system clock oscillator.

Figure 5-9. Examples of External Circuit of High-Speed System Clock Oscillator



(a) Crystal, ceramic oscillation

(b) External clock

Cautions are listed on the next page.

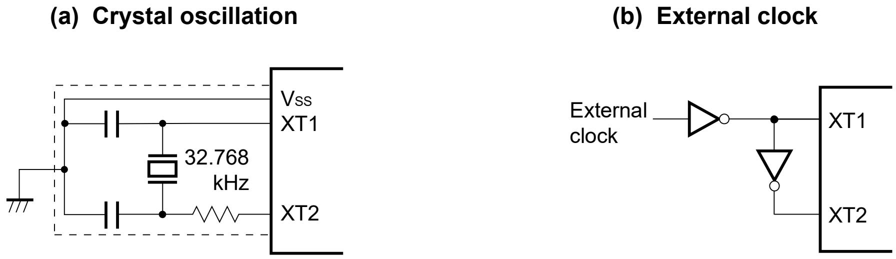
5.4.2 Subsystem clock oscillator

The subsystem clock oscillator oscillates with a crystal resonator (Standard: 32.768 kHz) connected to the XT1 and XT2 pins.

An external clock can be input to the subsystem clock oscillator. In this case, input the clock signal to the XT1 pin and the inverse signal to the XT2 pin.

Figure 5-10 shows examples of an external circuit of the subsystem clock oscillator.

Figure 5-10. Examples of External Circuit of Subsystem Clock Oscillator



(a) Crystal oscillation

(b) External clock

Cautions are listed on the next page.

Cautions 1. When using the high-speed system clock oscillator and subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-9 and 5-10 to avoid an adverse effect from wiring capacitance.

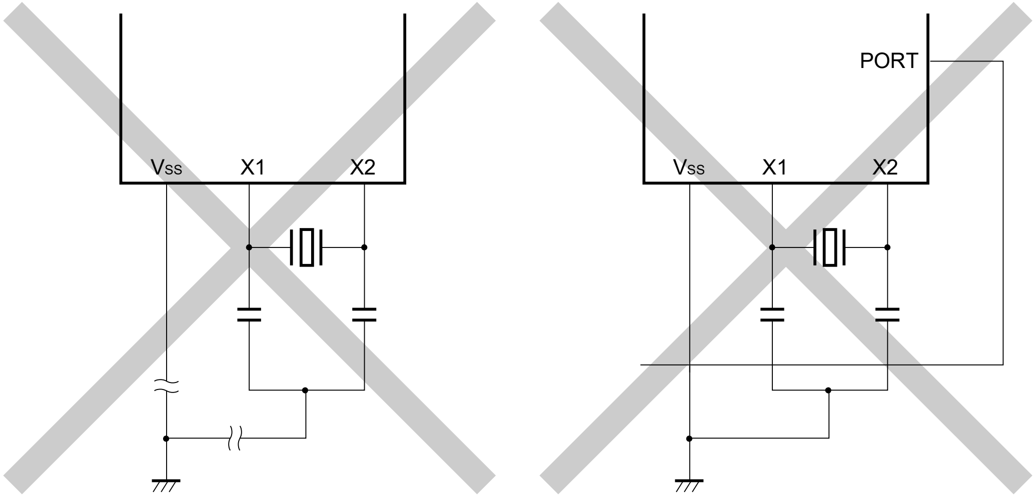
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

Note that the subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-11 shows examples of incorrect resonator connection.

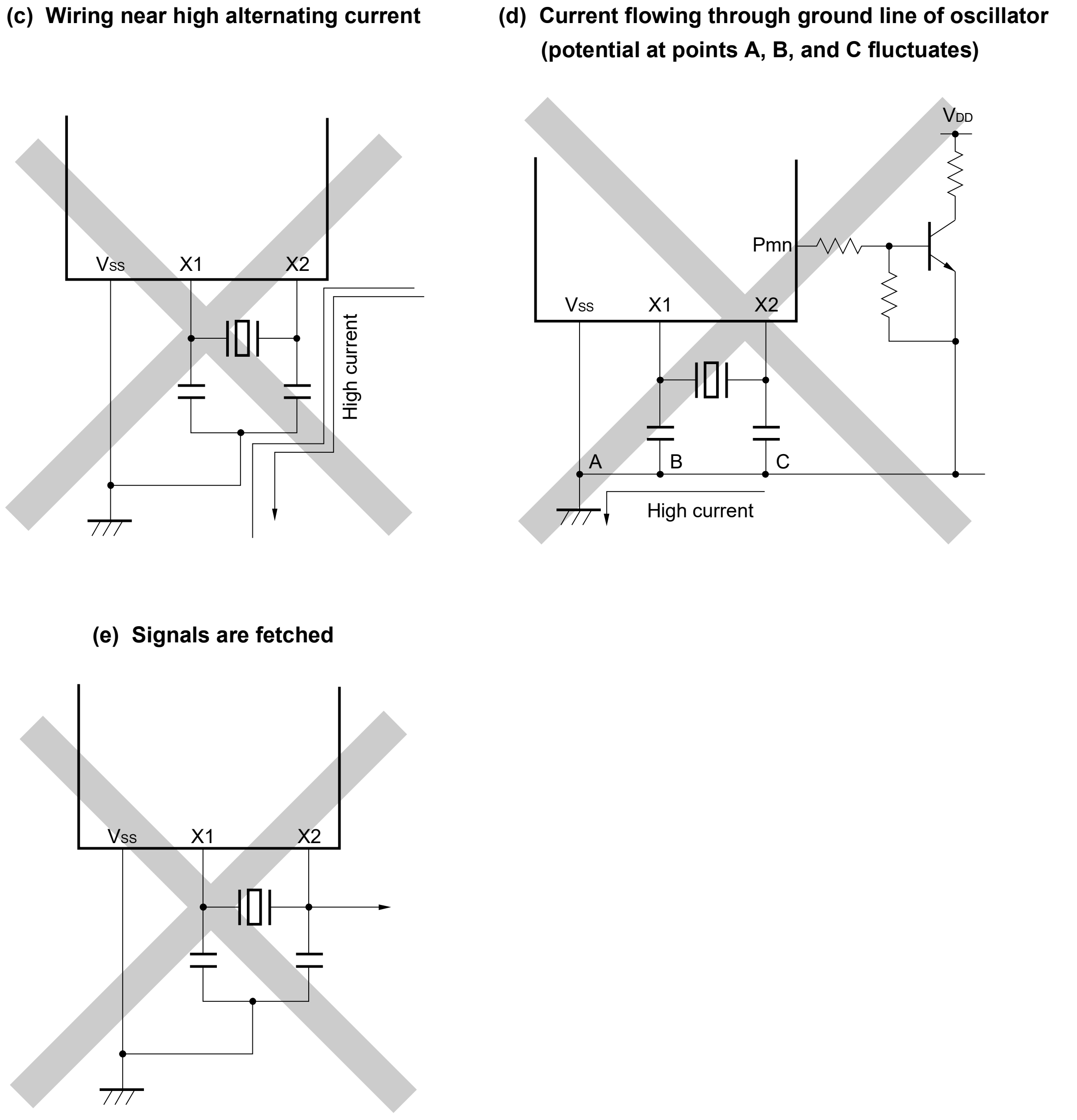
Figure 5-11. Examples of Incorrect Resonator Connection (1/2)

(a) Too long wiring **(b) Crossed signal line**



Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-11. Examples of Incorrect Resonator Connection (2/2)



(c) Wiring near high alternating current

(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)

(e) Signals are fetched

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Cautions 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

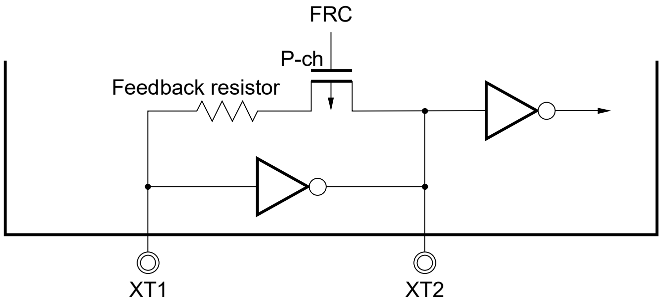
5.4.3 When subsystem clock is not used

If it is not necessary to use the subsystem clock for low power consumption operations and watch operations, connect the XT1 and XT2 pins as follows.

XT1: Connect directly to EV_{SS} or V_{SS}[Note]
XT2: Leave open

Note After reset is released, the on-chip feedback resistor must be set so that it is not used (bit 6 (FRC) of processor clock control register (PCC) = 1).

Figure 5-12. Subsystem Clock Feedback Resistor



Remark The feedback resistor is required to control the bias point of the oscillation waveform so that the bias point is in the middle of the power supply voltage.

5.4.4 Ring-OSC oscillator

Ring-OSC oscillator is incorporated in the 78K0/KD1+.

“Can be stopped by software” or “Cannot be stopped” can be selected by the option byte. The Ring-OSC clock always oscillates after $\overline{\text{RESET}}$ release (240 kHz (TYP.)).

5.4.5 Prescaler

The prescaler generates various clocks by dividing the high-speed system clock oscillator output when the high-speed system clock is selected as the clock to be supplied to the CPU.

Caution When the Ring-OSC clock is selected as the clock supplied to the CPU, the prescaler generates various clocks by dividing the Ring-OSC oscillator output (f_X = 240 kHz (TYP.)).

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode.

- High-speed system clock f_{XP}
- Ring-OSC clock f_R
- Subsystem clock f_{XT}
- CPU clock f_{CPU}
- Clock to peripheral hardware

The CPU starts operation when the on-chip Ring-OSC oscillator starts outputting after reset release in the 78K0/KD1+, thus enabling the following.

(1) Enhancement of security function

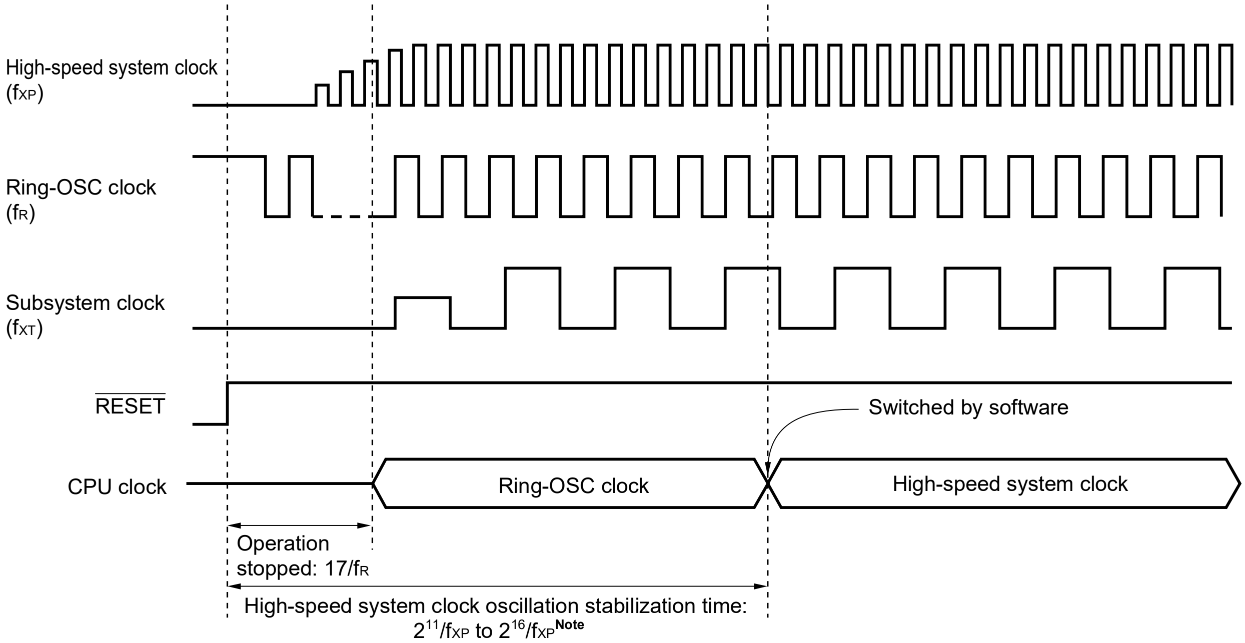
When the high-speed system clock is set as the CPU clock by the default setting, the device cannot operate if the high-speed system clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip Ring-OSC clock, so the device can be started by the Ring-OSC clock after reset release by the clock monitor (detection of high-speed system clock stop). Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(2) Improvement of performance

Because the CPU can be started without waiting for the high-speed system clock oscillation stabilization time, the total performance can be improved.

A timing diagram of the CPU default start using Ring-OSC is shown in Figure 5-13.

Figure 5-13. Timing Diagram of CPU Default Start Using Ring-OSC



Note Check using the oscillation stabilization time counter status register (OSTC).

(a) When the $\overline{\text{RESET}}$ signal is generated, bit 0 of the main clock mode register (MCM) is cleared to 0 and the Ring-OSC clock is set as the CPU clock. However, a clock is supplied to the CPU after 17 clocks of the Ring-OSC clock have elapsed after $\overline{\text{RESET}}$ release (or clock supply to the CPU stops for 17 clocks). During the $\overline{\text{RESET}}$ period, oscillation of the high-speed system clock and Ring-OSC clock is stopped.

(b) After $\overline{\text{RESET}}$ release, the CPU clock can be switched from the Ring-OSC clock to the high-speed system clock using bit 0 (MCM0) of the main clock mode register (MCM) after the high-speed system clock oscillation stabilization time has elapsed. At this time, check the oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) before switching the CPU clock. The CPU clock status can be checked using bit 1 (MCS) of MCM.

(c) Ring-OSC can be set to stopped/oscillating using the Ring-OSC mode register (RCM) when “Can be stopped by software” is selected for the Ring-OSC by the option byte, if the high-speed system or subsystem clock is used as the CPU clock. Make sure that MCS is 1 at this time.

(d) When Ring-OSC is used as the CPU clock, the high-speed system clock can be set to stopped/oscillating using the main OSC control register (MOC). Make sure that MCS is 0 at this time.

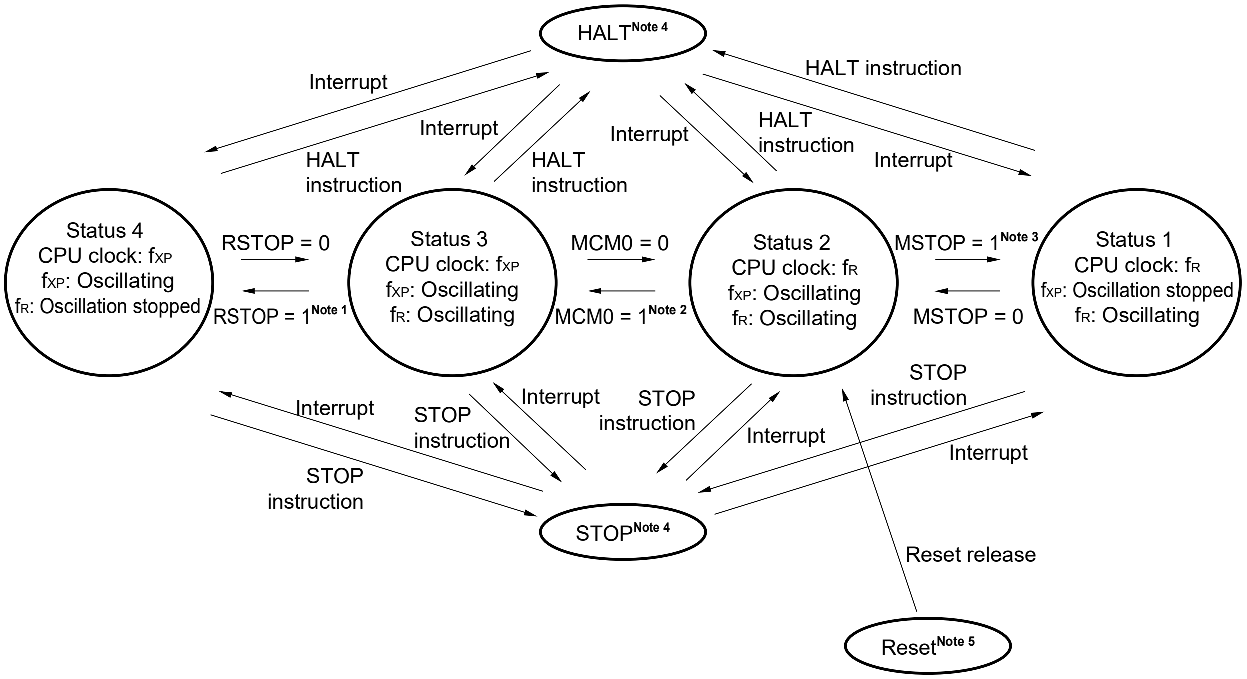
When the subsystem clock is used as the CPU clock, whether the high-speed system clock stops or oscillates can be set by the processor clock control register (PCC). In addition, HALT mode can be used during operation with the subsystem clock, but STOP mode cannot be used (subsystem clock oscillation cannot be stopped by the STOP instruction).

(e) Select the high-speed system clock oscillation stabilization time ($2^{11}/f_{XP}$, $2^{13}/f_{XP}$, $2^{14}/f_{XP}$, $2^{15}/f_{XP}$, $2^{16}/f_{XP}$) using the oscillation stabilization time select register (OSTS) when releasing STOP mode while high-speed system clock is being used as the CPU clock. In addition, when releasing STOP mode while $\overline{\text{RESET}}$ is released and Ring-OSC is being used as the CPU clock, check the high-speed system clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC).

A status transition diagram of this product is shown in Figure 5-14, and the relationship between the operation clocks in each operation status and between the oscillation control flag and oscillation status of each clock are shown in Tables 5-3 and 5-4, respectively.

Figure 5-14. Status Transition Diagram (1/4)

(1) When "Ring-OSC can be stopped by software" is selected by option byte (when subsystem clock is not used)



Notes 1. When shifting from status 3 to status 4, make sure that bit 1 (MCS) of the main clock mode register (MCM) is 1.

2. Before shifting from status 2 to status 3 after reset and STOP are released, check the high-speed system clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

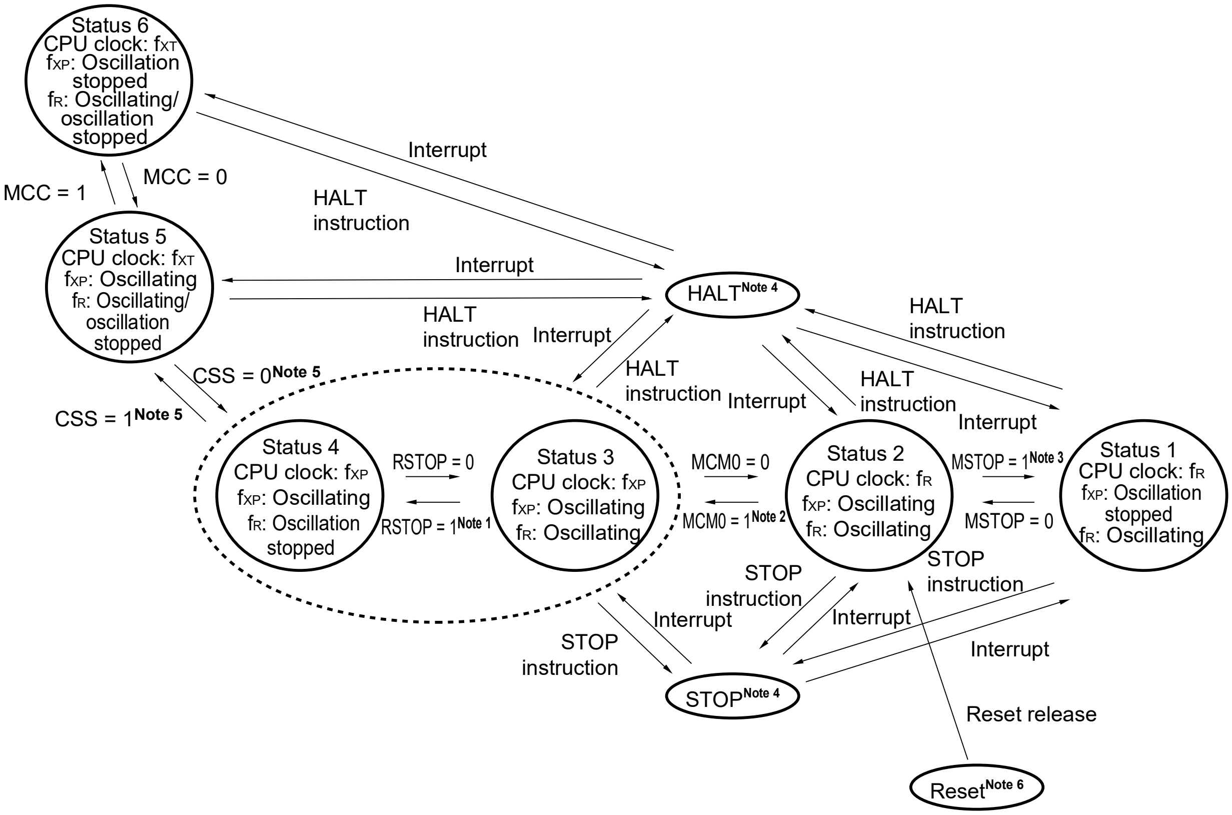
3. When shifting from status 2 to status 1, make sure that MCS is 0.

4. When "Ring-OSC can be stopped by software" is selected by the option byte, the watchdog timer stops operating in the HALT and STOP modes, regardless of the source clock of the watchdog timer. However, oscillation of Ring-OSC does not stop even in the HALT and STOP modes if RSTOP = 0.

5. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT)

Figure 5-14. Status Transition Diagram (2/4)

(2) When “Ring-OSC can be stopped by software” is selected by option byte (when subsystem clock is used)



Notes 1. When shifting from status 3 to status 4, make sure that bit 1 (MCS) of the main clock mode register (MCM) is 1.

2. Before shifting from status 2 to status 3 after reset and STOP are released, check the high-speed system clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

3. When shifting from status 2 to status 1, make sure that MCS is 0.

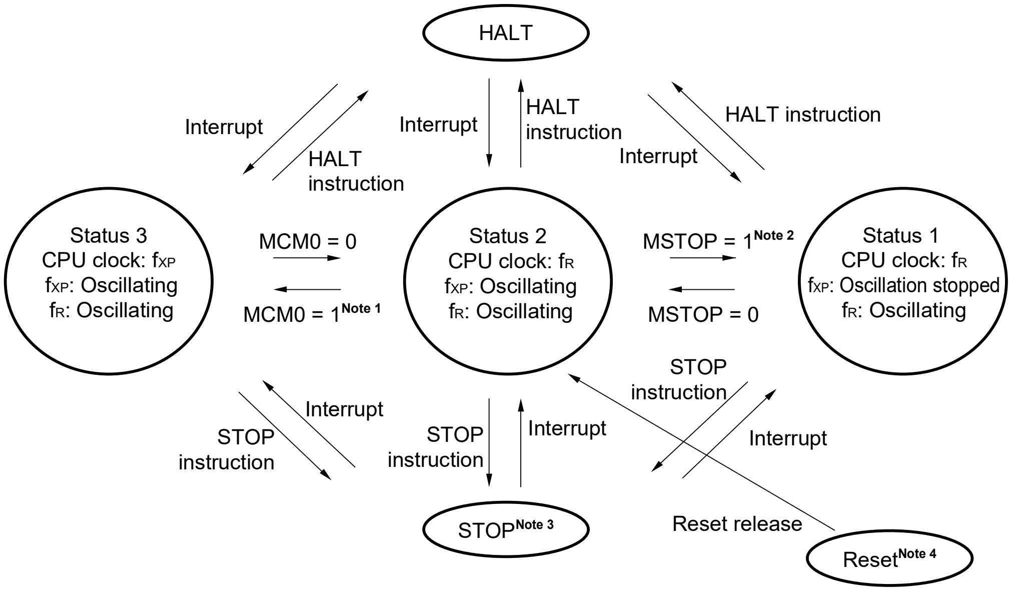
4. When “Ring-OSC can be stopped by software” is selected by the option byte, the clock supply to the watchdog timer is stopped after the HALT or STOP instruction has been executed, regardless of the setting of bit 0 (RSTOP) of the Ring-OSC mode register (RCM) and bit 0 (MCM0) of the main clock mode register (MCM).

5. The operation cannot be shifted between subsystem clock operation and Ring-OSC operation.

6. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT)

Figure 5-14. Status Transition Diagram (3/4)

(3) When "Ring-OSC cannot be stopped" is selected by option byte (when subsystem clock is not used)



Notes 1. Before shifting from status 2 to status 3 after reset and STOP are released, check the high-speed system clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

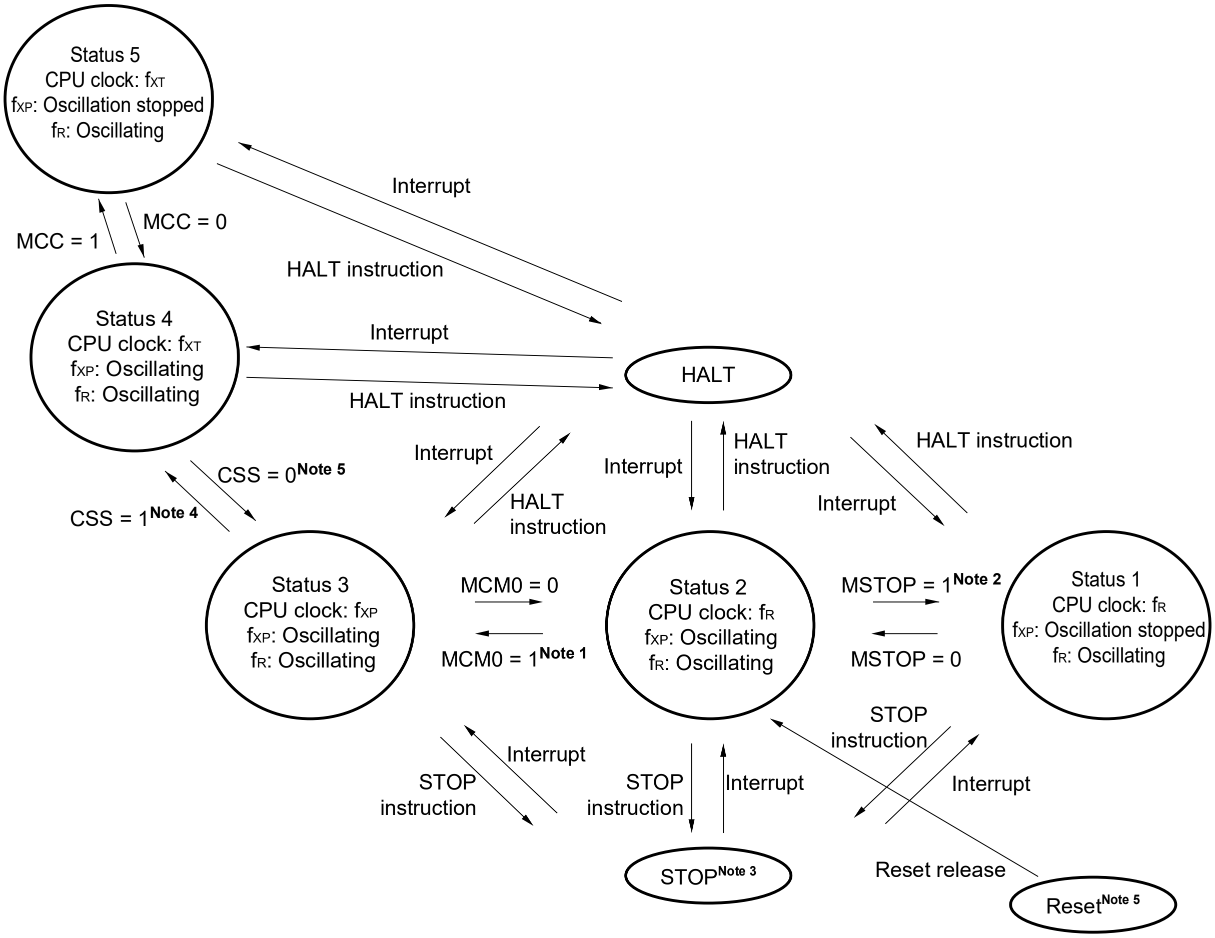
2. When shifting from status 2 to status 1, make sure that MCS is 0.

3. The watchdog timer operates using Ring-OSC even in STOP mode if "Ring-OSC cannot be stopped" is selected by the option byte. Ring-OSC division can be selected as the count source of 8-bit timer H1 (TMH1), so clear the watchdog timer using the TMH1 interrupt request before watchdog timer overflow. If this processing is not performed, an internal reset signal is generated at watchdog timer overflow after STOP instruction execution.

4. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT)

Figure 5-14. Status Transition Diagram (4/4)

(4) When “Ring-OSC cannot be stopped” is selected by option byte (when subsystem clock is used)



Notes 1. Before shifting from status 2 to status 3 after reset and STOP are released, check the high-speed system clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

2. When shifting from status 2 to status 1, make sure that MCS is 0.

3. The watchdog timer operates using Ring-OSC even in STOP mode if “Ring-OSC cannot be stopped” is selected by the option byte. Ring-OSC division can be selected as the count source of 8-bit timer H1 (TMH1), so clear the watchdog timer using the TMH1 interrupt request before watchdog timer overflow. If this processing is not performed, an internal reset signal is generated at watchdog timer overflow after STOP instruction execution.

4. The operation cannot be shifted between subsystem clock operation and Ring-OSC operation.

5. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT)

Table 5-3. Relationship Between Operation Clocks in Each Operation Status

Status Operation Mode	High-Speed System Clock Oscillator		Ring-OSC Oscillator			Subsystem Clock Oscillator	CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
	MSTOP = 0 MCC = 0	MSTOP = 1 MCC = 1	**Note 1**	**Note 2**					
				RSTOP = 0	RSTOP = 1			MCM0 = 0	MCM0 = 1
Reset	Stopped		Stopped			Oscillating	Ring-OSC	Stopped	
STOP			Oscillating	Oscillating	Stopped		**Note 3**	Stopped	
HALT	Oscillating	Stopped					**Note 4**	Ring-OSC	High-speed system clock

Notes 1. When "Cannot be stopped" is selected for Ring-OSC by the option byte.

2. When "Can be stopped by software" is selected for Ring-OSC by the option byte.

3. Operates using the CPU clock at STOP instruction execution.

4. Operates using the CPU clock at HALT instruction execution.

Caution The RSTOP setting is valid only when "Can be stopped by software" is set for Ring-OSC by the option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
MCC: Bit 7 of the processor clock control register (PCC)
RSTOP: Bit 0 of the Ring-OSC mode register (RCM)
MCM0: Bit 0 of the main clock mode register (MCM)

Table 5-4. Oscillation Control Flags and Clock Oscillation Status

		High-Speed System Clock Oscillator	Ring-OSC Oscillator
MSTOP = 1Note	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1	Setting prohibited	
MSTOP = 0Note	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped
MCC = 1Note	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1		Stopped
MCC = 0Note	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped

Note Setting high-speed system clock oscillator oscillating/stopped differs depending on the CPU clock used.

- When the Ring-OSC clock is used as the CPU clock: Set using the MSTOP bit
- When the subsystem clock is used as the CPU clock: Set using the MCC bit

Caution The RSTOP setting is valid only when "Can be stopped by software" is set for Ring-OSC by the option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
MCC: Bit 7 of the processor clock control register (PCC)
RSTOP: Bit 0 of the Ring-OSC mode register (RCM)

5.6 Time Required to Switch Between Ring-OSC Clock and High-Speed System Clock

Bit 0 (MCM0) of the main clock mode register (MCM) is used to switch between the Ring-OSC clock and high-speed system clock.

In the actual switching operation, switching does not occur immediately after MCM0 rewrite; several instructions are executed using the pre-switch clock after switching MCM0 (refer to **Table 5-5**).

Bit 1 (MCS) of MCM is used to judge that operation is performed using either the Ring-OSC clock or high-speed system clock.

To stop the original clock after switching the clock, wait for the number of clocks shown in Table 5-5 before stopping.

Table 5-5. Time Required to Switch Between Ring-OSC Clock and High-Speed System Clock

PCC			Time Required for Switching	
PCC2	PCC1	PCC0	High-Speed System Clock→ Ring-OSC	Ring-OSC → High-Speed System Clock
0	0	0	f_{XP}/f_R + 1 clock	2 clocks
0	0	1	$f_{XP}/2f_R$ + 1 clock	

Caution To calculate the maximum time, set f_R = 120 kHz.

Remarks 1. PCC: Processor clock control register

2. f_{XP}: High-speed system clock oscillation frequency

3. f_R: Ring-OSC clock oscillation frequency

4. The maximum time is the number of clocks of the CPU clock before switching.

5.7 Time Required for CPU Clock Switchover

The CPU clock can be switched using bits 0 to 2 (PCC0 to PCC2) and bit 4 (CSS) of the processor clock control register (PCC).

The actual switchover operation is not performed immediately after rewriting to the PCC; operation continues on the pre-switchover clock for several instructions (refer to **Table 5-6**).

Whether the system is operating on the high-speed system clock (or Ring-OSC clock) or the subsystem clock can be ascertained using bit 5 (CLS) of the PCC register.

Table 5-6. Maximum Time Required for CPU Clock Switchover

Set Value Before Switchover				Set Value After Switchover																							
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×
0	0	0	0					16 clocks				16 clocks				16 clocks				16 clocks				f_{XP}/f_{XT} clocks (489 clocks)			
	0	0	1	8 clocks								8 clocks				8 clocks				8 clocks				$f_{XP}/2f_{XT}$ clocks (245 clocks)			
	0	1	0	4 clocks				4 clocks								4 clocks				4 clocks				$f_{XP}/4f_{XT}$ clocks (123 clocks)			
	0	1	1	2 clocks				2 clocks				2 clocks								2 clocks				$f_{XP}/8f_{XT}$ clocks (62 clocks)			
	1	0	0	1 clock				1 clock				1 clock				1 clock								$f_{XP}/16f_{XT}$ clocks (31 clocks)			
1	×	×	×	1 clock				1 clock				1 clock				1 clock				1 clock							

Cautions 1. Selection of the CPU clock cycle division factor (PCC0 to PCC2) and switchover from the high-speed system clock to the subsystem clock (changing CSS from 0 to 1) should not be set simultaneously.

Simultaneous setting is possible, however, for selection of the CPU clock cycle division factor (PCC0 to PCC2) and switchover from the subsystem clock to the high-speed system clock (changing CSS from 1 to 0).

2. While the CPU is operating on Ring-OSC, setting the following values is prohibited.

- **CSS, PCC2, PCC1, PCC0 = 0, 0, 1, 0**
- **CSS, PCC2, PCC1, PCC0 = 0, 0, 1, 1**
- **CSS, PCC2, PCC1, PCC0 = 0, 1, 0, 0**

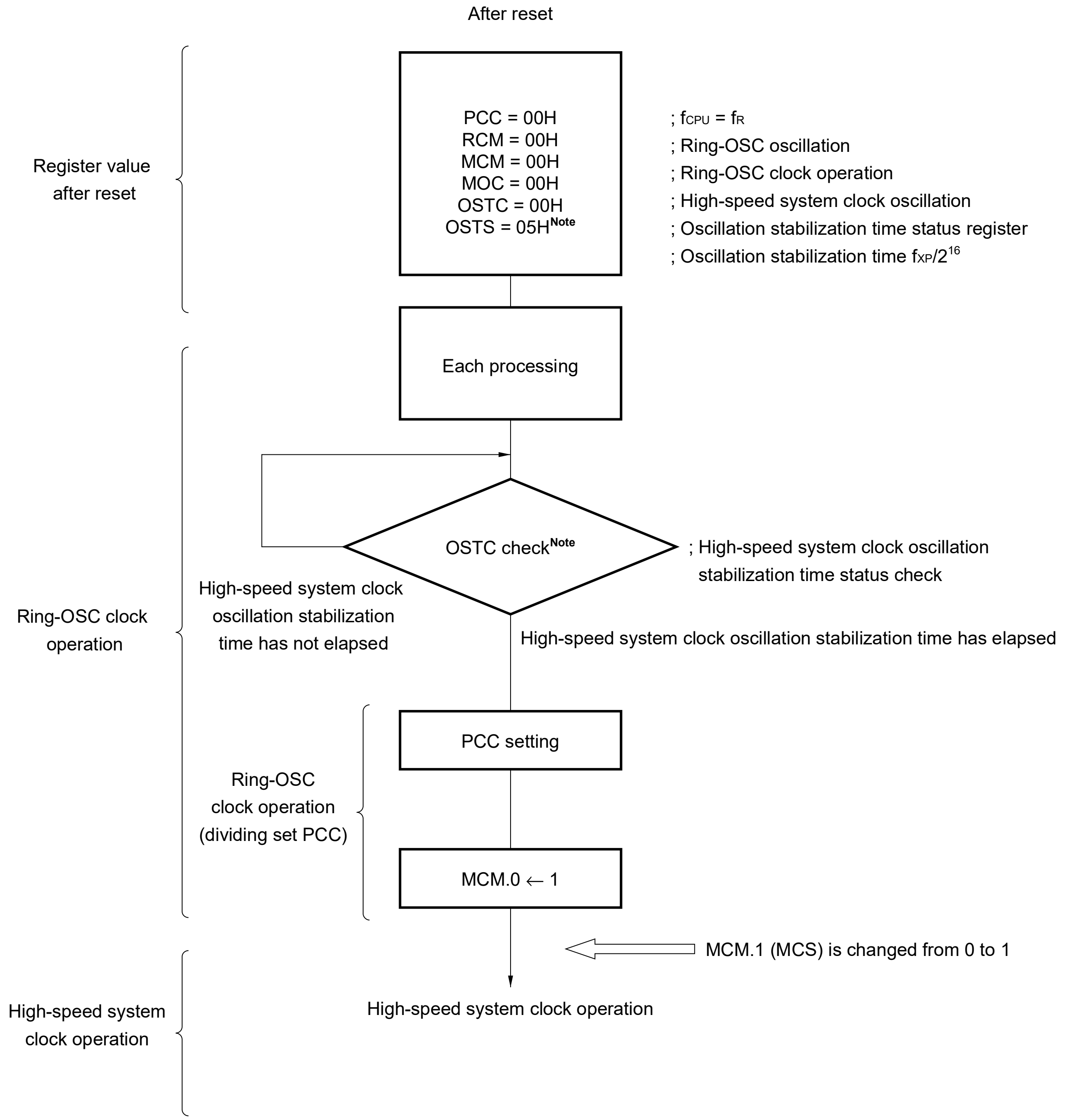
Remarks 1. The maximum time is the number of clocks of the pre-switchover CPU clock.

2. Figures in parentheses apply to operation with f_{XP} = 16 MHz and f_{XT} = 32.768 kHz.

5.8 Clock Switching Flowchart and Register Setting

5.8.1 Switching from Ring-OSC clock to high-speed system clock

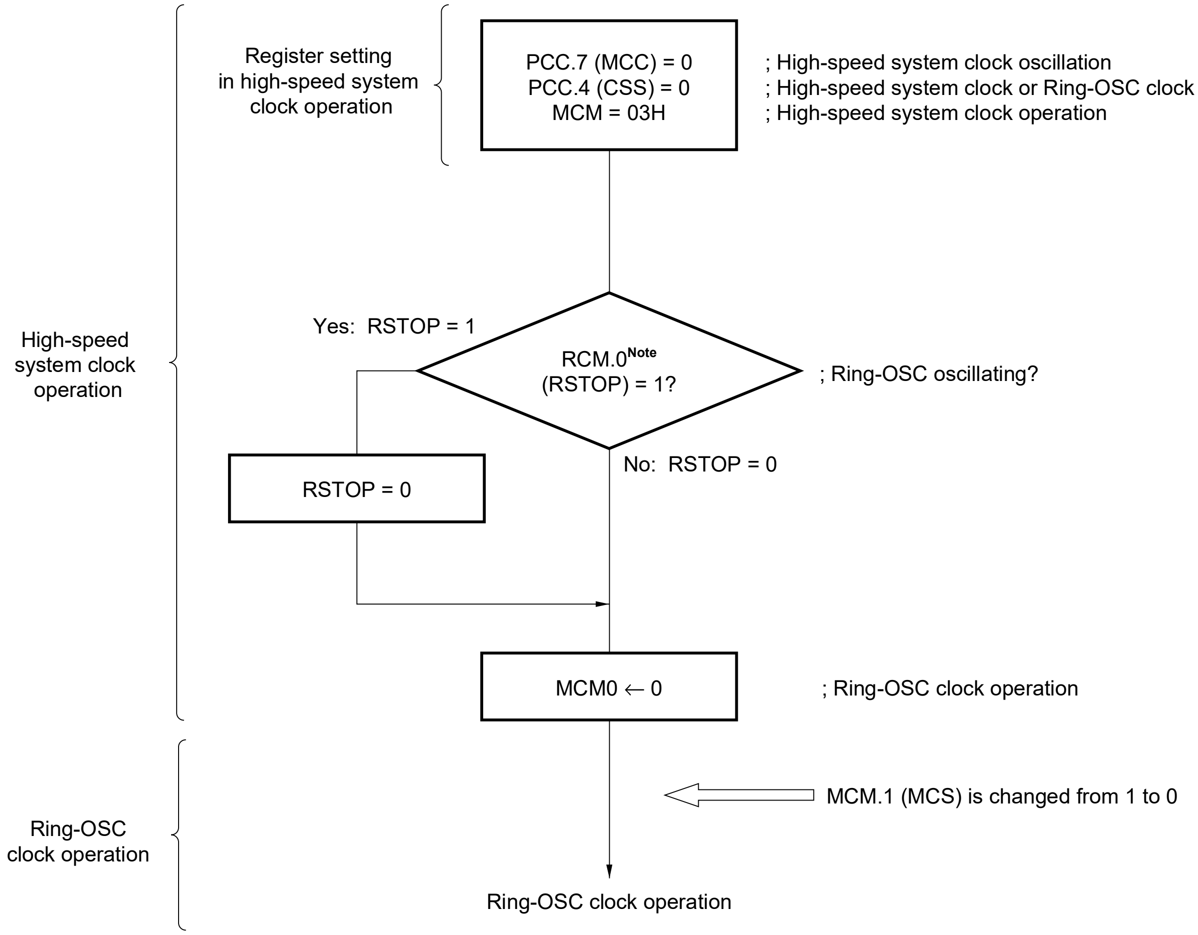
Figure 5-15. Switching from Ring-OSC Clock to High-Speed System Clock (Flowchart)


After reset
Register value after reset
PCC = 00H
RCM = 00H
MCM = 00H
MOC = 00H
OSTC = 00H
OSTS = 05H Note
; fCPU = fR
; Ring-OSC oscillation
; Ring-OSC clock operation
; High-speed system clock oscillation
; Oscillation stabilization time status register
; Oscillation stabilization time fXP/2^16
Each processing
Ring-OSC clock operation
OSTC check Note
; High-speed system clock oscillation stabilization time status check
High-speed system clock oscillation stabilization time has not elapsed
High-speed system clock oscillation stabilization time has elapsed
PCC setting
Ring-OSC clock operation (dividing set PCC)
MCM.0 ← 1
MCM.1 (MCS) is changed from 0 to 1
High-speed system clock operation
High-speed system clock operation

Note Check the oscillation stabilization wait time of the high-speed system clock oscillator after reset release using the OSTC register and then switch to the high-speed system clock operation after the oscillation stabilization wait time has elapsed. The OSTS register setting is valid only after STOP mode is released by interrupt during high-speed system clock operation.

5.8.2 Switching from high-speed system clock to Ring-OSC clock

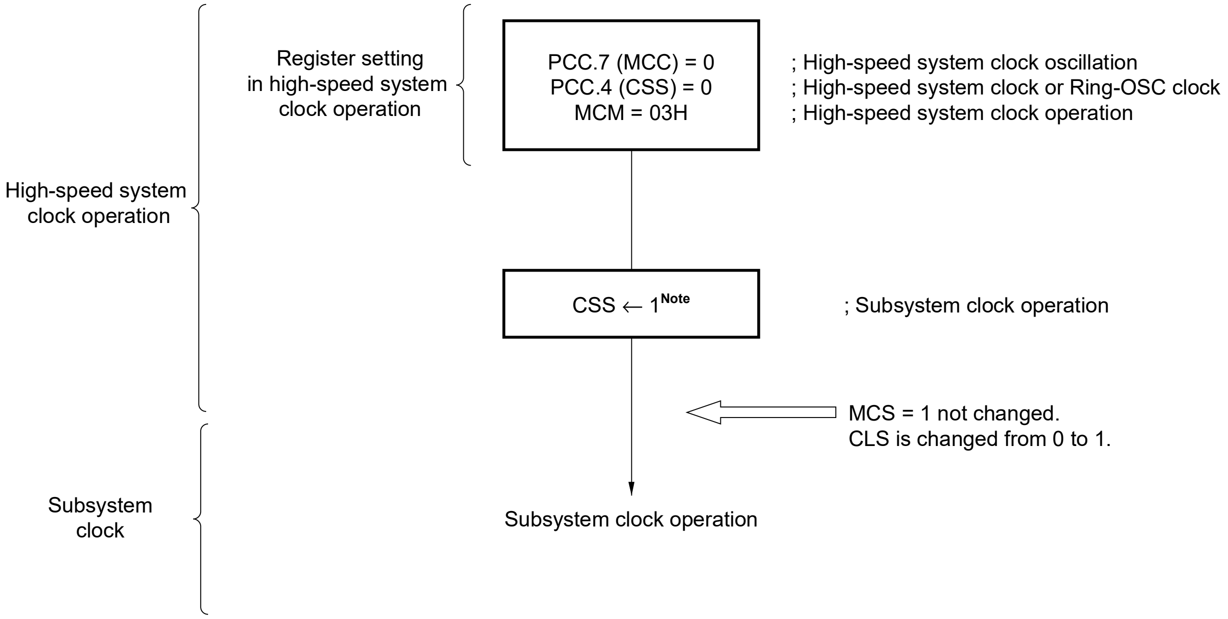
Figure 5-16. Switching from High-Speed System Clock to Ring-OSC Clock (Flowchart)



Note Required only when “can be stopped by software” is selected for Ring-OSC by the option byte.

5.8.3 Switching from high-speed system clock to subsystem clock

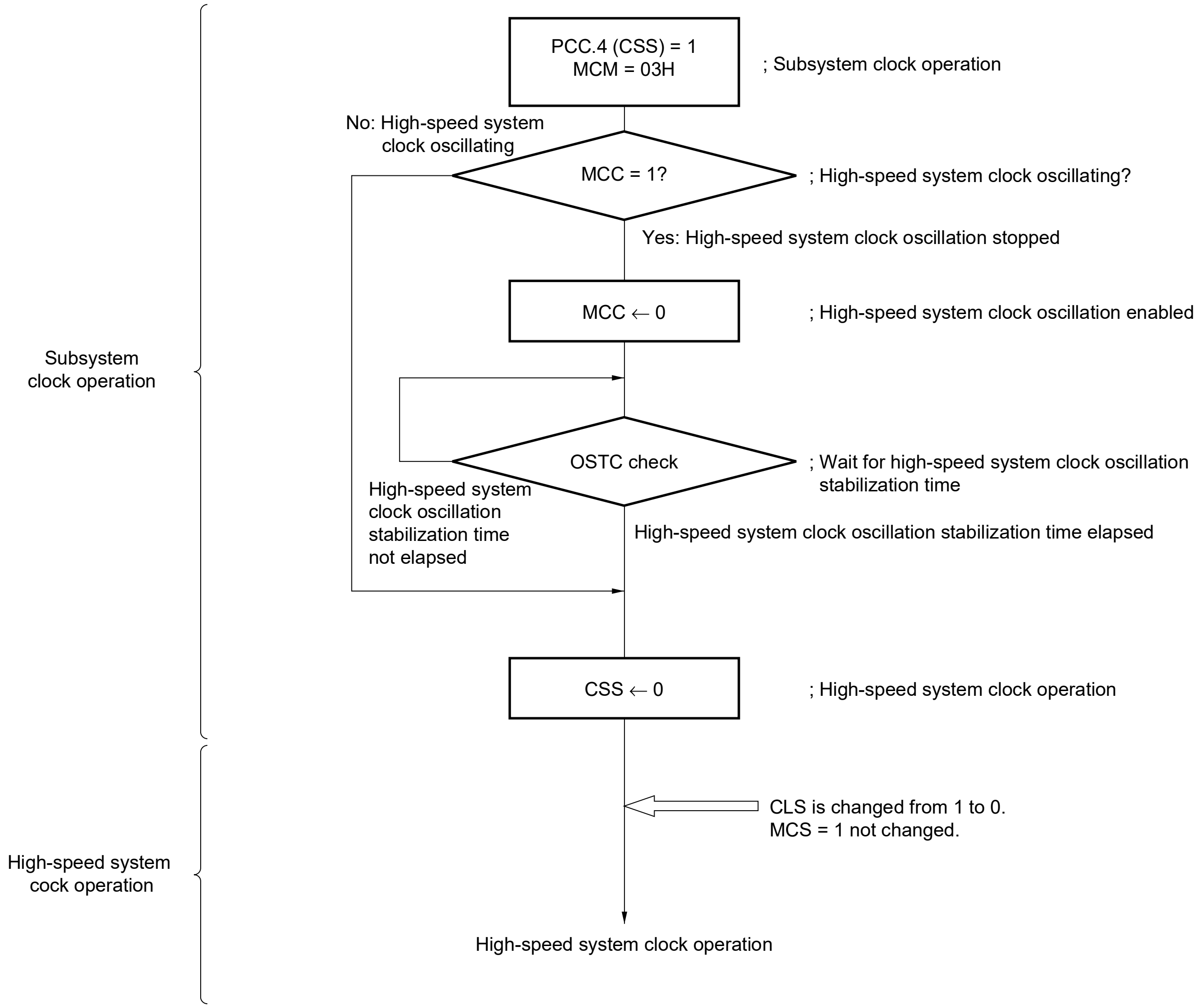
Figure 5-17. Switching from High-Speed System Clock to Subsystem Clock (Flowchart)


High-speed system clock operation
Register setting in high-speed system clock operation
PCC.7 (MCC) = 0
PCC.4 (CSS) = 0
MCM = 03H
; High-speed system clock oscillation
; High-speed system clock or Ring-OSC clock
; High-speed system clock operation
CSS ← 1Note
; Subsystem clock operation
MCS = 1 not changed.
CLS is changed from 0 to 1.
Subsystem clock
Subsystem clock operation

Note Set CSS to 1 after confirming that oscillation of the subsystem clock is stabilized.

5.8.4 Switching from subsystem clock to high-speed system clock

Figure 5-18. Switching from Subsystem Clock to High-Speed System Clock (Flowchart)


Subsystem
clock operation
PCC.4 (CSS) = 1
MCM = 03H
; Subsystem clock operation
No: High-speed system
clock oscillating
MCC = 1?
; High-speed system clock oscillating?
Yes: High-speed system clock oscillation stopped
MCC ← 0
; High-speed system clock oscillation enabled
OSTC check
; Wait for high-speed system clock oscillation
stabilization time
High-speed system
clock oscillation
stabilization time
not elapsed
High-speed system clock oscillation stabilization time elapsed
CSS ← 0
; High-speed system clock operation
CLS is changed from 1 to 0.
MCS = 1 not changed.
High-speed system
cock operation
High-speed system clock operation

5.8.5 Register settings

The table below shows the statuses of the setting flags and status flags when each mode is set.

Table 5-7. Clock and Register Setting

f_{CPU}	Mode	Setting Flag					Status Flag	
		PCC Register		MCM Register	MOC Register	RCM Register	PCC Register	MCM Register
		MCC	CSS	MCM0	MSTOP	RSTOP[Note 1]	CLS	MCS
High-speed system clock[Note 2]	Ring-OSC oscillating	0	0	1	0	0	0	1
	Ring-OSC stopped	0	0	1	0	1	0	1
Ring-OSC clock	High-speed system clock oscillating	0	0	0	0	0	0	0
	High-speed system clock stopped	0[Note 3]	0	0	1	0	0	0
Subsystem clock[Note 4]	High-speed system clock oscillating, Ring-OSC oscillating	0	1	1[Note 5]	0[Note 6]	0	1	1
	High-speed system clock stopped, Ring-OSC oscillating	1	1	1[Note 5]	0[Note 6]	0	1	1
	High-speed system clock oscillating, Ring-OSC stopped	0	1	1[Note 5]	0[Note 6]	1	1	1
	High-speed system clock stopped, Ring-OSC stopped	1	1	1[Note 5]	0[Note 6]	1	1	1

Notes 1. Valid only when "can be stopped by software" is selected for Ring-OSC by the option byte.

2. Do not set MCC = 1 or MSTOP = 1 during high-speed system clock operation (even if MCC = 1 or MSTOP = 1 is set, the high-speed system clock oscillation does not stop).

3. Do not set MCC = 1 during Ring-OSC operation (even if MCC = 1 is set, the high-speed system clock oscillation does not stop). To stop high-speed system clock oscillation during Ring-OSC operation, use MSTOP.

4. Shifting to subsystem clock operation mode must be performed from the high-speed system clock operation mode. From subsystem clock operation mode, only high-speed system clock operation mode can be shifted to.

5. Do not set MCM0 = 0 (shifting to Ring-OSC) during subsystem clock operation.

6. Do not set MSTOP = 1 during subsystem clock operation (even if MSTOP = 1 is set, high-speed system clock oscillation does not stop). To stop high-speed system clock oscillation during subsystem clock operation, use MCC.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER 00

6.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square-wave output
- One-shot pulse output

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(3) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

(4) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(5) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(6) One-shot pulse output

16-bit timer/event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

6.2 Configuration of 16-Bit Timer/Event Counter 00

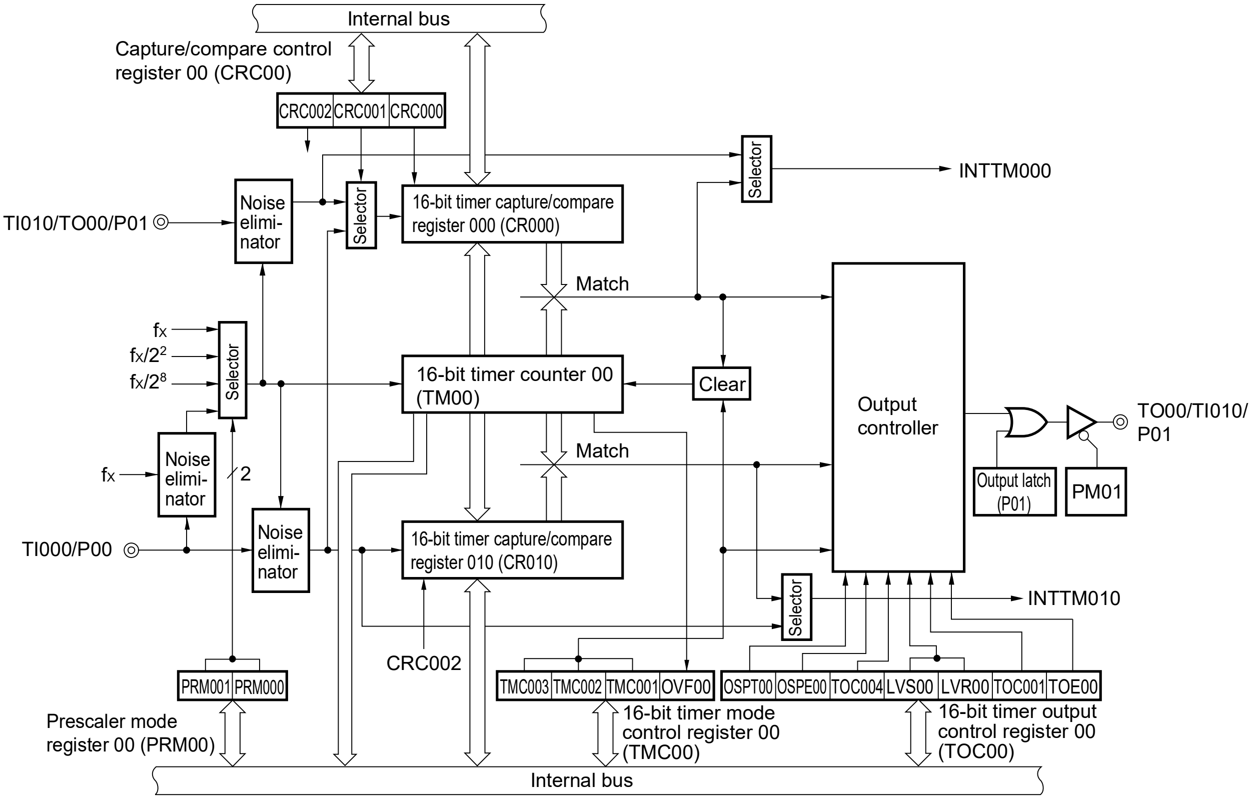
16-bit timer/event counter 00 includes the following hardware.

Table 6-1. Configuration of 16-Bit Timer/Event Counter 00

Item	Configuration
Timer counter	16 bits (TM00)
Register	16-bit timer capture/compare register: 16 bits (CR000, CR010)
Timer input	TI000, TI010
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) 16-bit timer capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 0 (PM0) Port register 0 (P0)

Figure 6-1 shows the block diagram.

Figure 6-1. Block Diagram of 16-Bit Timer/Event Counter 00



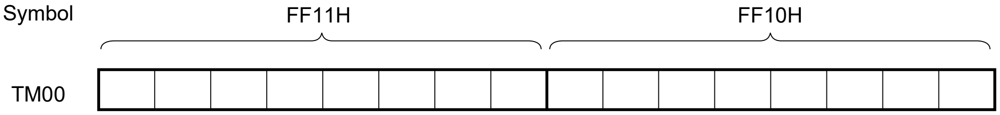
(1) 16-bit timer counter 00 (TM00)

TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the input clock.

Figure 6-2. Format of 16-Bit Timer Counter 00 (TM00)

Address: FF10H, FF11H After reset: 0000H R



The count value is reset to 0000H in the following cases.

<1> At $\overline{\text{RESET}}$ input

<2> If TMC003 and TMC002 are cleared

<3> If the valid edge of the TI000 pin is input in the mode in which clear & start occurs when inputting the valid edge of the TI000 pin

<4> If TM00 and CR000 match in the mode in which clear & start occurs on a match of TM00 and CR000

<5> OSPT00 is set in one-shot pulse output mode

(2) 16-bit timer capture/compare register 000 (CR000)

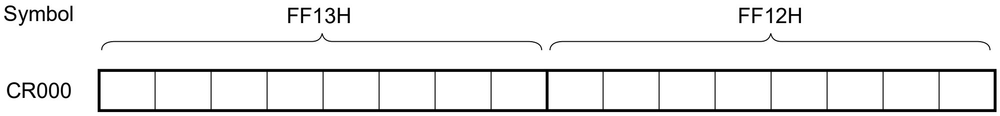
CR000 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC000) of capture/compare control register 00 (CRC00).

CR000 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CR000 to 0000H.

Figure 6-3. Format of 16-Bit Timer Capture/Compare Register 000 (CR000)

Address: FF12H, FF13H After reset: 0000H R/W



- **When CR000 is used as a compare register**

  The value set in CR000 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM000) is generated if they match. The set value is held until CR000 is rewritten.

- **When CR000 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin or the TI010 pin as the capture trigger. The TI000 or TI010 valid edge is set using prescaler mode register 00 (PRM00) (refer to **Table 6-2**).

Table 6-2. CR000 Capture Trigger and Valid Edges of TI000 and TI010 Pins

(1) TI000 pin valid edge selected as capture trigger (CRC001 = 1, CRC000 = 1)

CR000 Capture Trigger	TI000 Pin Valid Edge		
		ES001	ES000
Falling edge	Rising edge	0	1
Rising edge	Falling edge	0	0
No capture operation	Both rising and falling edges	1	1

(2) TI010 pin valid edge selected as capture trigger (CRC001 = 0, CRC000 = 1)

CR000 Capture Trigger	TI010 Pin Valid Edge		
		ES101	ES100
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES001, ES000 = 1, 0 and ES101, ES100 = 1, 0 is prohibited.

2. ES001, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)
ES101, ES100: Bits 7 and 6 of prescaler mode register 00 (PRM00)
CRC001, CRC000: Bits 1 and 0 of capture/compare control register 00 (CRC00)

Cautions 1. Set a value other than 0000H in CR000 in the mode in which clear & start occurs on a match of TM00 and CR000.

2. If CR000 is cleared to 0000H in the free-running mode and in the clear mode using the valid edge of the TI000 pin, an interrupt request (INTTM000) is generated when the value of CR000 changes from 0000H to 0001H following TM00 overflow (FFFFH). Moreover, INTTM000 is generated after a match of TM00 and CR000 is detected, a valid edge of the TI010 pin is detected, and the timer is cleared by a one-shot trigger.

3. When P01 is used as the valid edge input of the TI010 pin, it cannot be used as the timer output (TO00). Moreover, when P01 is used as TO00, it cannot be used as the valid edge input of the TI010 pin.

4. When CR000 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

5. Do not rewrite CR000 during TM00 operation.

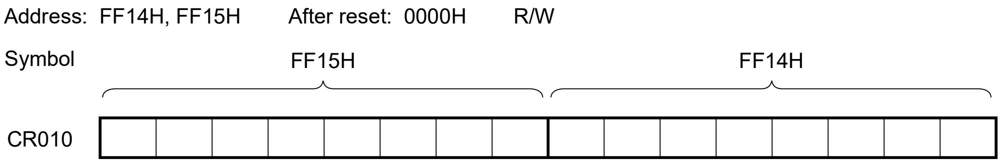
(3) 16-bit timer capture/compare register 010 (CR010)

CR010 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC002) of capture/compare control register 00 (CRC00).

CR010 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CR010 to 0000H.

Figure 6-4. Format of 16-Bit Timer Capture/Compare Register 010 (CR010)



- **When CR010 is used as a compare register**

  The value set in CR010 is constantly compared with the 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM010) is generated if they match. The set value is held until CR010 is rewritten.

- **When CR010 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set using prescaler mode register 00 (PRM00) (refer to **Table 6-3**).

Table 6-3. CR010 Capture Trigger and Valid Edge of TI000 Pin (CRC002 = 1)

CR010 Capture Trigger	TI000 Pin Valid Edge		
		ES001	ES000
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES001, ES000 = 1, 0 is prohibited.

2. ES001, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)
CRC002: Bit 2 of capture/compare control register 00 (CRC00)

Cautions 1. If CR010 is cleared to 0000H, an interrupt request (INTTM010) is generated when the value of CR010 changes from 0000H to 0001H following TM00 overflow (FFFFH). Moreover, INTTM010 is generated after a match of TM00 and CR010 is detected, a valid edge of the TI000 pin is detected, and the timer is cleared by a one-shot trigger.

2. When CR010 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

3. CR010 can be rewritten during TM00 operation. For details, refer to Caution 2 in Figure 6-15.

6.3 Registers Controlling 16-Bit Timer/Event Counter 00

The following six registers are used to control 16-bit timer/event counter 00.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 0 (PM0)
- Port register 0 (P0)

(1) 16-bit timer mode control register 00 (TMC00)

This register sets the 16-bit timer operating mode, the 16-bit timer counter 00 (TM00) clear mode, and output timing, and detects an overflow.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC00 to 00H.

Caution 16-bit timer counter 00 (TM00) starts operation at the moment TMC002 and TMC003 are set to values other than 0, 0 (operation stop mode), respectively. Clear TMC002 and TMC003 to 0, 0 to stop the operation.

Figure 6-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address FFBAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	Operating mode and clear mode selection	TO00 inversion timing selection	Interrupt request generation
0	0	0	Operation stop (TM00 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM00 and CR000 or match between TM00 and CR010	Generated on match between TM00 and CR000, or match between TM00 and CR010
0	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 pin valid edge	
1	0	0	Clear & start occurs on TI000 pin valid edge	–	
1	0	1			
1	1	0	Clear & start occurs on match between TM00 and CR000	Match between TM00 and CR000 or match between TM00 and CR010	
1	1	1		Match between TM00 and CR000, match between TM00 and CR010 or TI000 pin valid edge	

OVF00	16-bit timer counter 00 (TM00) overflow detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF00 flag.

2. Set the valid edge of the TI000/P00 pin using prescaler mode register 00 (PRM00).

3. If any of the following modes is selected: the mode in which clear & start occurs on match between TM00 and CR000, the mode in which clear & start occurs at the TI000 pin valid edge, or free-running mode, when the set value of CR000 is FFFFH and the TM00 value changes from FFFFH to 0000H, the OVF00 flag is set to 1.

Remark TO00: 16-bit timer/event counter 00 output pin
TI000: 16-bit timer/event counter 00 input pin
TM00: 16-bit timer counter 00
CR000: 16-bit timer capture/compare register 000
CR010: 16-bit timer capture/compare register 010

(2) Capture/compare control register 00 (CRC00)

This register controls the operation of the 16-bit timer capture/compare registers (CR000, CR010).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CRC00 to 00H.

Figure 6-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FFBCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR010 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC001	CR000 capture trigger selection
0	Captures on valid edge of TI010 pin
1	Captures on valid edge of TI000 pin by reverse phase

CRC000	CR000 operating mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC00.

2. When the mode in which clear & start occurs on a match between TM00 and CR000 is selected with 16-bit timer mode control register 00 (TMC00), CR000 should not be specified as a capture register.

3. The capture operation is not performed if both the rising and falling edges are specified as the valid edge of the TI000 pin.

4. To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

(3) 16-bit timer output control register 00 (TOC00)

This register controls the operation of the 16-bit timer/event counter 00 output controller. It sets/resets the timer output F/F (LV00), enables/disables output inversion and 16-bit timer/event counter 00 timer output, enables/disables the one-shot pulse output operation, and sets the one-shot pulse output trigger via software.

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TOC00 to 00H.

Figure 6-7. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FFBDH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger control via software
0	No one-shot pulse trigger
1	One-shot pulse trigger

OSPE00	One-shot pulse output operation control
0	Successive pulse output mode
1	One-shot pulse output mode[Note]

TOC004	Timer output F/F control using match of CR010 and TM00
0	Disables inversion operation
1	Enables inversion operation

LVS00	LVR00	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC001	Timer output F/F control using match of CR000 and TM00
0	Disables inversion operation
1	Enables inversion operation

TOE00	Timer output control
0	Disables output (output fixed to level 0)
1	Enables output

Note The one-shot pulse output mode operates correctly only in the free-running mode and the mode in which clear & start occurs at the TI000 pin valid edge. In the mode in which clear & start occurs on a match between the TM00 register and CR000 register, one-shot pulse output is not possible because an overflow does not occur.

Cautions 1. Timer operation must be stopped before setting other than TOC004.

2. If LVS00 and LVR00 are read, 0 is read.

3. OSPT00 is automatically cleared after data is set, so 0 is read.

4. Do not set OSPT00 to 1 other than in one-shot pulse output mode.

5. A write interval of two cycles or more of the count clock selected by prescaler mode register 00 (PRM00) is required to write to OSPT00 successively.

6. Do not set LVS00 to 1 before TOE00, and do not set LVS00 and TOE00 to 1 simultaneously.

7. Perform <1> and <2> below in the following order, not at the same time.

<1> Set TOC001, TOC004, TOE00, OSPE00: Timer output operation setting

<2> Set LVS00, LVR00: Timer output F/F setting

(4) Prescaler mode register 00 (PRM00)

This register is used to set the 16-bit timer counter 00 (TM00) count clock and TI000 and TI010 pin input valid edges.

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM00 to 00H.

Figure 6-8. Format of Prescaler Mode Register 00 (PRM00)

Address: FFBBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

PRM001	PRM000	Count clock selection[Note 1]
0	0	f_X (10 MHz)
0	1	$f_X/2^2$ (2.5 MHz)
1	0	$f_X/2^8$ (39.06 kHz)
1	1	TI000 valid edge[Note 2]

Notes 1. Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Count clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Count clock ≤ 5 MHz

2. The external clock requires a pulse two cycles longer than internal clock (f_X).

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the count clock is the Ring-OSC clock, the operation of 16-bit timer/event counter 00 is not guaranteed. When an external clock is used and when the Ring-OSC clock is selected and supplied to the CPU, the operation of 16-bit timer/event counter 00 is not guaranteed, either, because the Ring-OSC clock is supplied as the sampling clock to eliminate noise.

2. Always set data to PRM00 after stopping the timer operation.

3. If the valid edge of the TI000 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI000 pin and the capture trigger.

Cautions 4. If the TI000 or TI010 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI010 pin to enable the operation of 16-bit timer counter 00 (TM00). Care is therefore required when pulling up the TI000 or TI010 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

5. When P01 is used as the TI010 pin valid edge input pin, it cannot be used as the timer output (TO00), and when used as TO00, it cannot be used as the TI010 pin valid edge input pin.

Remarks 1. f_x: High-speed system clock oscillation frequency

2. Figures in parentheses are for operation with f_x = 10 MHz.

(5) Port mode register 0 (PM0)

This register sets port 0 input/output in 1-bit units.

When using the P01/TO00/TI010 pin for timer output, clear PM01 and the output latch of P01 to 0.

When using the P01/TO00/TI010 pin for timer input, set PM01 to 1. At this time, the output latch of P01 may be 0 or 1.

PM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM0 to FFH.

Figure 6-9. Format of Port Mode Register 0 (PM0)

Address: FF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

PM0n	P0n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Operation of 16-Bit Timer/Event Counter 00

6.4.1 Interval timer operation

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-10 allows operation as an interval timer.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (refer to **Figure 6-10** for the set value).
<2> Set any value to the CR000 register.
<3> Set the count clock by using the PRM00 register.
<4> Set the TMC00 register to start the operation (refer to **Figure 6-10** for the set value).

Caution CR000 cannot be rewritten during TM00 operation.

Remark For how to enable the INTTM000 interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

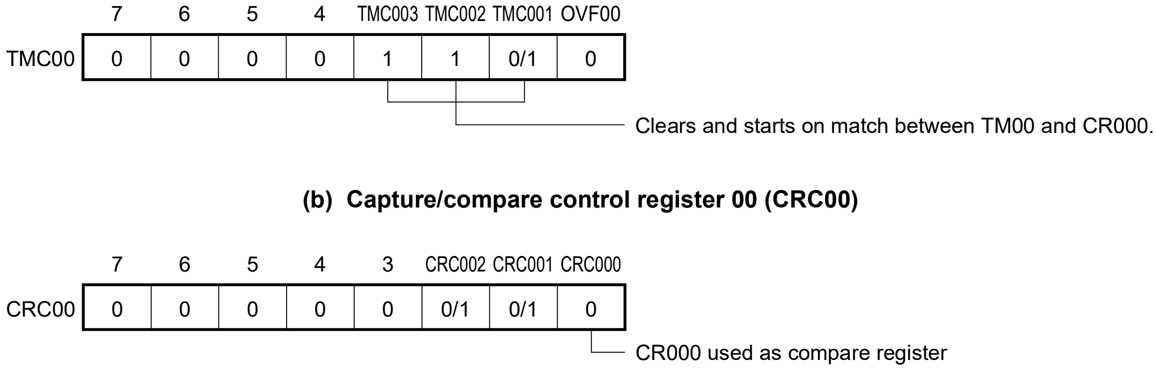
Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 000 (CR000) as the interval.

When the count value of 16-bit timer counter 00 (TM00) matches the value set in CR000, counting continues with the TM00 value cleared to 0 and the interrupt request signal (INTTM000) is generated.

The count clock of 16-bit timer/event counter 00 can be selected with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00).

Figure 6-10. Control Register Settings for Interval Timer Operation (1/2)

(a) 16-bit timer mode control register 00 (TMC00)



	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM00 and CR000.

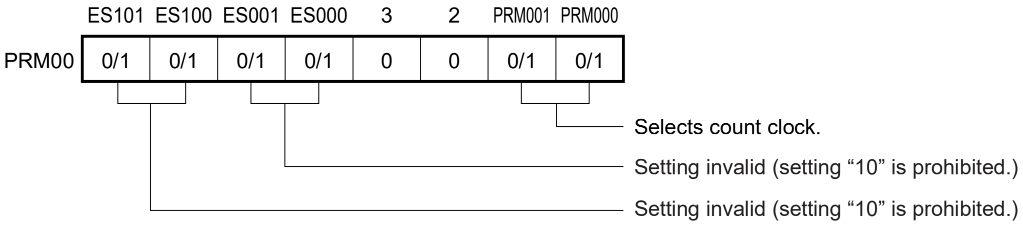
(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR000 used as compare register

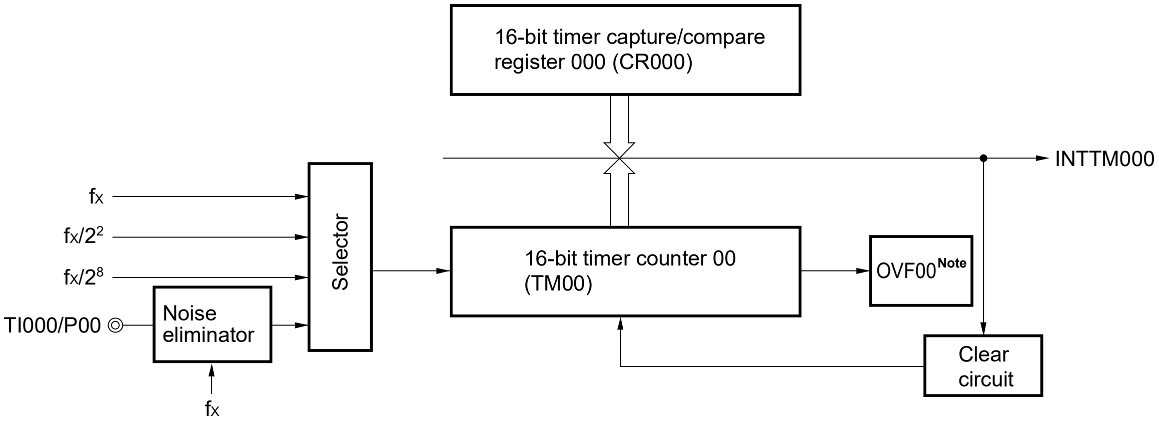
Figure 6-10. Control Register Settings for Interval Timer Operation (2/2)

(c) Prescaler mode register 00 (PRM00)


ES101
ES100
ES001
ES000
3
2
PRM001
PRM000
PRM00
0/1
0/1
0/1
0/1
0
0
0/1
0/1
Selects count clock.
Setting invalid (setting "10" is prohibited.)
Setting invalid (setting "10" is prohibited.)

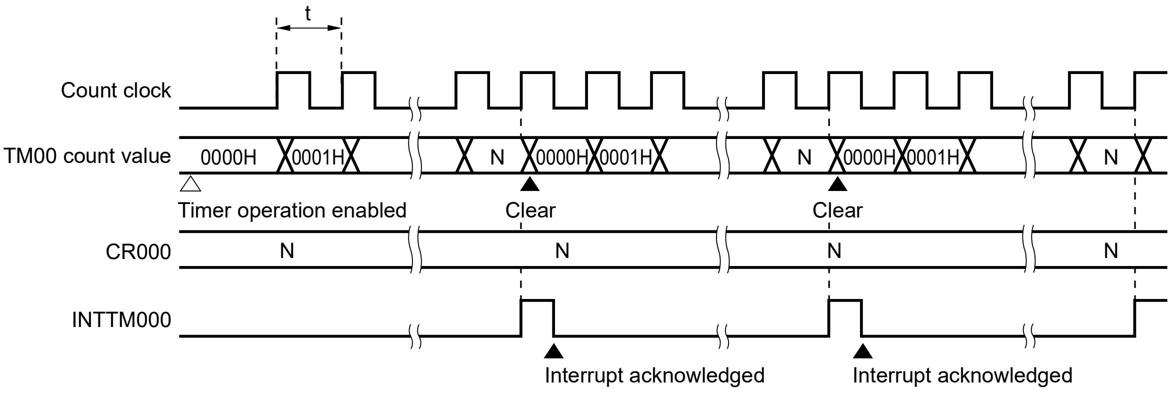
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer. See the description of the respective control registers for details.

Figure 6-11. Interval Timer Configuration Diagram



Note OVF00 is set to 1 only when 16-bit timer capture/compare register 000 is set to FFFFH.

Figure 6-12. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t
N = 0001H to FFFFH

6.4.2 PPG output operations

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 6-13 allows operation as PPG (Programmable Pulse Generator) output.

Setting

The basic operation setting procedure is as follows.

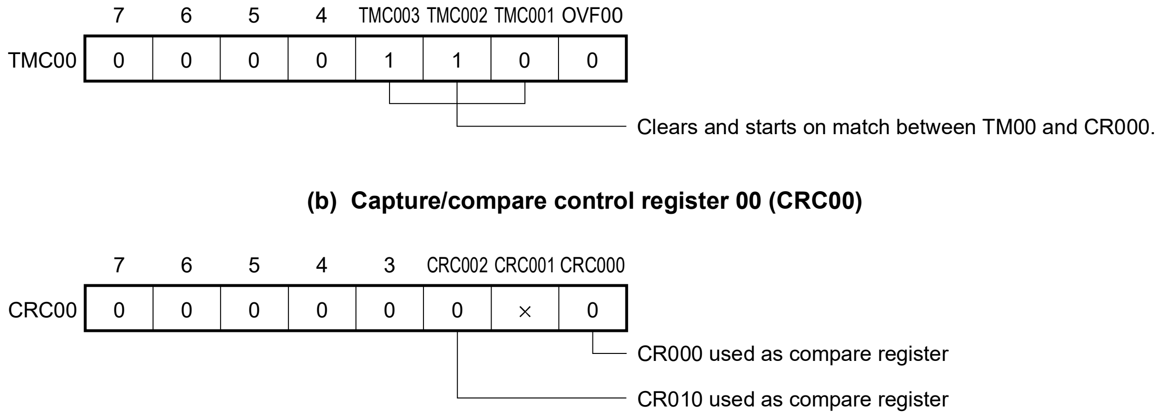
<1> Set the CRC00 register (refer to **Figure 6-13** for the set value).
<2> Set any value to the CR000 register as the cycle.
<3> Set any value to the CR010 register as the duty factor.
<4> Set the TOC00 register (refer to **Figure 6-13** for the set value).
<5> Set the count clock by using the PRM00 register.
<6> Set the TMC00 register to start the operation (refer to **Figure 6-13** for the set value).

Caution To change the value of the duty factor (the value of the CR010 register) during operation, refer to Caution 2 in Figure 6-15 PPG Output Operation Timing.

Remarks 1. For the setting of the TO00 pin, refer to **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

In the PPG output operation, rectangular waves are output from the TO00 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 010 (CR010) and in 16-bit timer capture/compare register 000 (CR000), respectively.

Figure 6-13. Control Register Settings for PPG Output Operation (1/2)



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0	0

Clears and starts on match between TM00 and CR000.

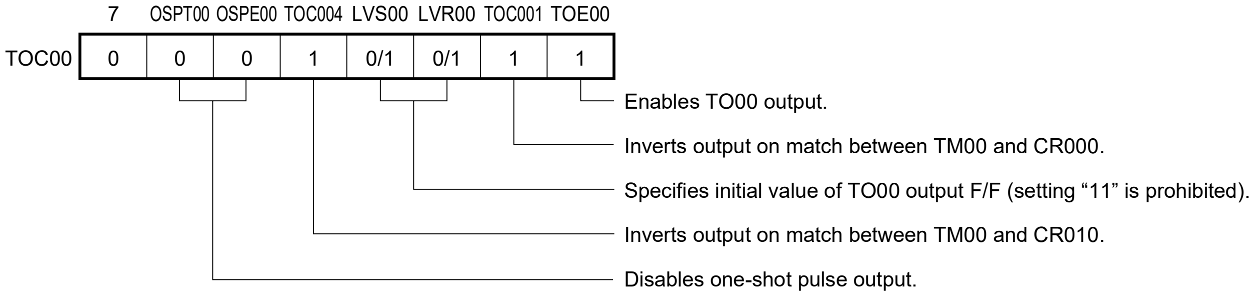
(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0	×	0

CR000 used as compare register
CR010 used as compare register

Figure 6-13. Control Register Settings for PPG Output Operation (2/2)

(c) 16-bit timer output control register 00 (TOC00)



	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	0	1	0/1	0/1	1	1

Enables TO00 output.

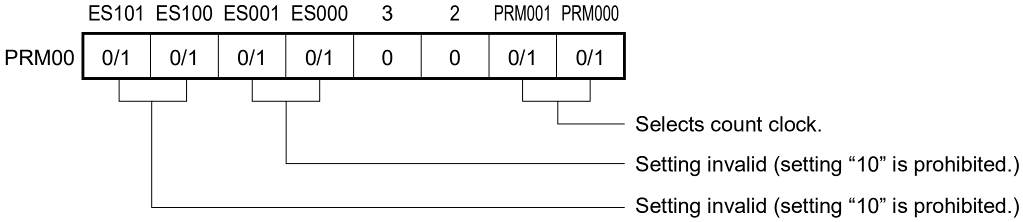
Inverts output on match between TM00 and CR000.

Specifies initial value of TO00 output F/F (setting “11” is prohibited).

Inverts output on match between TM00 and CR010.

Disables one-shot pulse output.

(d) Prescaler mode register 00 (PRM00)



	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0/1	0/1	0	0	0/1	0/1

Selects count clock.

Setting invalid (setting “10” is prohibited.)

Setting invalid (setting “10” is prohibited.)

Cautions 1. Values in the following range should be set in CR000 and CR010:
0000H ≤ CR010 < CR000 ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR000 setting value + 1) has a duty of (CR010 setting value + 1)/(CR000 setting value + 1).

Remark ×: don’t care

Figure 6-14. Configuration Diagram of PPG Output

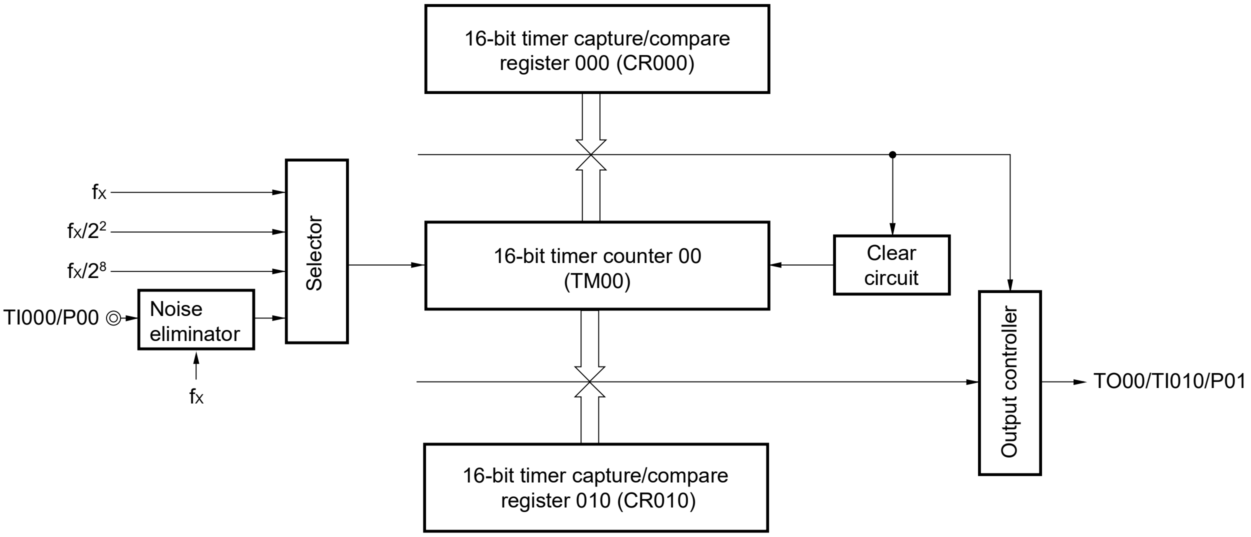
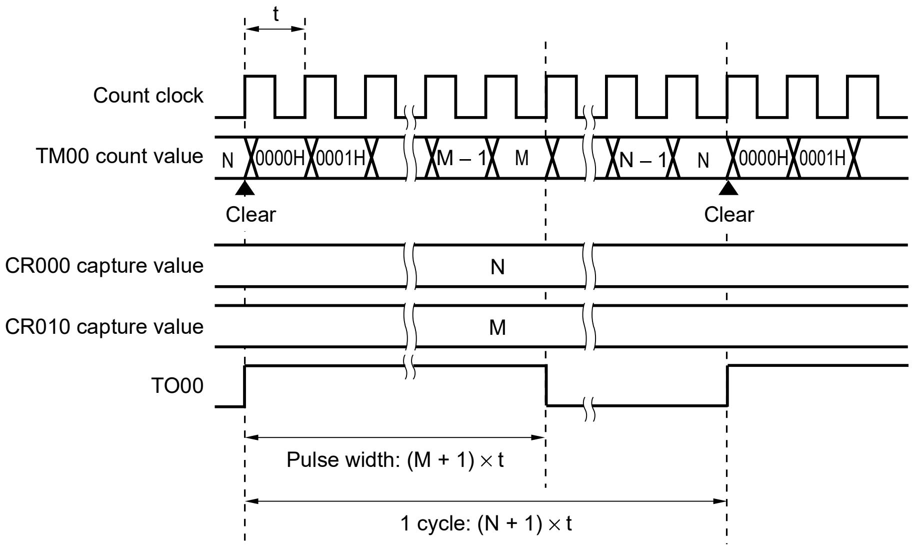

16-bit timer capture/compare register 000 (CR000)
fX
fX/2²
fX/2⁸
TI000/P00
Noise eliminator
fX
Selector
16-bit timer counter 00 (TM00)
Clear circuit
Output controller
TO00/TI010/P01
16-bit timer capture/compare register 010 (CR010)

Figure 6-15. PPG Output Operation Timing



Cautions 1. CR000 cannot be rewritten during TM00 operation.

2. In the PPG output operation, change the pulse width (rewrite CR010) during TM00 operation using the following procedure.

<1> Disable the timer output inversion operation by match of TM00 and CR010 (TOC004 = 0)

<2> Disable the INTTM010 interrupt (TMMK010 = 1)

<3> Rewrite CR010

<4> Wait for 1 cycle of the TM00 count clock

<5> Enable the timer output inversion operation by match of TM00 and CR010 (TOC004 = 1)

<6> Clear the interrupt request flag of INTTM010 (TMIF010 = 0)

<7> Enable the INTTM010 interrupt (TMMK010 = 0)

Remark 0000H ≤ M < N ≤ FFFFH

6.4.3 Pulse width measurement operations

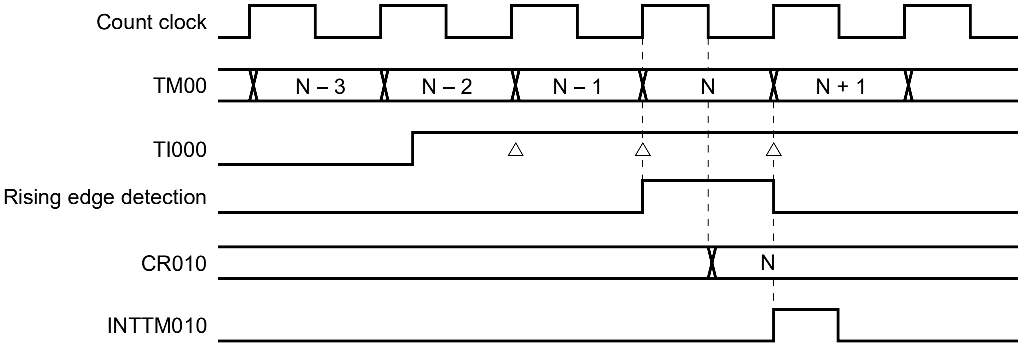
It is possible to measure the pulse width of the signals input to the TI000 pin and TI010 pin using 16-bit timer counter 00 (TM00).

There are two measurement methods: measuring with TM00 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI000 pin.

When an interrupt occurs, read the valid value of the capture register, check the overflow flag, and then calculate the necessary pulse width. Clear the overflow flag after checking it.

The capture operation is not performed until the signal pulse width is sampled in the count clock cycle selected by prescaler mode register 00 (PRM00) and the valid level of the TI000 or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-16. CR010 Capture Operation with Rising Edge Specified



Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (refer to **Figures 6-17**, **6-20**, **6-22**, and **6-24** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set the TMC00 register to start the operation (refer to **Figures 6-17**, **6-20**, **6-22**, and **6-24** for the set value).

Caution To use two capture registers, set the TI000 and TI010 pins.

Remarks 1. For the setting of the TI000 (or TI010) pin, refer to **6.3 (5) Port mode register 0 (PM0)**.

2. For how to enable the INTTM000 (or INTTM010) interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

(1) Pulse width measurement with free-running counter and one capture register

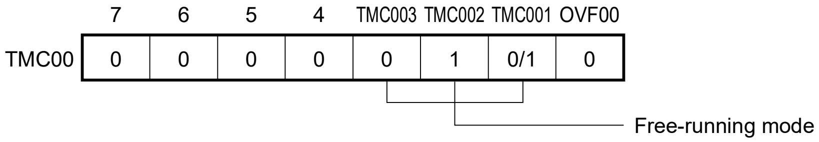
When 16-bit timer counter 00 (TM00) is operated in free-running mode, and the edge specified by prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an external interrupt request signal (INTTM010) is set.

Specify both the rising and falling edges of the TI000 pin by using bits 4 and 5 (ES000 and ES001) of PRM00.

Sampling is performed using the count clock selected by PRM00, and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

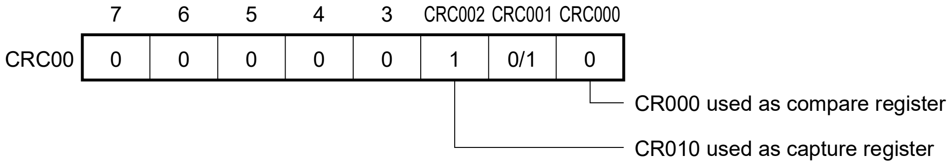
Figure 6-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register (When TI000 and CR010 Are Used)

(a) 16-bit timer mode control register 00 (TMC00)



	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

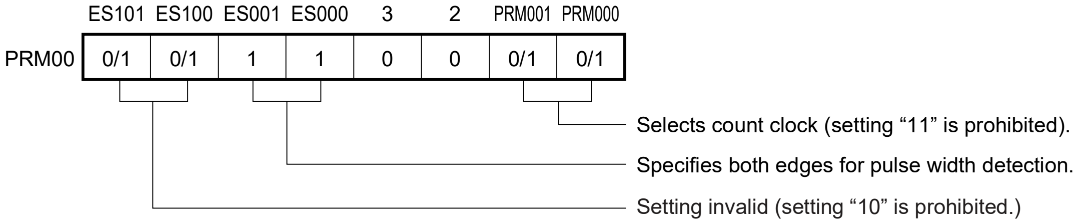
(b) Capture/compare control register 00 (CRC00)



	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	0/1	0

CR000 used as compare register
CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)



	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	1	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).
Specifies both edges for pulse width detection.
Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 6-18. Configuration Diagram for Pulse Width Measurement with Free-Running Counter

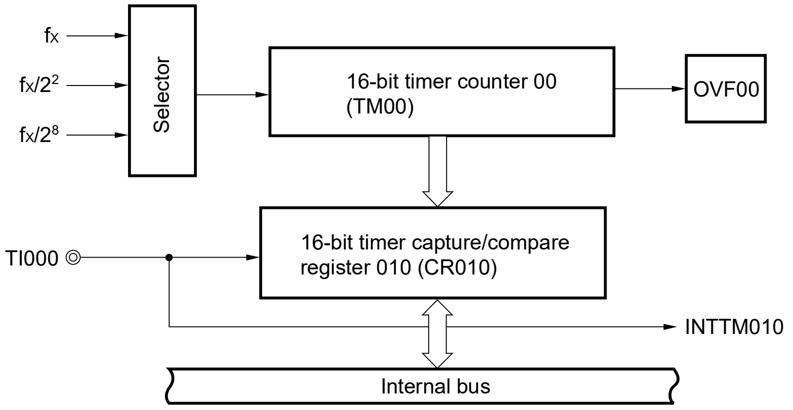
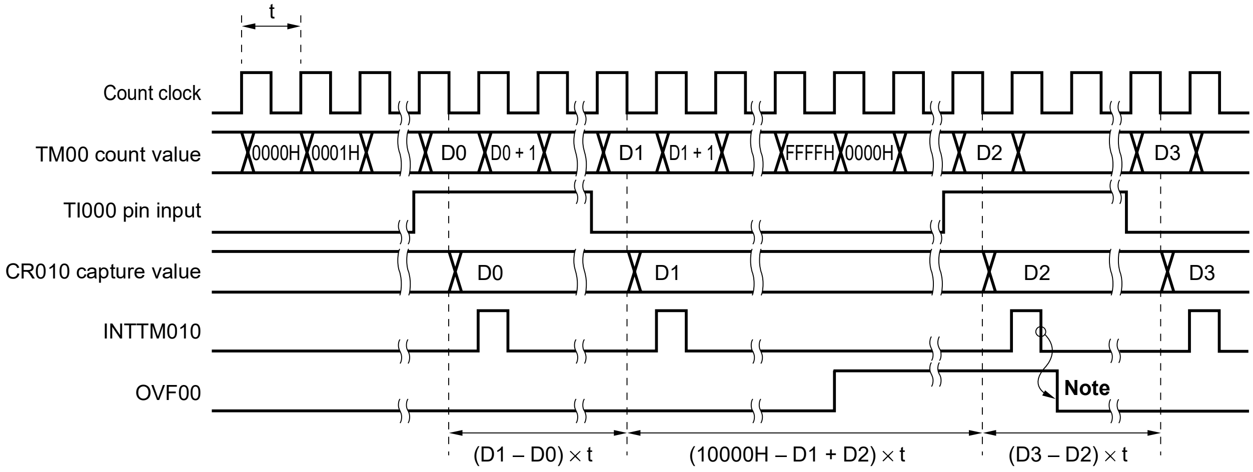

fx
fx/2²
fx/2⁸
Selector
16-bit timer counter 00 (TM00)
OVF00
16-bit timer capture/compare register 010 (CR010)
TI000
INTTM010
Internal bus

Figure 6-19. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



Note Clear OVF00 by software.

(2) Measurement of two pulse widths with free-running counter

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI000 pin and the TI010 pin.

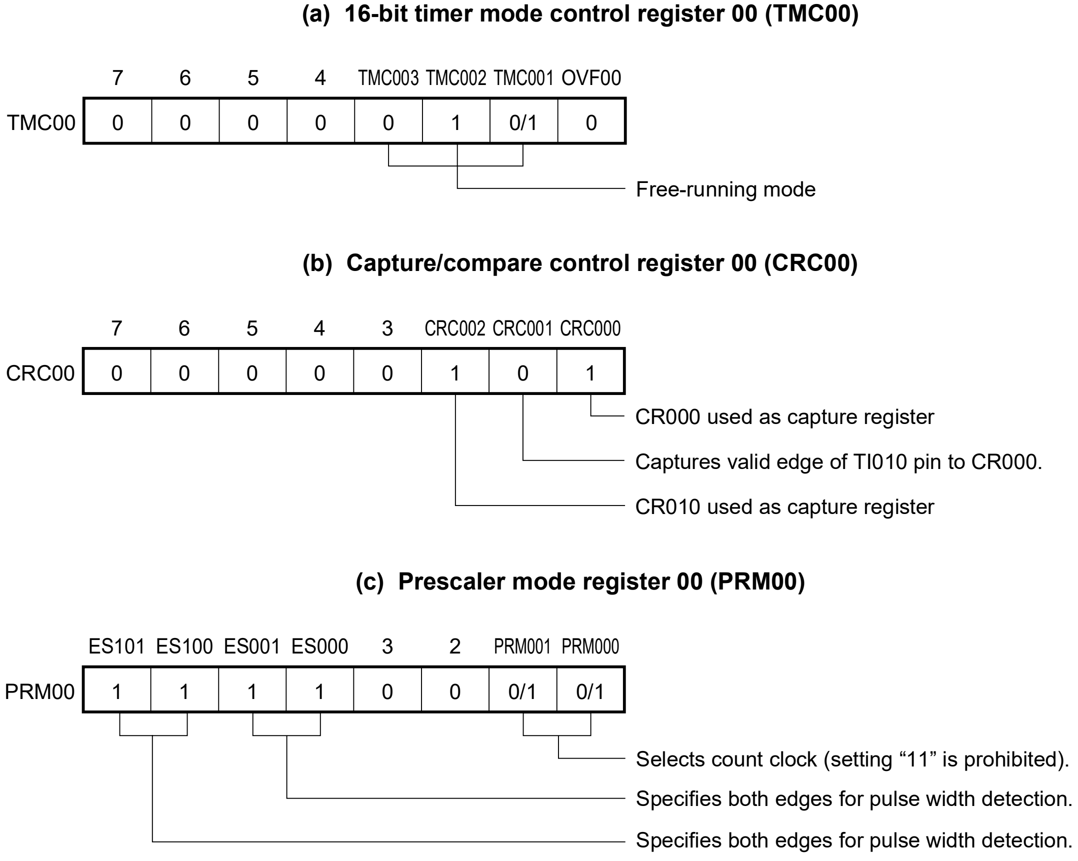
When the edge specified by bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an interrupt request signal (INTTM010) is set.

Also, when the edge specified by bits 6 and 7 (ES100 and ES101) of PRM00 is input to the TI010 pin, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000) and an interrupt request signal (INTTM000) is set.

Specify both the rising and falling edges as the edges of the TI000 and TI010 pins, by using bits 4 and 5 (ES000 and ES001) and bits 6 and 7 (ES100 and ES101) of PRM00.

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 or TI010 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-20. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	0	1

CR000 used as capture register

Captures valid edge of TI010 pin to CR000.

CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	1	1	1	1	0	0	0/1	0/1

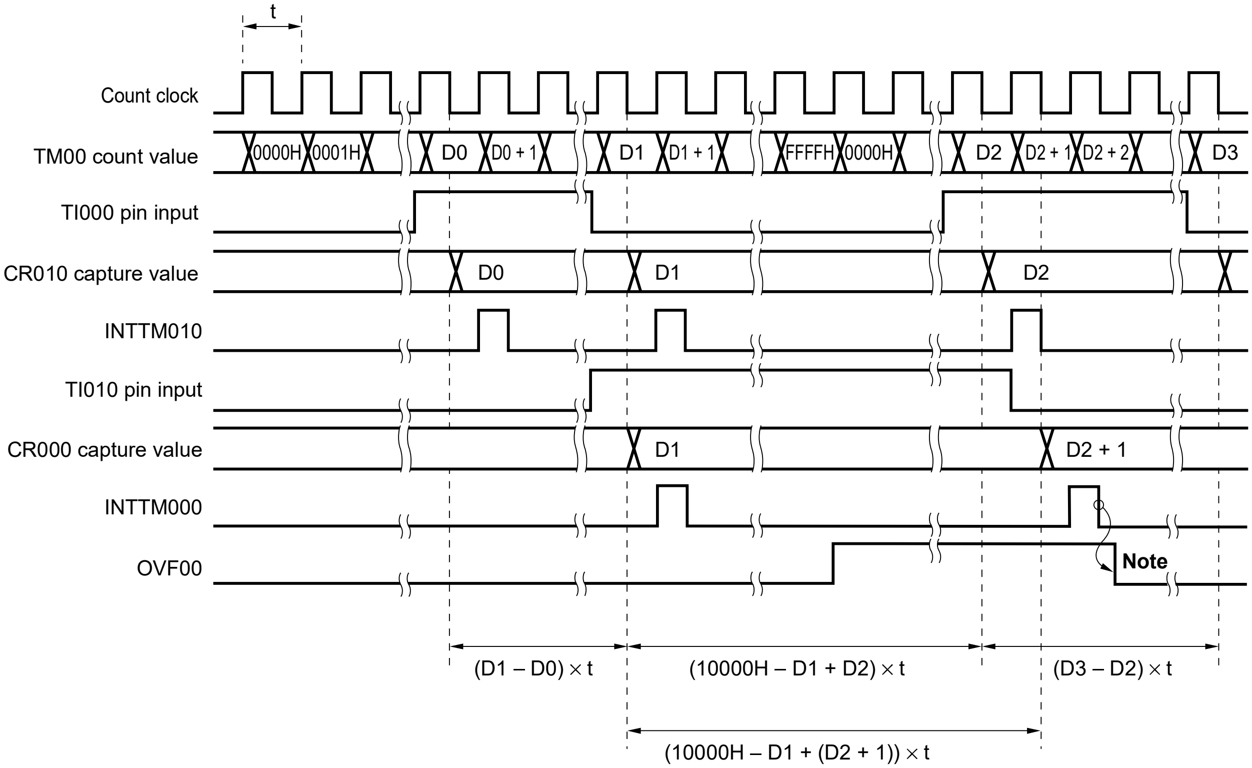
Selects count clock (setting “11” is prohibited).

Specifies both edges for pulse width detection.

Specifies both edges for pulse width detection.

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 6-21. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Note Clear OVF00 by software.

(3) Pulse width measurement with free-running counter and two capture registers

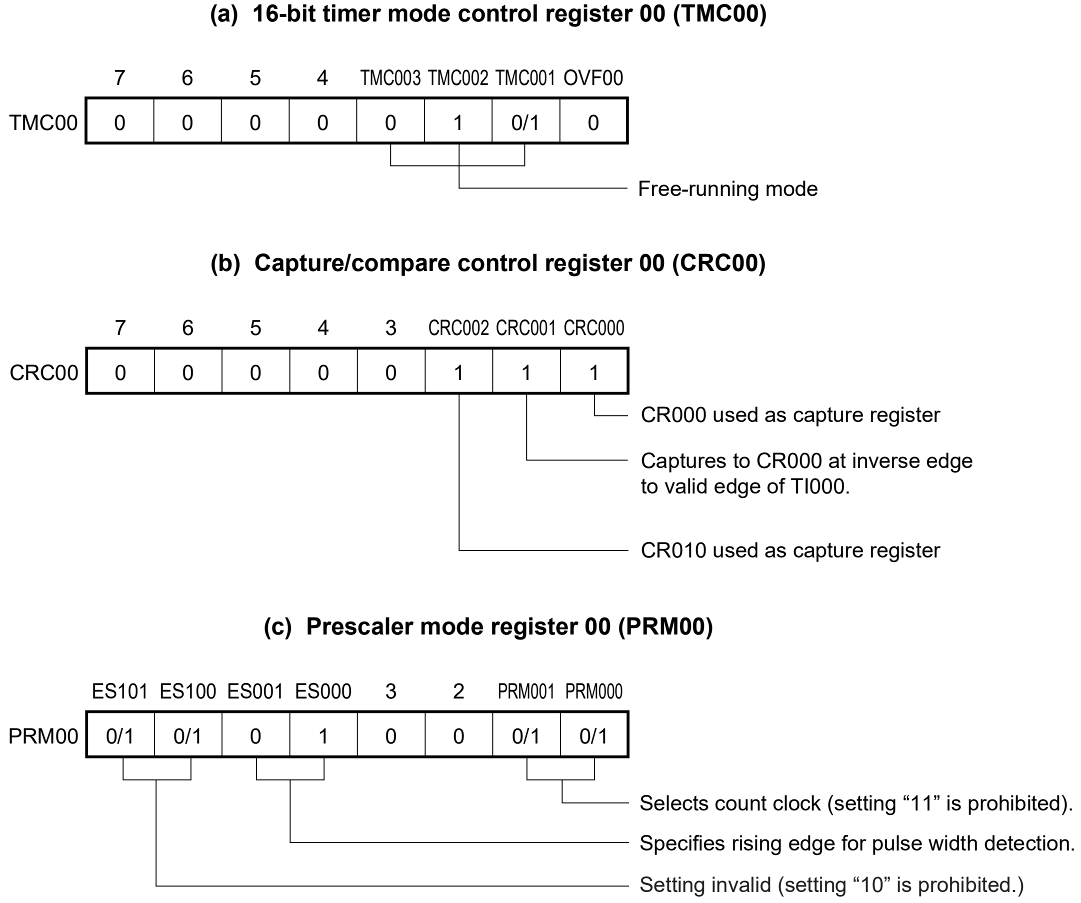
When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to measure the pulse width of the signal input to the TI000 pin.

When the edge specified by bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 010 (CR010) and an interrupt request signal (INTTM010) is set.

Also, when the inverse edge to that of the capture operation is input into CR010, the value of TM00 is taken into 16-bit timer capture/compare register 000 (CR000).

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-22. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	1	1

CR000 used as capture register

Captures to CR000 at inverse edge to valid edge of TI000.

CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

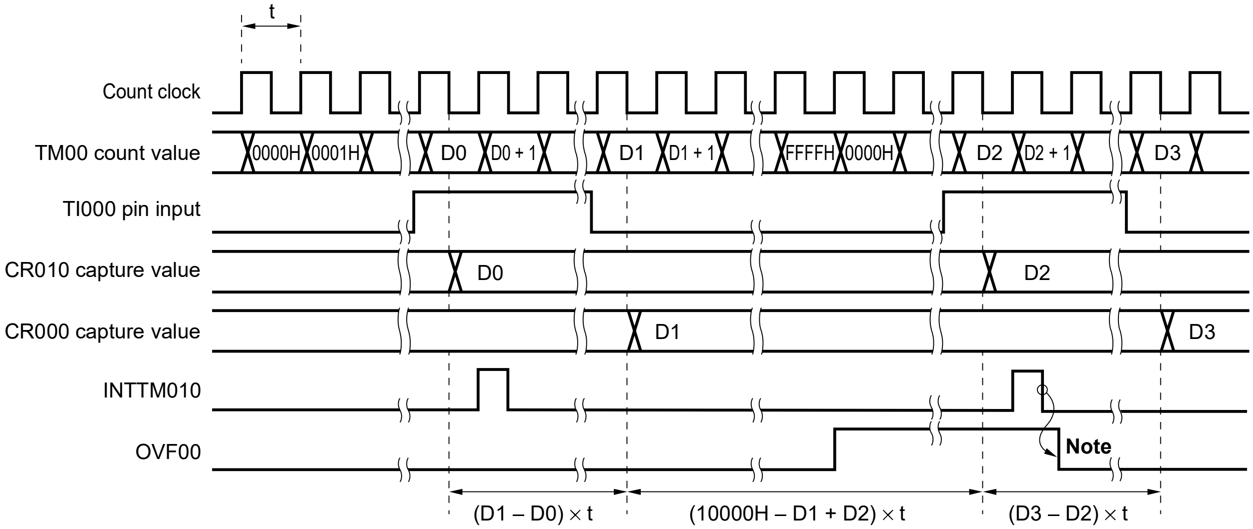
Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 6-23. Timing of Pulse Width Measurement Operation with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Note Clear OVF00 by software.

(4) Pulse width measurement by means of restart

When input of a valid edge to the TI000 pin is detected, the count value of 16-bit timer counter 00 (TM00) is taken into 16-bit timer capture/compare register 010 (CR010), and then the pulse width of the signal input to the TI000 pin is measured by clearing TM00 and restarting the count operation.

Either of two edges—rising or falling—can be selected using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00) and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-24. Control Register Settings for Pulse Width Measurement by Means of Restart (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	0	0/1	0

Clears and starts at valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	1	1

CR000 used as capture register

Captures to CR000 at inverse edge to valid edge of TI000.

CR010 used as capture register

(c) Prescaler mode register 00 (PRM00)

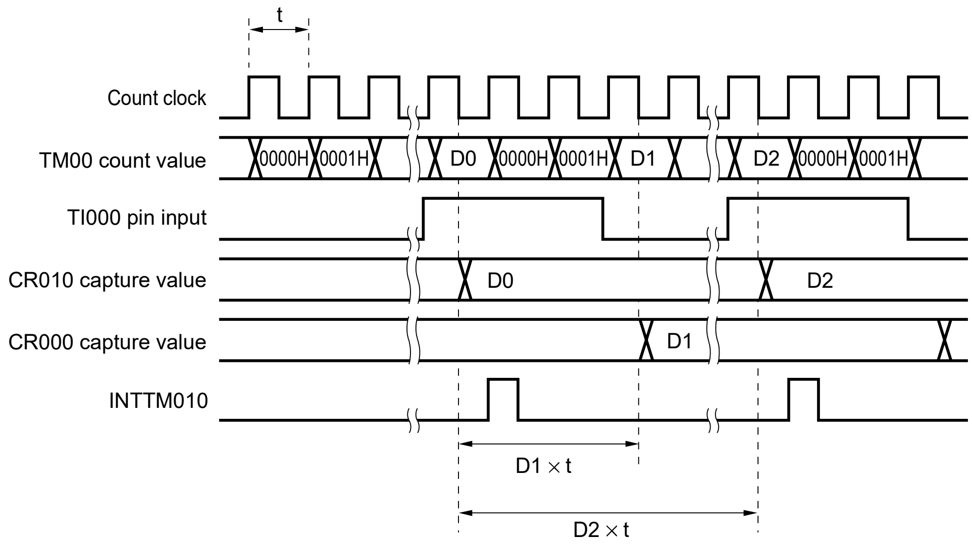
	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Figure 6-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



6.4.4 External event counter operation

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (refer to **Figure 6-26** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set any value to the CR000 register (0000H cannot be set).
<4> Set the TMC00 register to start the operation (refer to **Figure 6-26** for the set value).

Remarks 1. For the setting of the TI000 pin, refer to **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

The external event counter counts the number of external clock pulses input to the TI000 pin using 16-bit timer counter 00 (TM00).

TM00 is incremented each time the valid edge specified by prescaler mode register 00 (PRM00) is input.

When the TM00 count value matches the 16-bit timer capture/compare register 000 (CR000) value, TM00 is cleared to 0 and the interrupt request signal (INTTM000) is generated.

Input a value other than 0000H to CR000 (a count operation with 1-bit pulse cannot be carried out).

Any of three edges—rising, falling, or both edges—can be selected using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

Sampling is performed using the internal clock (fx) and an operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 6-26. Control Register Settings in External Event Counter Mode (with Rising Edge Specified) (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM00 and CR000.

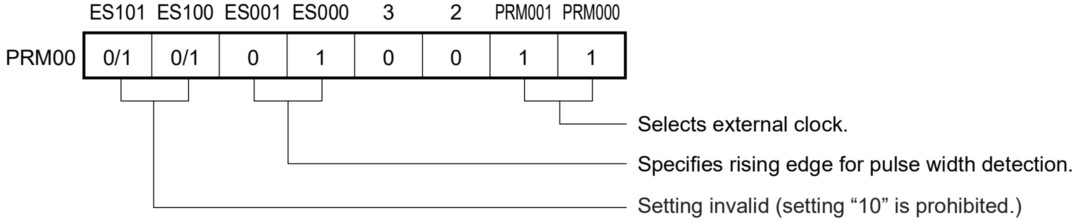
(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR000 used as compare register

Figure 6-26. Control Register Settings in External Event Counter Mode (with Rising Edge Specified) (2/2)

(c) Prescaler mode register 00 (PRM00)

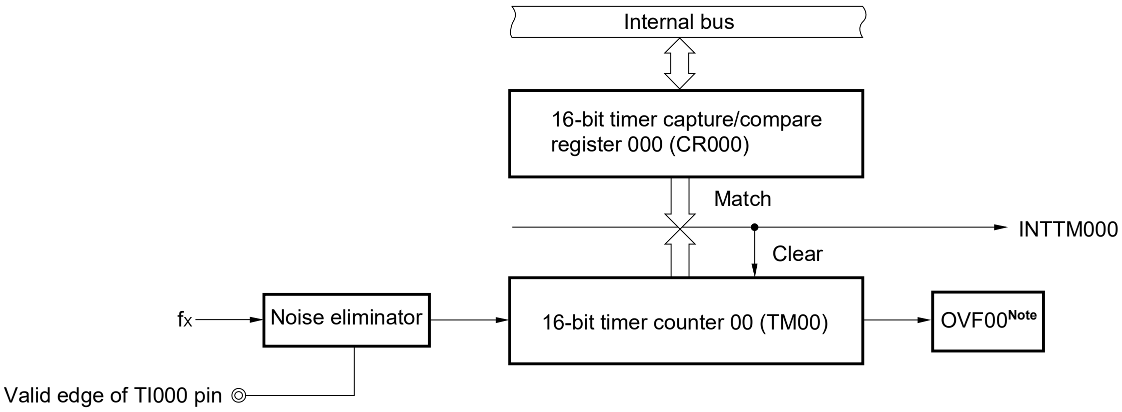


	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	1	1

Selects external clock.
Specifies rising edge for pulse width detection.
Setting invalid (setting “10” is prohibited.)

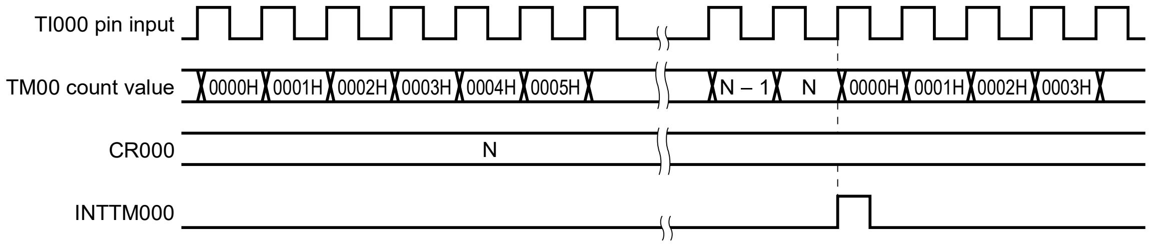
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

Figure 6-27. Configuration Diagram of External Event Counter



Note OVF00 is set to 1 only when CR000 is set to FFFFH.

Figure 6-28. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM00 should be read.

6.4.5 Square-wave output operation

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (refer to **Figure 6-29** for the set value).
<3> Set the TOC00 register (refer to **Figure 6-29** for the set value).
<4> Set any value to the CR000 register (0000H cannot be set).
<5> Set the TMC00 register to start the operation (refer to **Figure 6-29** for the set value).

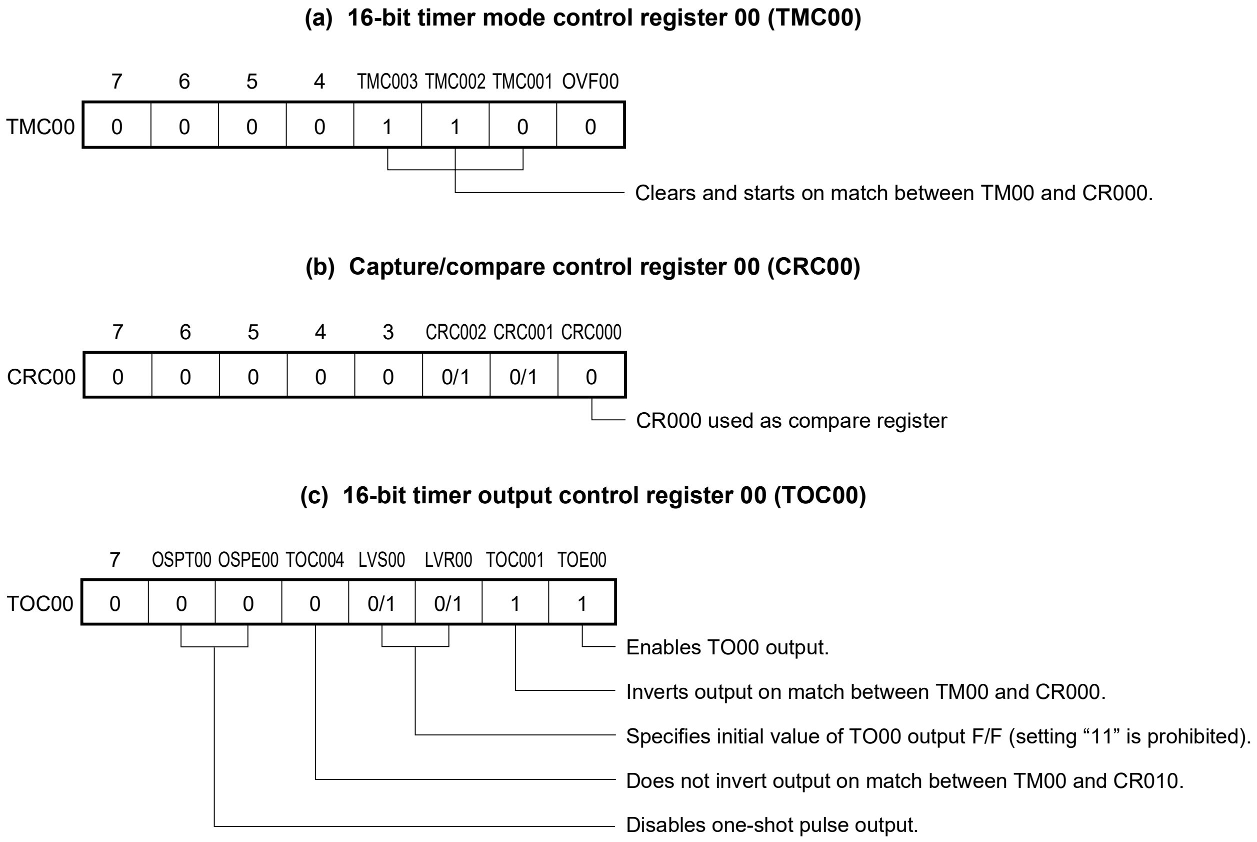
Caution CR000 cannot be rewritten during TM00 operation.

Remarks 1. For the setting of the TO00 pin, refer to **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

A square wave with any selected frequency can be output at intervals determined by the count value preset to 16-bit timer capture/compare register 000 (CR000).

The TO00 pin output status is reversed at intervals determined by the count value preset to CR000 + 1 by setting bit 0 (TOE00) and bit 1 (TOC001) of 16-bit timer output control register 00 (TOC00) to 1. This enables a square wave with any selected frequency to be output.

Figure 6-29. Control Register Settings in Square-Wave Output Mode (1/2)



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0	0

Clears and starts on match between TM00 and CR000.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR000 used as compare register

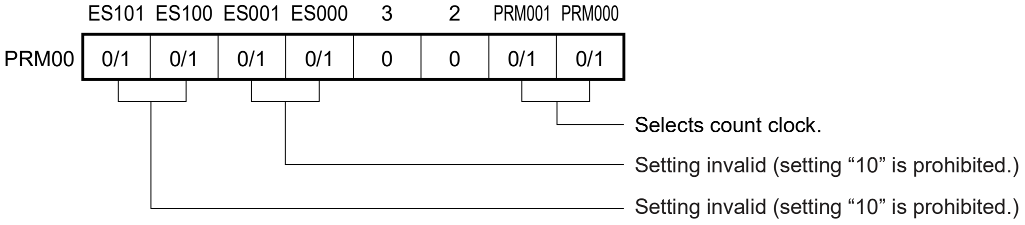
(c) 16-bit timer output control register 00 (TOC00)

	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	0	0	0/1	0/1	1	1

Enables TO00 output.
Inverts output on match between TM00 and CR000.
Specifies initial value of TO00 output F/F (setting “11” is prohibited).
Does not invert output on match between TM00 and CR010.
Disables one-shot pulse output.

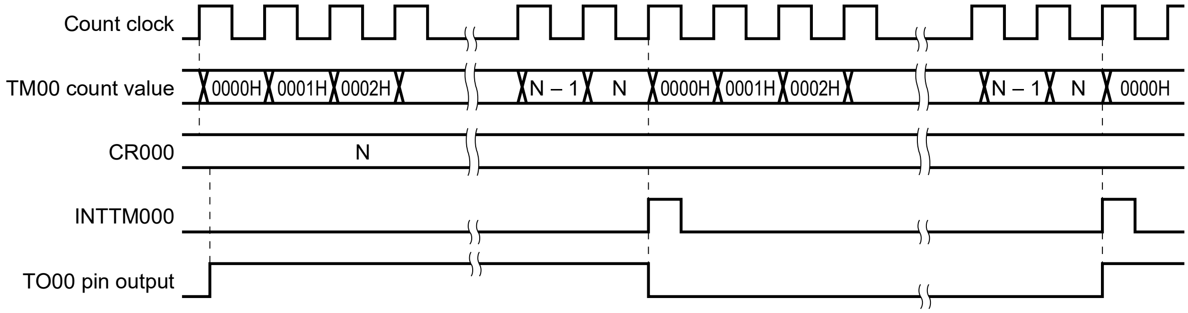
Figure 6-29. Control Register Settings in Square-Wave Output Mode (2/2)

(d) Prescaler mode register 00 (PRM00)


ES101
ES100
ES001
ES000
3
2
PRM001
PRM000
PRM00
0/1
0/1
0/1
0/1
0
0
0/1
0/1
Selects count clock.
Setting invalid (setting “10” is prohibited.)
Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the description of the respective control registers for details.

Figure 6-30. Square-Wave Output Operation Timing



6.4.6 One-shot pulse output operation

16-bit timer/event counter 00 can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI000 pin input).

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (refer to **Figures 6-31** and **6-33** for the set value).
<3> Set the TOC00 register (refer to **Figures 6-31** and **6-33** for the set value).
<4> Set any value to the CR000 and CR010 registers (0000H cannot be set).
<5> Set the TMC00 register to start the operation (refer to **Figures 6-31** and **6-33** for the set value).

Remarks 1. For the setting of the TO00 pin, refer to **6.3 (5) Port mode register 0 (PM0)**.
2. For how to enable the INTTM000 (if necessary, INTTM010) interrupt, refer to **CHAPTER 16 INTERRUPT FUNCTIONS**.

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 6-31, and by setting bit 6 (OSPT00) of the TOC00 register to 1 by software.

By setting the OSPT00 bit to 1, 16-bit timer/event counter 00 is cleared and started, and its output becomes active at the count value (N) set in advance to 16-bit timer capture/compare register 010 (CR010). After that, the output becomes inactive at the count value (M) set in advance to 16-bit timer capture/compare register 000 (CR000)[Note].

Even after the one-shot pulse has been output, the TM00 register continues its operation. To stop the TM00 register, the TMC003 and TMC002 bits of the TMC00 register must be cleared to 00.

Note The case where N < M is described here. When N > M, the output becomes active with the CR000 register and inactive with the CR010 register. Do not set N to M.

Cautions 1. Do not set the OSPT00 bit to 1 while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed.
2. When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate-function port pin.
Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

Figure 6-31. Control Register Settings for One-Shot Pulse Output with Software Trigger

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0	0/1	0

CR000 used as compare register

CR010 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	1	1	0/1	0/1	1	1

Enables TO00 output.

Inverts output upon match between TM00 and CR000.

Specifies initial value of TO00 output F/F (setting “11” is prohibited).

Inverts output upon match between TM00 and CR010.

Sets one-shot pulse output mode.

Set to 1 for output.

(d) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0/1	0/1	0	0	0/1	0/1

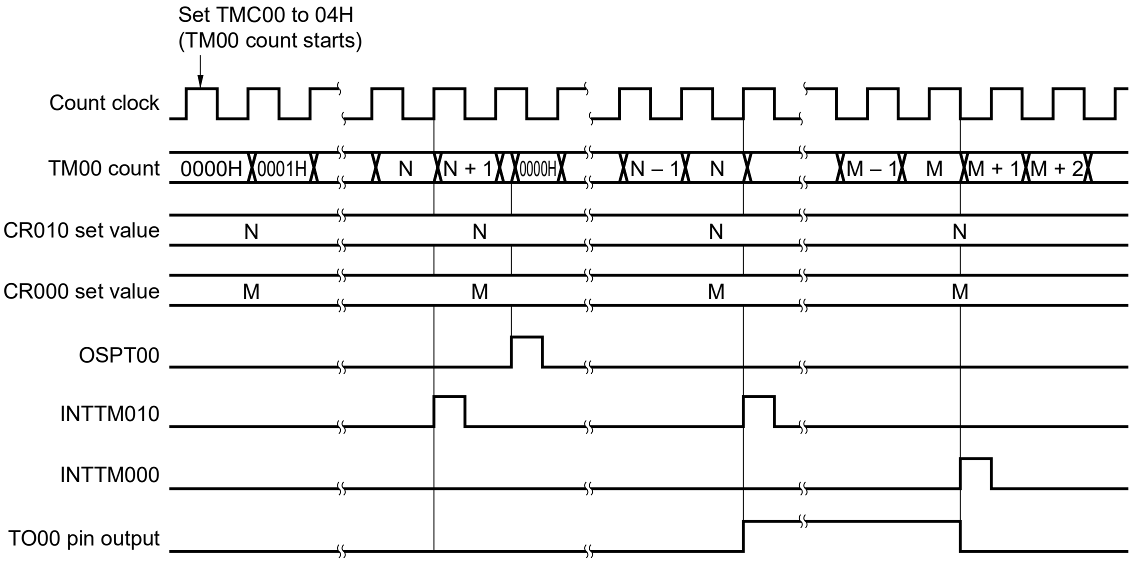
Selects count clock.

Setting invalid
(setting “10” is prohibited.)

Setting invalid
(setting “10” is prohibited.)

Caution Do not clear the CR000 and CR010 registers to 0000H.

Figure 6-32. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution 16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC003 and TMC002 bits.

Remark N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 6-33, and by using the valid edge of the TI000 pin as an external trigger.

The valid edge of the TI000 pin is specified by bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00). The rising, falling, or both the rising and falling edges can be specified.

When the valid edge of the TI000 pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 010 (CR010). After that, the output becomes inactive at the count value set in advance to 16-bit timer capture/compare register 000 (CR000)[Note].

Note The case where N < M is described here. When N > M, the output becomes active with the CR000 register and inactive with the CR010 register. Do not set N to M.

Caution Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.

Figure 6-33. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	0	0	0

Clears and starts at valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0	0/1	0

CR000 used as compare register

CR010 used as compare register

(c) 16-bit timer output control register 00 (TOC00)

	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	1	1	0/1	0/1	1	1

Enables TO00 output.

Inverts output upon match between TM00 and CR000.

Specifies initial value of TO00 output F/F (setting “11” is prohibited).

Inverts output upon match between TM00 and CR010.

Sets one-shot pulse output mode.

(d) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

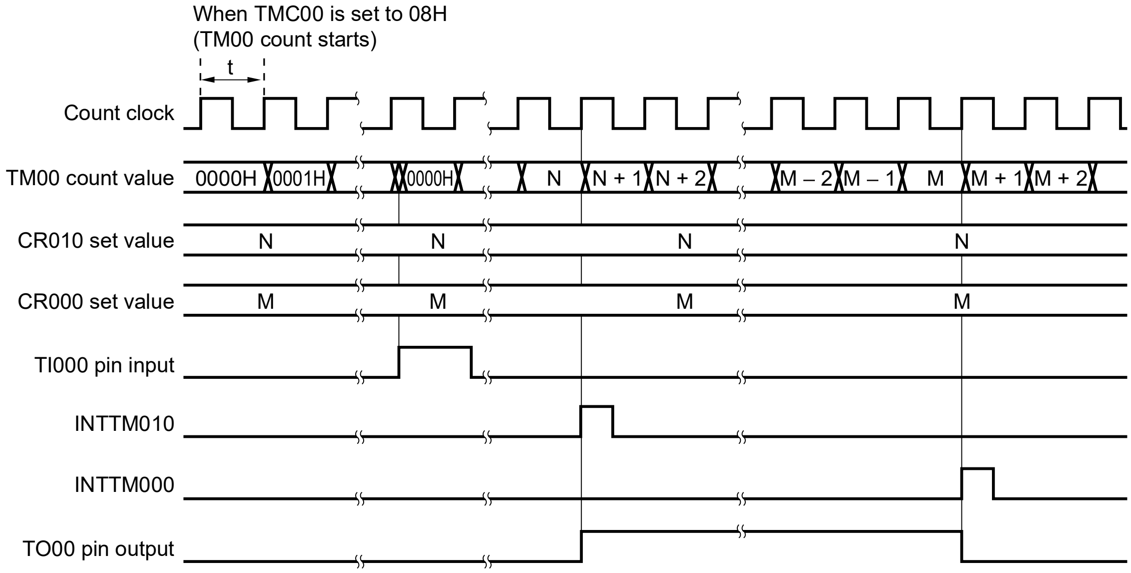
Selects count clock (setting “11” is prohibited).

Specifies the rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Caution Do not clear the CR000 and CR010 registers to 0000H.

Figure 6-34. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution **16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC003 and TMC002 bits.**

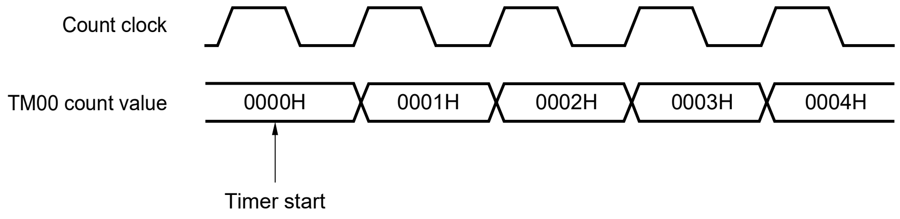
Remark N < M

6.5 Cautions for 16-Bit Timer/Event Counter 00

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 00 (TM00) is started asynchronously to the count clock.

Figure 6-35. Start Timing of 16-Bit Timer Counter 00 (TM00)



(2) 16-bit timer capture/compare register 000 setting

In the mode in which clear & start occurs on a match between TM00 and CR000, set 16-bit timer capture/compare register 000 (CR000) to other than 0000H. This means a 1-pulse count operation cannot be performed when 16-bit timer/event counter 00 is used as an external event counter.

(3) Capture register data retention

The values of 16-bit timer capture/compare registers 000 and 010 (CR000 and CR010) are not guaranteed after 16-bit timer/event counter 00 has been stopped.

(4) Valid edge setting

Set the valid edge of the TI000 pin after clearing bits 2 and 3 (TMC002 and TMC003) of 16-bit timer mode control register 00 (TMC00) to 0, 0, respectively, and then stopping timer operation. The valid edge is set using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

(5) Re-triggering one-shot pulse

(a) One-shot pulse output by software

When a one-shot pulse is output, do not set the OSPT00 bit to 1. Do not output the one-shot pulse again until INTTM000, which occurs upon a match with the CR000 register, or INTTM010, which occurs upon a match with the CR010 register, occurs.

(b) One-shot pulse output with external trigger

If the external trigger occurs again while a one-shot pulse is output, it is ignored.

(c) One-shot pulse output function

When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate-function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

(6) Operation of OVF00 flag

<1> The OVF00 flag is also set to 1 in the following case.

When any of the following modes is selected: the mode in which clear & start occurs on a match between TM00 and CR000, the mode in which clear & start occurs at the TI000 pin valid edge, or the free-running mode

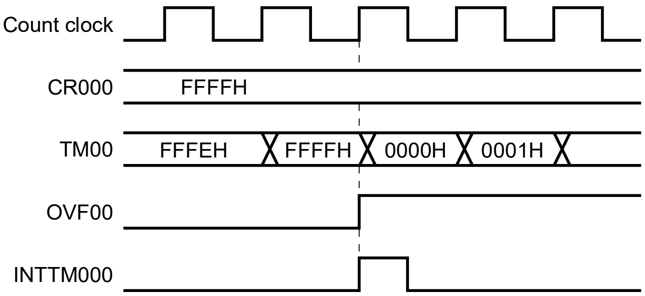
↓

CR000 is set to FFFFH

↓

TM00 is counted up from FFFFH to 0000H.

Figure 6-36. Operation Timing of OVF00 Flag



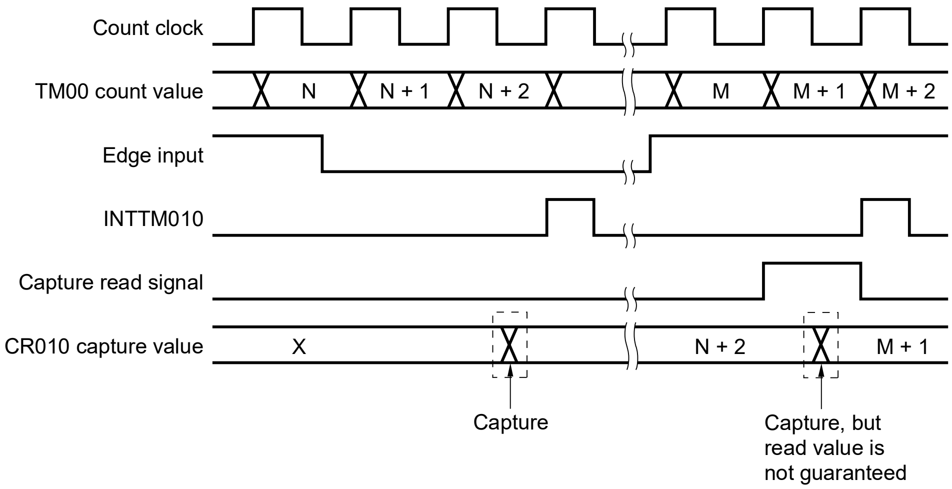
<2> Even if the OVF00 flag is cleared before the next count clock is counted (before TM00 becomes 0001H) after the occurrence of TM00 overflow, the OVF00 flag is re-set newly and clear is disabled.

(7) Conflicting operations

Conflict between the read period of the 16-bit timer capture/compare register (CR000/CR010) and capture trigger input (CR000/CR010 used as capture register)

Capture trigger input has priority. The data read from CR000/CR010 is undefined.

Figure 6-37. Capture Register Data Retention Timing



(8) Timer operation

<1> Even if 16-bit timer counter 00 (TM00) is read, the value is not captured by 16-bit timer capture/compare register 010 (CR010).

<2> Regardless of the CPU's operation mode, when the timer stops, the input signals to the TI000/TI010 pins are not acknowledged.

<3> The one-shot pulse output mode operates correctly only in the free-running mode and the mode in which clear & start occurs at the TI000 valid edge. In the mode in which clear & start occurs on a match between the TM00 register and CR000 register, one-shot pulse output is not possible because an overflow does not occur.

(9) Capture operation

<1> If the TI000 pin valid edge is specified as the count clock, a capture operation by the capture register specified as the trigger for the TI000 pin is not possible.

<2> To ensure the reliability of the capture operation, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

<3> The capture operation is performed at the falling edge of the count clock. An interrupt request input (INTTM000/INTTM010), however, is generated at the rise of the next count clock.

(10) Compare operation

A capture operation may not be performed for CR000/CR010 set in compare mode even if a capture trigger has been input.

(11) Edge detection

<1> If the TI000 or TI010 pin is high level immediately after system reset and the rising edge or both the rising and falling edges are specified as the valid edge of the TI000 or TI010 pin to enable the 16-bit timer counter 00 (TM00) operation, a rising edge is detected immediately after the operation is enabled. Be careful therefore when pulling up the TI000 or TI010 pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to eliminate noise differs when the TI000 pin valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the count clock is f_X, and in the latter case the count clock is selected by prescaler mode register 00 (PRM00). The capture operation is only performed when a valid level is detected twice by sampling the valid edge, thus eliminating noise with a short pulse width.

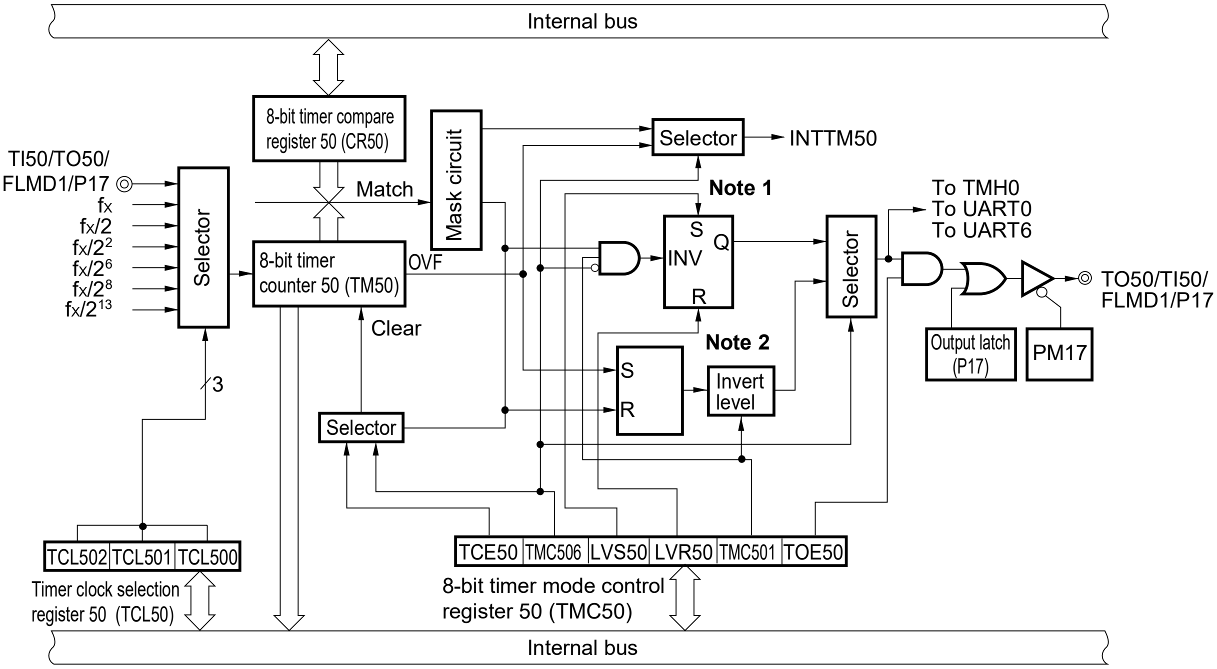
7.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

Figures 7-1 and 7-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

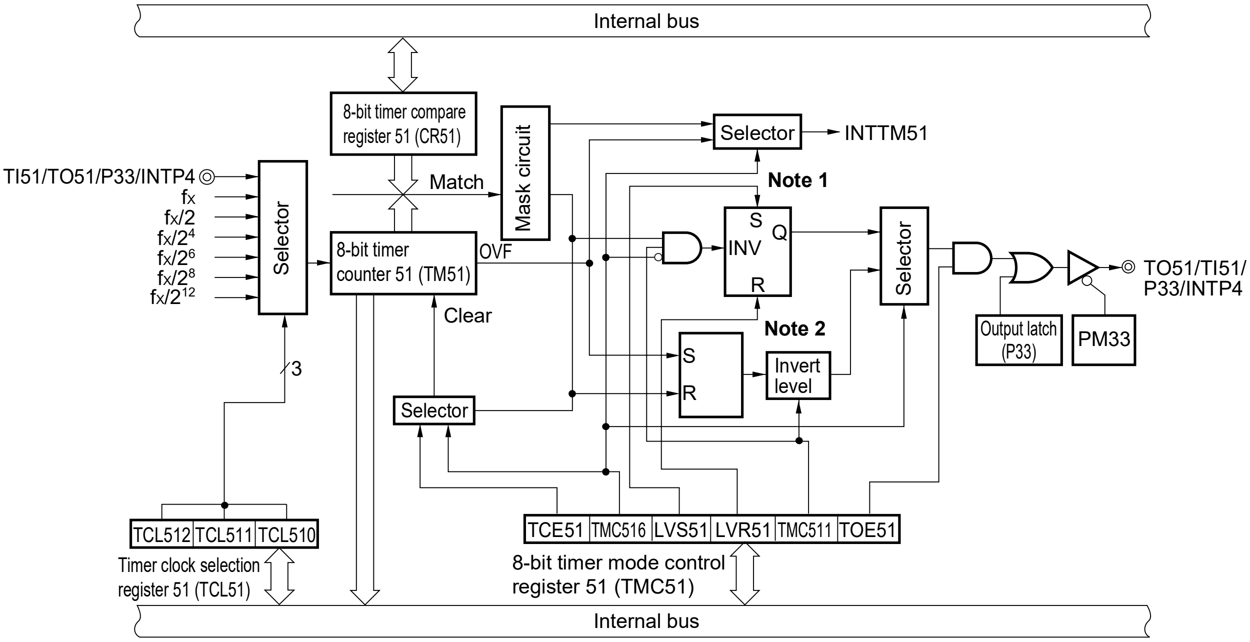
Figure 7-1. Block Diagram of 8-Bit Timer/Event Counter 50



Notes 1. Timer output F/F

2. PWM output F/F

Figure 7-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

7.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 include the following hardware.

Table 7-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n
Timer output	TO5n
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 1 (PM1) or port mode register 3 (PM3) Port register 1 (P1) or port register 3 (P3)

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51)   After reset: 00H   R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> $\overline{\text{RESET}}$ input

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

Remark n = 0, 1

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In PWM mode, when the TO5n pin becomes active due to a TM5n overflow and the values of TM5n and CR5n match, the TO5n pin becomes inactive.

The value of CR5n can be set within 00H to FFH.

$\overline{\text{RESET}}$ input clears CR5n to 00H.

Figure 7-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 1 (PM1) or port mode register 3 (PM3)
- Port register 1 (P1) or port register 3 (P3)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of TI5n pin input.

TCL5n can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TCL5n to 00H.

Remark n = 0, 1

Figure 7-5. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection[Note]
0	0	0	TI50 pin falling edge
0	0	1	TI50 pin rising edge
0	1	0	fx (10 MHz)
0	1	1	fx/2 (5 MHz)
1	0	0	fx/2^2 (2.5 MHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^8 (39.06 kHz)
1	1	1	fx/2^{13} (1.22 kHz)

Note Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Count clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Count clock ≤ 5 MHz

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the count clock is the Ring-OSC clock, the operation of 8-bit timer/event counter 50 is not guaranteed.

2. When rewriting TCL50 to other data, stop the timer operation beforehand.

3. Be sure to clear bits 3 to 7 to 0.

Remarks 1. fx: High-speed system clock oscillation frequency

2. Figures in parentheses apply to operation at fx = 10 MHz.

Figure 7-6. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection[Note]
0	0	0	TI51 falling edge
0	0	1	TI51 rising edge
0	1	0	fx (10 MHz)
0	1	1	fx/2 (5 MHz)
1	0	0	fx/2^4 (625 kHz)
1	0	1	fx/2^6 (156.25 kHz)
1	1	0	fx/2^8 (39.06 kHz)
1	1	1	fx/2^{12} (2.44 kHz)

Note Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Count clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Count clock ≤ 5 MHz

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the count clock is the Ring-OSC clock, the operation of 8-bit timer/event counter 51 is not guaranteed.

2. When rewriting TCL51 to other data, stop the timer operation beforehand.

3. Be sure to clear bits 3 to 7 to 0.

Remarks 1. fx: High-speed system clock oscillation frequency

2. Figures in parentheses apply to operation at fx = 10 MHz.

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip-flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMC5n to 00H.

Remark n = 0, 1

Figure 7-7. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF6BH After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TM50 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

(Refer to **Caution** and **Remark** on the next page.)

Figure 7-8. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF43H After reset: 00H R/W[Note]

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TM51 output is low level)
1	Output enabled

Note Bits 2 and 3 are write-only.

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Perform <1> to <4> below in the following order, not at the same time.

<1> Set TMC5n1, TMC5n6: Operation mode setting

<2> Set TOE5n to enable output: Timer output enable

<3> Set LVS5n, LVR5n (refer to Caution 1): Timer F/F setting

<4> Set TCE5n

3. Stop operation before rewriting TMC5n6.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n pin regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode registers 1 and 3 (PM1, PM3)

These registers set port 1 and 3 input/output in 1-bit units.

When using the P17/TO50/TI50/FLMD1 and P33/TO51/TI51/INTP4 pins for timer output, clear PM17 and PM33 and the output latches of P17 and P33 to 0.

When using the P17/TO50/TI50/FLMD1 and P33/TO51/TI51/INTP4 pins for timer input, set PM17 and PM33 to 1. The output latches of P17 and P33 at this time may be 0 or 1.

PM1 and PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM1 and PM3 to FFH.

Figure 7-9. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 7-10. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of 8-Bit Timer/Event Counters 50 and 51

7.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set each register.

- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don’t care)

<2> After TCE5n = 1 is set, the count operation starts.

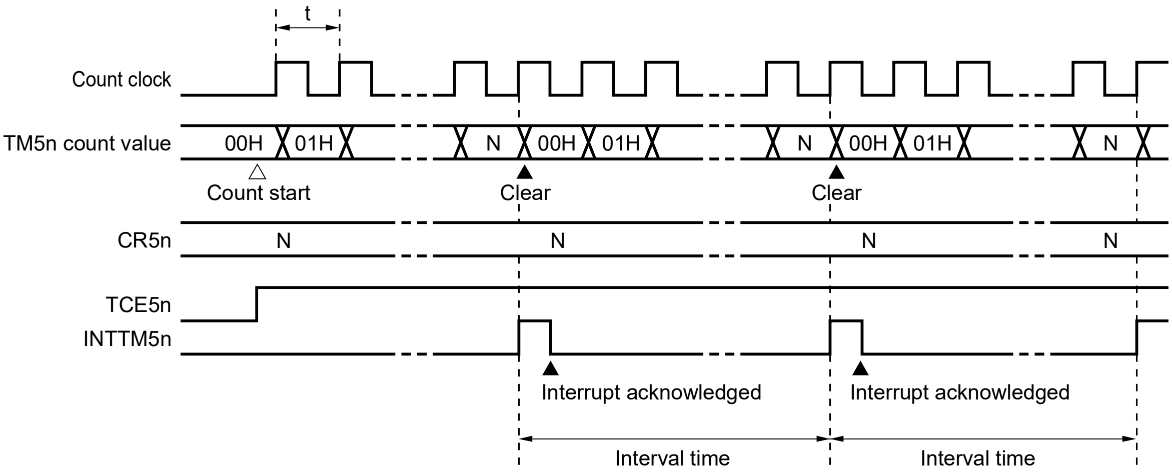
<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> INTTM5n is generated repeatedly at the same interval. Clear TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Figure 7-11. Interval Timer Operation Timing (1/2)

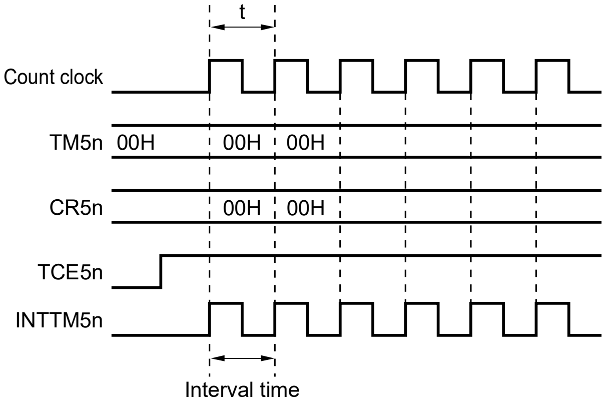
(a) Basic operation



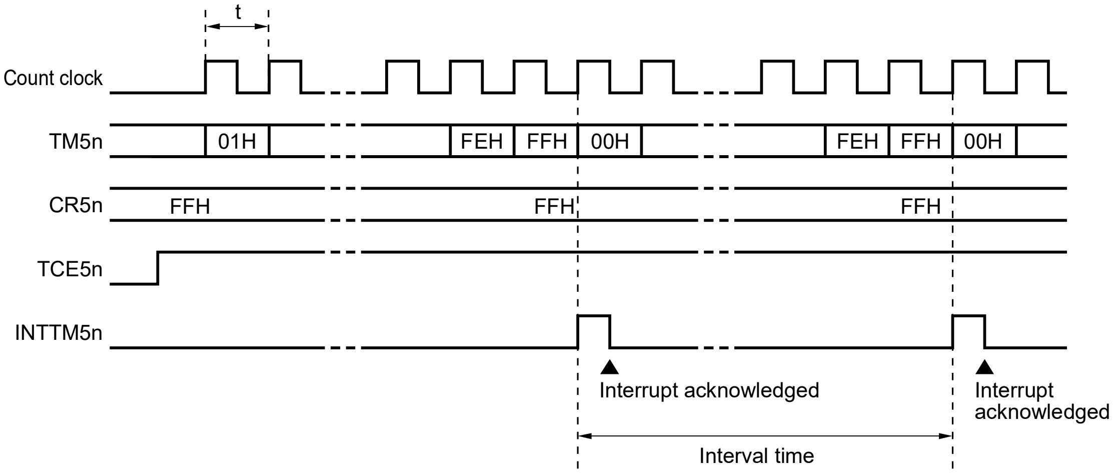
Remark Interval time = (N + 1) × t
N = 00H to FFH
n = 0, 1

Figure 7-11. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

7.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI5n pin by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

Setting

<1> Set each register.

- Set the port mode register (PM17 or PM33)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B × = Don’t care)

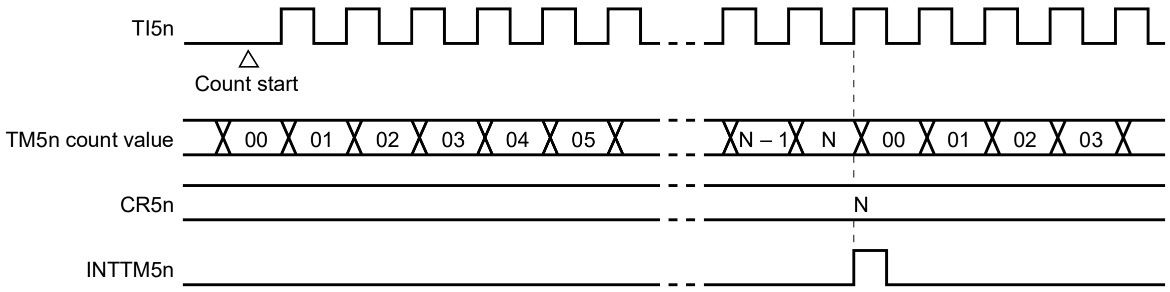
<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM17
8-bit timer/event counter 51: PM33

Figure 7-12. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

7.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.

The frequency is as follows.

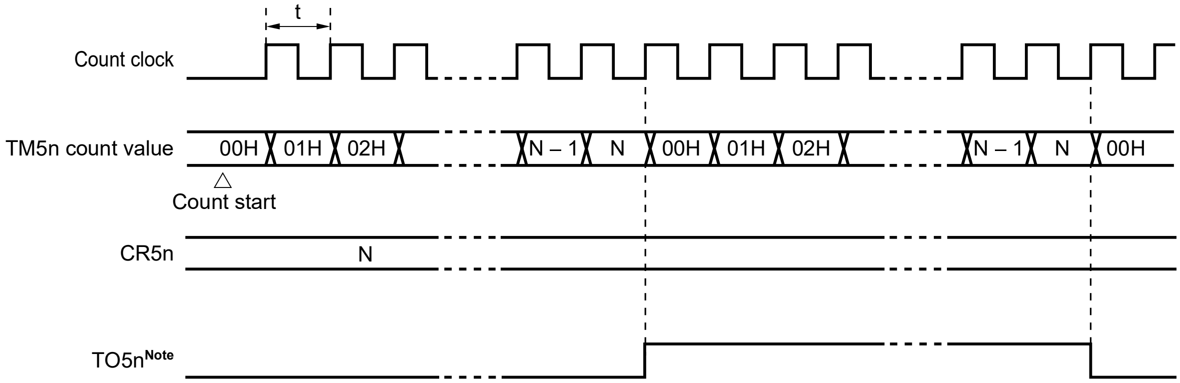
Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 7-13. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

7.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P17 or P33)[Note] and port mode register (PM17 or PM33)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P17, PM17
8-bit timer/event counter 51: P33, PM33

PWM output operation

<1> PWM output (output from TO5n) outputs an inactive level until an overflow occurs.

<2> When an overflow occurs, the active level is output. The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.

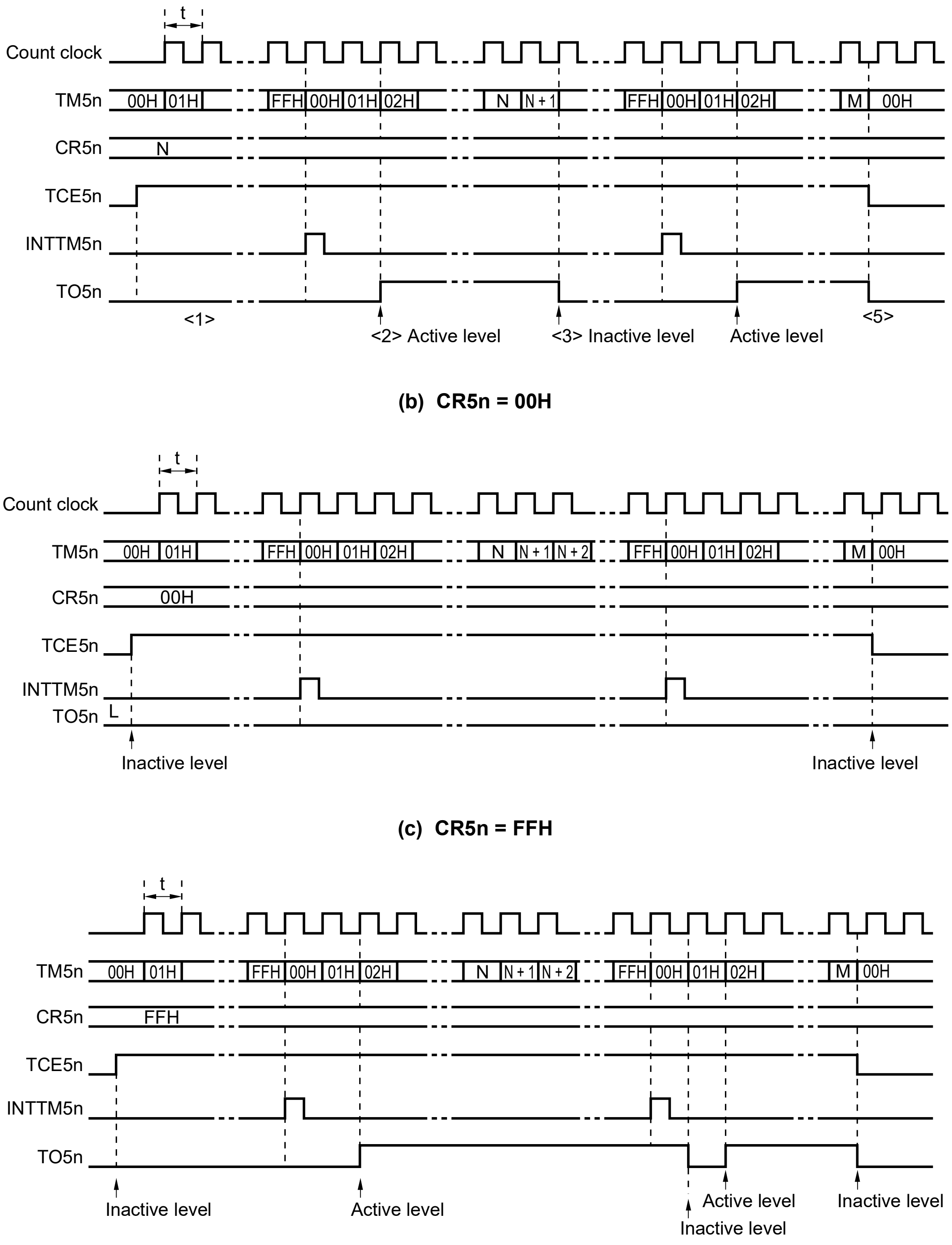
<4> Operations <2> and <3> are repeated until the count operation stops.

<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, refer to **Figures 7-14** and **7-15**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = N/2^8
  (N = 00H to FFH)

Remark n = 0, 1

Figure 7-14. PWM Output Operation Timing



(a) Basic operation (active level = H)

(b) CR5n = 00H

(c) CR5n = FFH

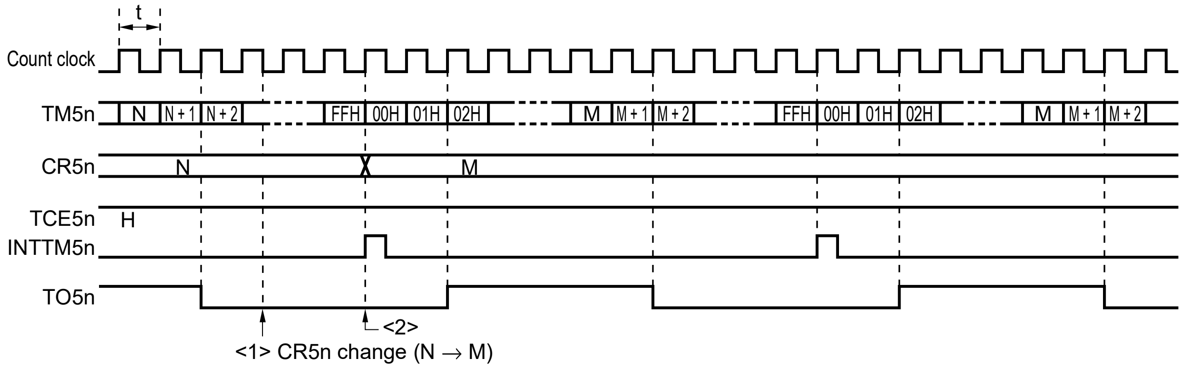
Remarks 1. <1> to <3> and <5> in Figure 7-14 (a) correspond to <1> to <3> and <5> in PWM output operation in **7.4.4 (1) PWM output basic operation**.

2. n = 0, 1

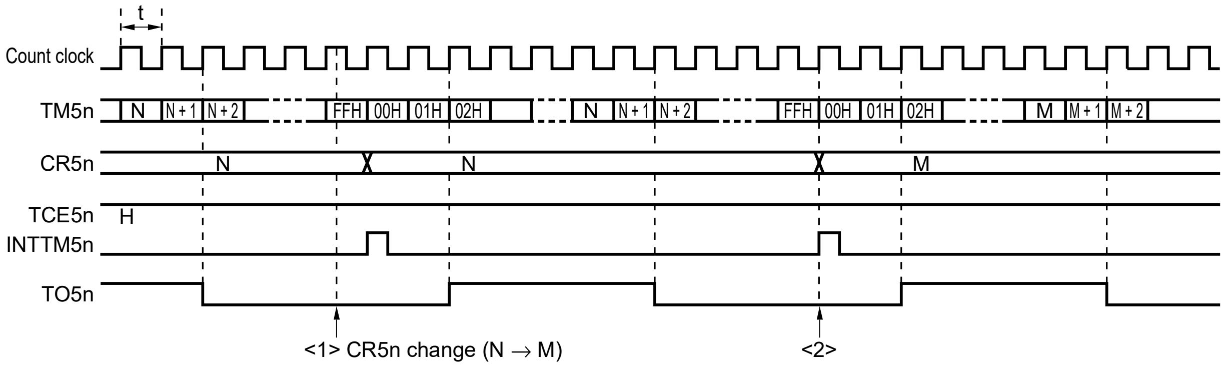
(2) Operation with CR5n changed

Figure 7-15. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



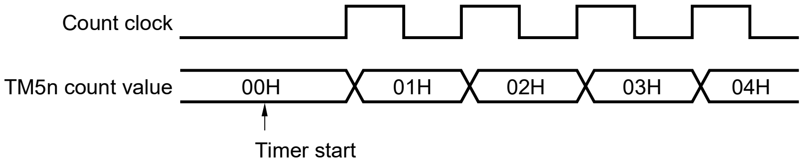
Caution When reading from CR5n between <1> and <2> in Figure 7-15, the value read differs from the actual value (read value: M, actual value of CR5n: N).

7.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

Figure 7-16. 8-Bit Timer Counter 5n Start Timing



Remark n = 0, 1

CHAPTER 8 8-BIT TIMERS H0 AND H1

8.1 Functions of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 have the following functions.

- Interval timer
- PWM output mode
- Square-wave output
- Carrier generator mode (8-bit timer H1 only)

8.2 Configuration of 8-Bit Timers H0 and H1

8-bit timers H0 and H1 include the following hardware.

Table 8-1. Configuration of 8-Bit Timers H0 and H1

Item	Configuration
Timer register	8-bit timer counter Hn
Registers	8-bit timer H compare register 0n (CMP0n) 8-bit timer H compare register 1n (CMP1n)
Timer output	TOHn
Control registers	8-bit timer H mode register n (TMHMDn) 8-bit timer H carrier control register 1 (TMCYC1)[Note] Port mode register 1 (PM1) Port register 1 (P1)

Note 8-bit timer H1 only

Remark n = 0, 1

Figures 8-1 and 8-2 show the block diagrams.

Figure 8-1. Block Diagram of 8-Bit Timer H0

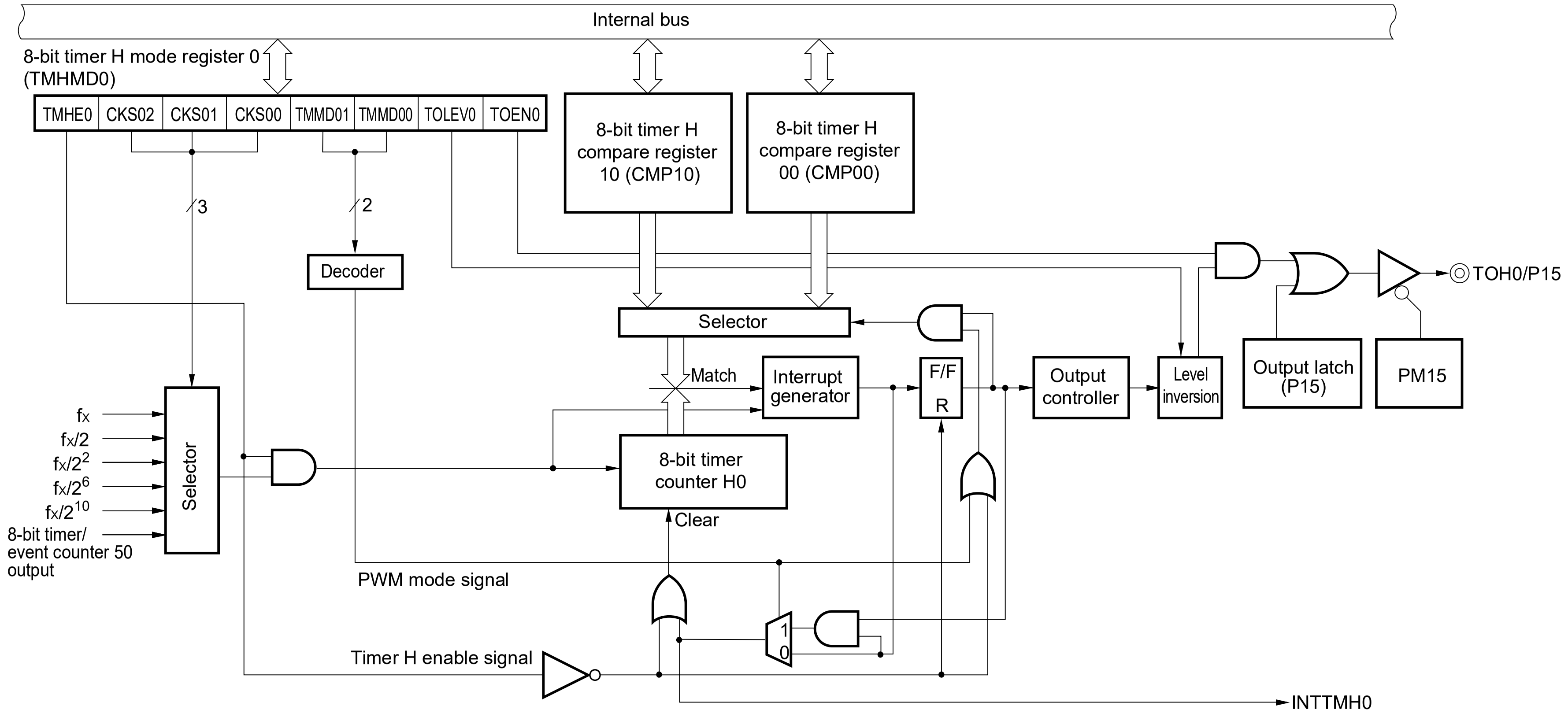
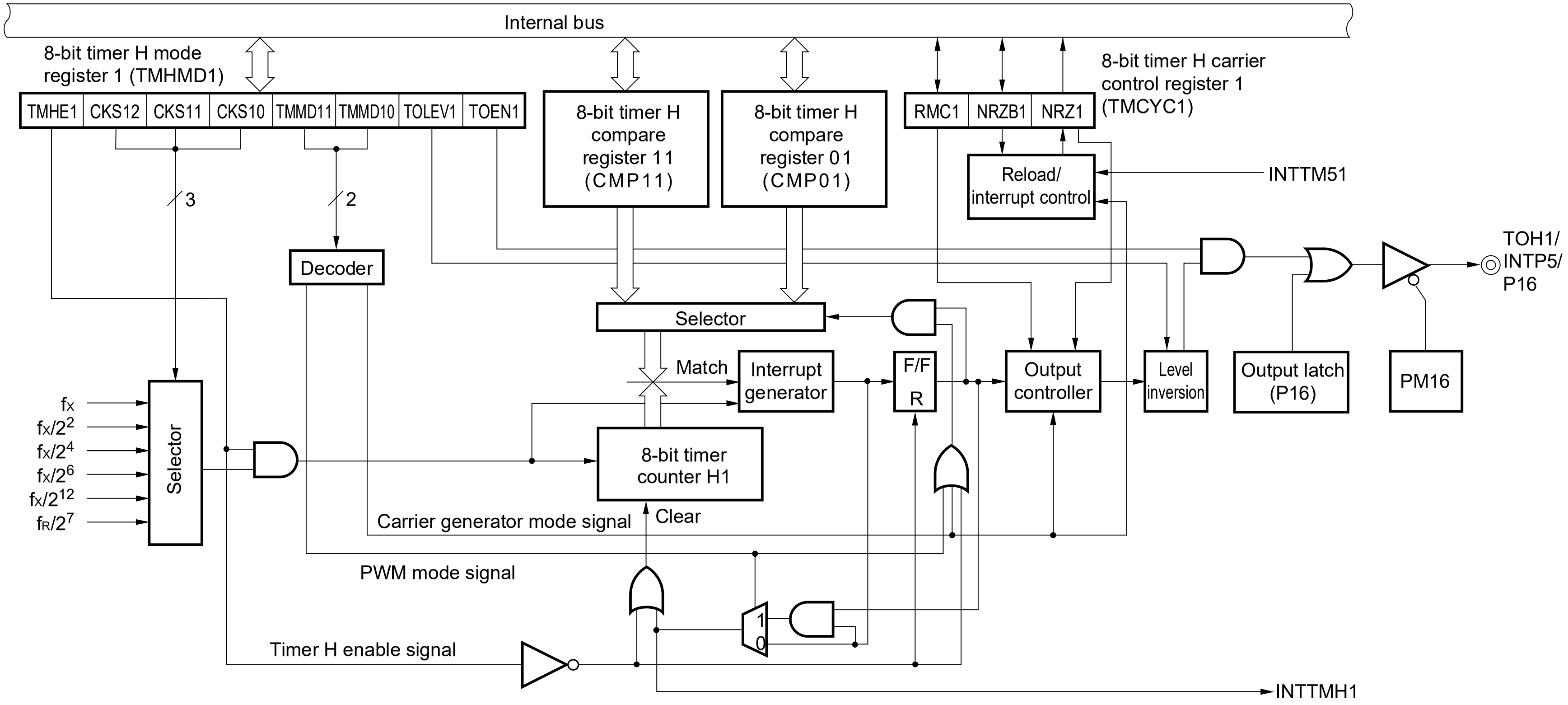


Figure 8-2. Block Diagram of 8-Bit Timer H1



(1) 8-bit timer H compare register 0n (CMP0n)

This register can be read or written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CMP0n to 00H.

Figure 8-3. Format of 8-Bit Timer H Compare Register 0n (CMP0n)

Address: FF18H (CMP00), FF1AH (CMP01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP0n (n = 0, 1)								

Caution CMP0n cannot be rewritten during timer count operation.

(2) 8-bit timer H compare register 1n (CMP1n)

This register can be read or written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CMP1n to 00H.

Figure 8-4. Format of 8-Bit Timer H Compare Register 1n (CMP1n)

Address: FF19H (CMP10), FF1BH (CMP11) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP1n (n = 0, 1)								

CMP1n can be rewritten during timer count operation.

An interrupt request signal (INTTMHn) is generated if the timer count values and CMP1n match after setting CMP1n in carrier generator mode. The timer count value is cleared at the same time. If the CMP1n value is rewritten during timer operation, transferring is performed at the timing at which the count value and CMP1n value match. If the transfer timing and writing from CPU to CMP1n conflict, transfer is not performed.

Caution In the PWM output mode and carrier generator mode, be sure to set CMP1n when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to CMP1n).

Remark n = 0, 1

8.3 Registers Controlling 8-Bit Timers H0 and H1

The following four registers are used to control 8-bit timers H0 and H1.

- 8-bit timer H mode register n (TMHMDn)
- 8-bit timer H carrier control register 1 (TMCYC1)[Note]
- Port mode register 1 (PM1)
- Port register 1 (P1)

Note 8-bit timer H1 only

(1) 8-bit timer H mode register n (TMHMDn)

This register controls the mode of timer H.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TMHMDn to 00H.

Remark n = 0, 1

Figure 8-5. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF69H After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock (f_{CNT}) selection[Note 1]	
0	0	0	f_X	(10 MHz)
0	0	1	$f_X/2$	(5 MHz)
0	1	0	$f_X/2^2$	(2.5 MHz)
0	1	1	$f_X/2^6$	(156.25 kHz)
1	0	0	$f_X/2^{10}$	(9.77 kHz)
1	0	1	TM50 output[Note 2]	
Other than above			Setting prohibited	

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Notes 1. Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Count clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Count clock ≤ 5 MHz

2. Note the following points when selecting the TM50 output as the count clock.

- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.
- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the count clock is the Ring-OSC clock, the operation of 8-bit timer H0 is not guaranteed.

2. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited.

3. In the PWM output mode, be sure to set 8-bit timer H compare register 10 (CMP10) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP10).

Remarks 1. f_X: High-speed system clock oscillation frequency

2. Figures in parentheses apply to operation at f_X = 10 MHz.

3. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

Figure 8-6. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock (f_{CNT}) selection[Note]
0	0	0	f_X (10 MHz)
0	0	1	$f_X/2^2$ (2.5 MHz)
0	1	0	$f_X/2^4$ (625 kHz)
0	1	1	$f_X/2^6$ (156.25 kHz)
1	0	0	$f_X/2^{12}$ (2.44 kHz)
1	0	1	$f_R/2^7$ (1.88 kHz (TYP.))
Other than above			Setting prohibited

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Note Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Count clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Count clock ≤ 5 MHz

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the count clock is the Ring-OSC clock, the operation of 8-bit timer H1 is not guaranteed (except when CKS12, CKS11, CKS10 = 1, 0, 1 ($f_R/2^7$)).

2. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited.

Cautions 3. In the PWM output mode and carrier generator mode, be sure to set 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

4. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

Remarks 1. f_X: High-speed system clock oscillation frequency

2. f_R: Ring-OSC clock oscillation frequency

3. Figures in parentheses apply to operation at f_X = 10 MHz, f_R = 240 kHz (TYP.).

(2) 8-bit timer H carrier control register 1 (TMCYC1)

This register controls the remote control output and carrier pulse output status of 8-bit timer H1.

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

$\overline{\text{RESET}}$ input clears TMCYC1 to 00H.

Figure 8-7. Format of 8-Bit Timer H Carrier Control Register 1 (TMCYC1)

Address: FF6DH After reset: 00H R/W[Note]

	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	Remote control output
0	0	Low-level output
0	1	High-level output
1	0	Low-level output
1	1	Carrier pulse output

NRZ1	Carrier pulse output status flag
0	Carrier output disabled status (low-level status)
1	Carrier output enabled status (RMC1 = 1: Carrier pulse output, RMC1 = 0: High-level status)

Note Bit 0 is read-only.

(3) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P15/TOH0 and P16/TOH1/INTP5 pins for timer output, clear PM15 and PM16 and the output latches of P15 and P16 to 0.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM1 to FFH.

Figure 8-8. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 8-Bit Timers H0 and H1

8.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter Hn and compare register 0n (CMP0n) match, an interrupt request signal (INTTMHn) is generated and 8-bit timer counter Hn is cleared to 00H.

Compare register 1n (CMP1n) is not used in interval timer mode. Since a match of 8-bit timer counter Hn and the CMP1n register is not detected even if the CMP1n register is set, timer output is not affected.

By setting bit 0 (TOENn) of timer H mode register n (TMHMDn) to 1, a square wave of any frequency (duty = 50%) is output from TOHn.

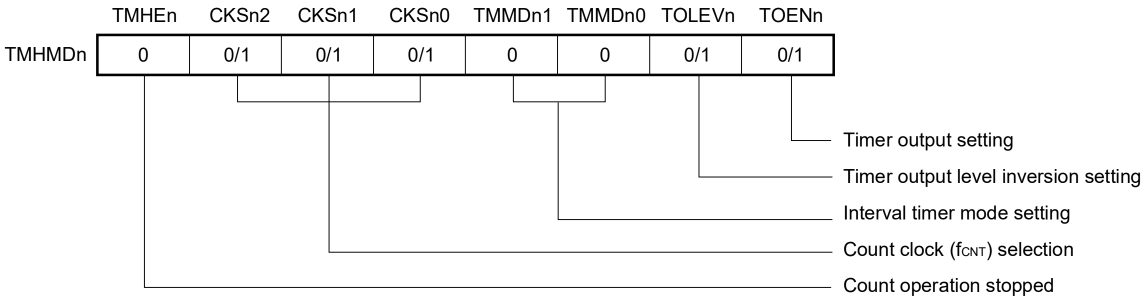
(1) Usage

Generates the INTTMHn signal repeatedly at the same interval.

<1> Set each register.

Figure 8-9. Register Setting During Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register n (TMHMDn)



	TMHEn	CKSn2	CKSn1	CKSn0	TMMDn1	TMMDn0	TOLEVn	TOENn
TMHMDn	0	0/1	0/1	0/1	0	0	0/1	0/1

Timer output setting
Timer output level inversion setting
Interval timer mode setting
Count clock (f_{CNT}) selection
Count operation stopped

(ii) CMP0n register setting

- Compare value (N)

<2> Count operation starts when TMHEn = 1.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the INTTMHn signal is generated and 8-bit timer counter Hn is cleared to 00H.

Interval time = (N + 1)/f_{CNT}

<4> Subsequently, the INTTMHn signal is generated at the same interval. To stop the count operation, clear TMHEn to 0.

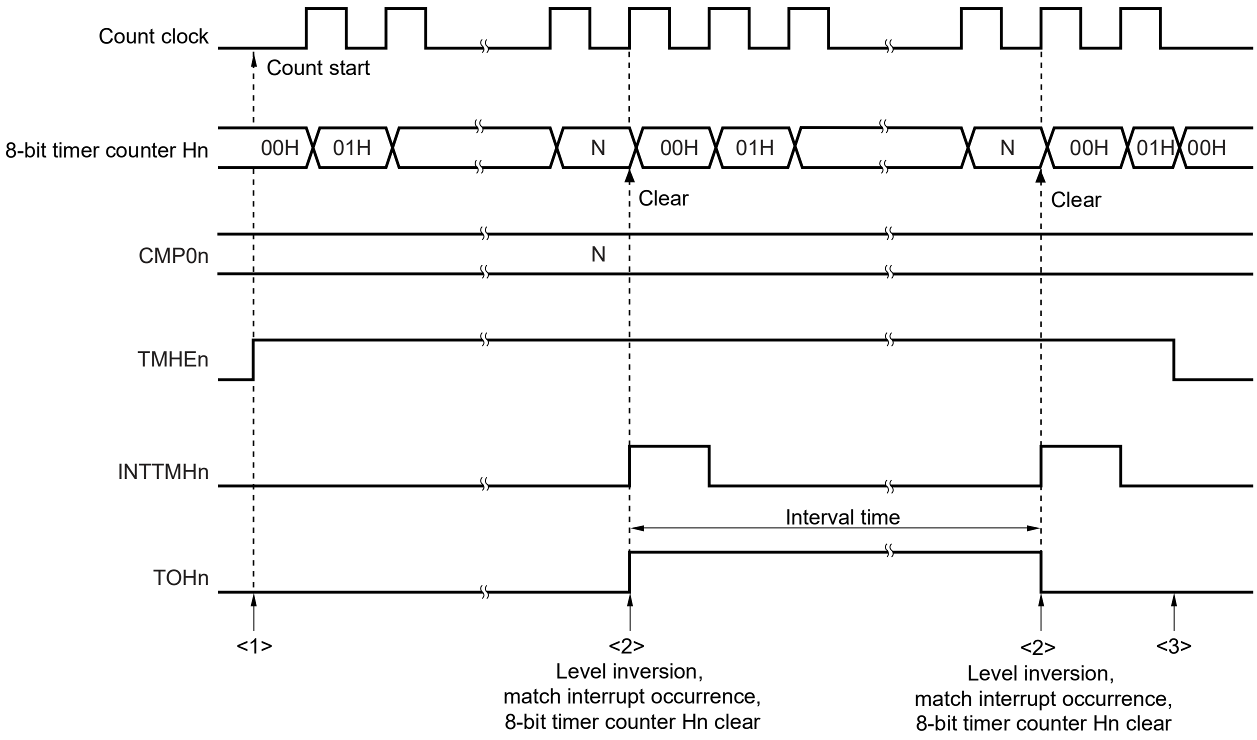
Remark n = 0, 1

(2) Timing chart

The timing of the interval timer/square-wave output operation is shown below.

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

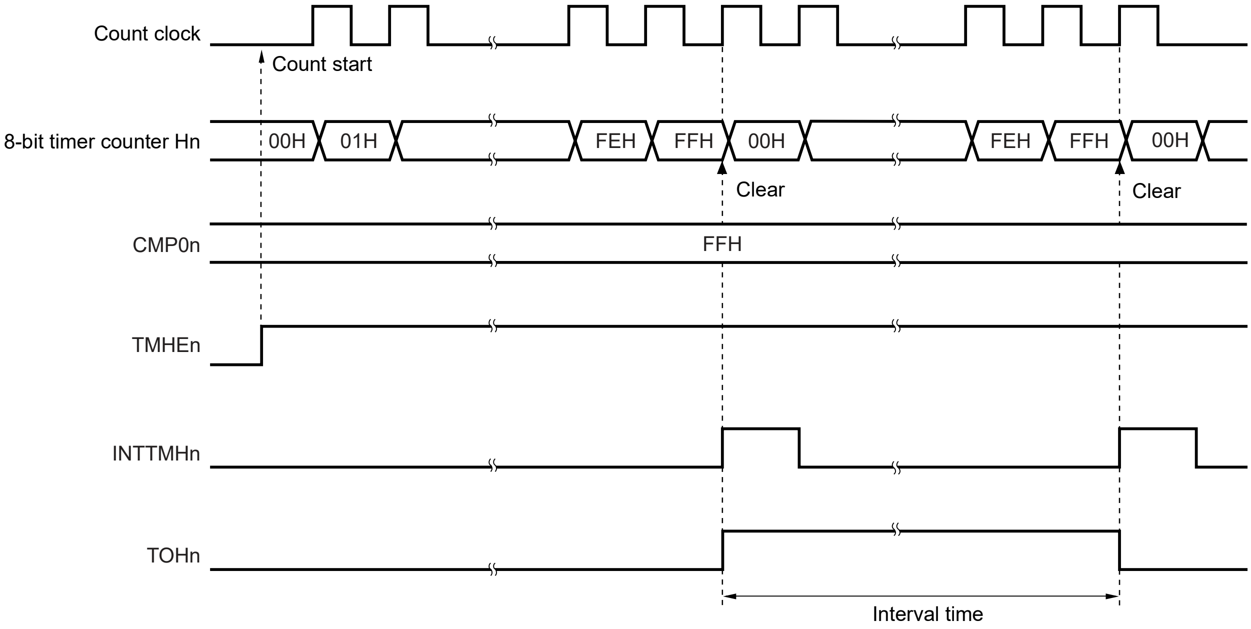
<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output level is inverted, and the INTTMHn signal is output.

<3> The INTTMHn signal and TOHn output become inactive by clearing the TMHEn bit to 0 during timer Hn operation. If these are inactive from the first, the level is retained.

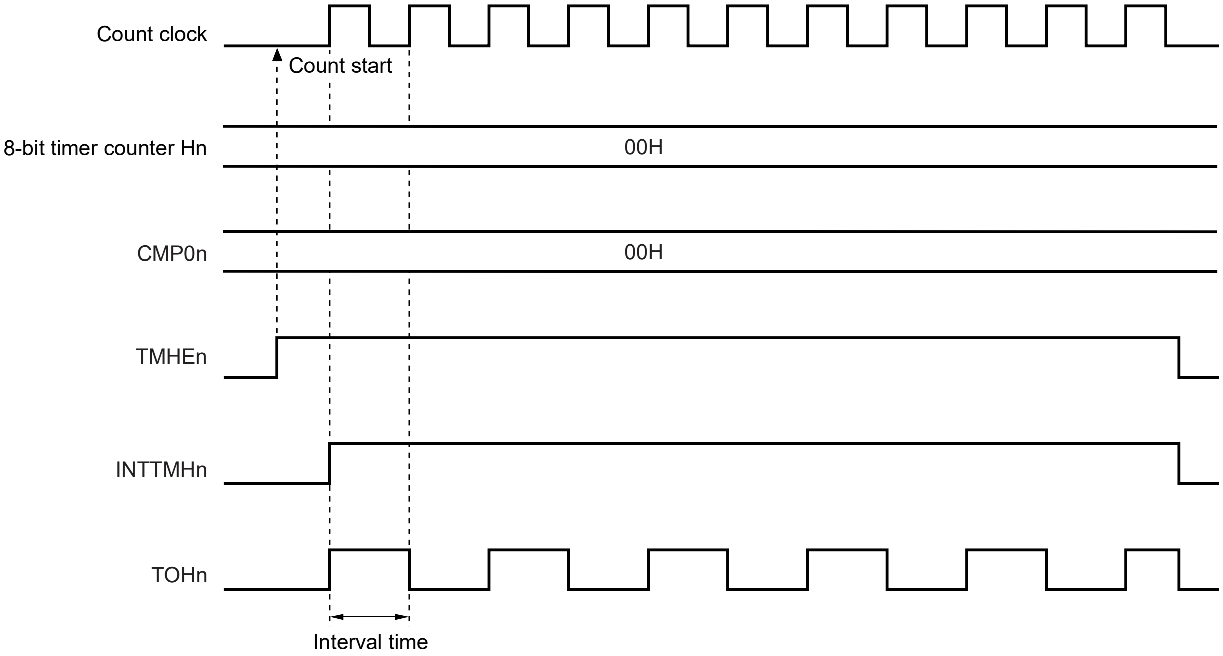
Remark n = 0, 1
N = 01H to FEH

Figure 8-10. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP0n = FFH



(c) Operation when CMP0n = 00H



Remark n = 0, 1

8.4.2 Operation as PWM output mode

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

8-bit timer compare register 0n (CMP0n) controls the cycle of timer output (TOHn). Rewriting the CMP0n register during timer operation is prohibited.

8-bit timer compare register 1n (CMP1n) controls the duty of timer output (TOHn). Rewriting the CMP1n register during timer operation is possible.

The operation in PWM output mode is as follows.

TOHn output becomes active and 8-bit timer counter Hn is cleared to 0 when 8-bit timer counter Hn and the CMP0n register match after the timer count is started. TOHn output becomes inactive when 8-bit timer counter Hn and the CMP1n register match.

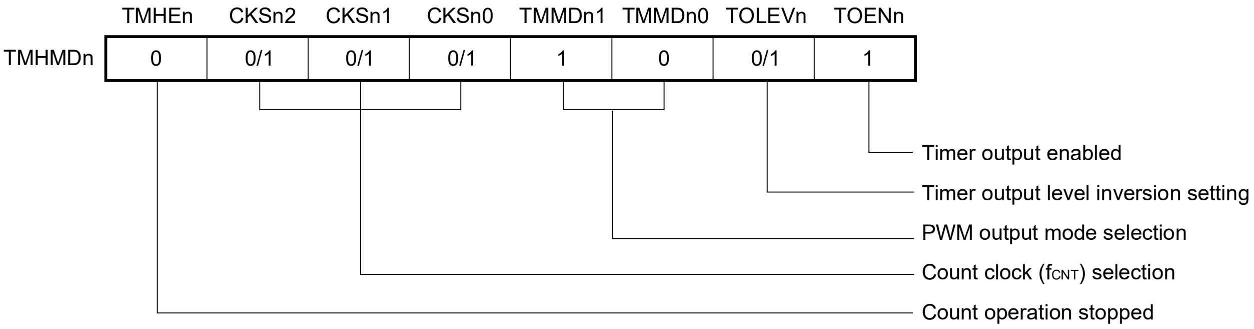
(1) Usage

In PWM output mode, a pulse for which an arbitrary duty and arbitrary cycle can be set is output.

<1> Set each register.

Figure 8-11. Register Setting in PWM Output Mode

(i) Setting timer H mode register n (TMHMDn)



	TMHEn	CKSn2	CKSn1	CKSn0	TMMDn1	TMMDn0	TOLEVn	TOENn
TMHMDn	0	0/1	0/1	0/1	1	0	0/1	1

Timer output enabled
Timer output level inversion setting
PWM output mode selection
Count clock (f_{CNT}) selection
Count operation stopped

(ii) Setting CMP0n register

- Compare value (N): Cycle setting

(iii) Setting CMP1n register

- Compare value (M): Duty setting

Remarks 1. n = 0, 1

2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> The count operation starts when TMHEn = 1.

<3> The CMP0n register is the compare register that is to be compared first after counter operation is enabled. When the values of 8-bit timer counter Hn and the CMP0n register match, 8-bit timer counter Hn is cleared, an interrupt request signal (INTTMHn) is generated, and TOHn output becomes active. At the same time, the compare register to be compared with 8-bit timer counter Hn is changed from the CMP0n register to the CMP1n register.

<4> When 8-bit timer counter Hn and the CMP1n register match, TOHn output becomes inactive and the compare register to be compared with 8-bit timer counter Hn is changed from the CMP1n register to the CMP0n register. At this time, 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHEn = 0.

If the setting value of the CMP0n register is N, the setting value of the CMP1n register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

PWM pulse output cycle = (N + 1)/f_{CNT}
Duty = Active width : Total width of PWM = (M + 1) : (N + 1)

Cautions 1. In PWM output mode, three operation clocks (signal selected using the CKSn2 to CKSn0 bits of the TMHMDn register) are required to transfer the CMP1n register value after rewriting the register.

2. Be sure to set the CMP1n register when starting the timer count operation (TMHEn = 1) after the timer count operation was stopped (TMHEn = 0) (be sure to set again even if setting the same value to the CMP1n register).

Remark n = 0, 1

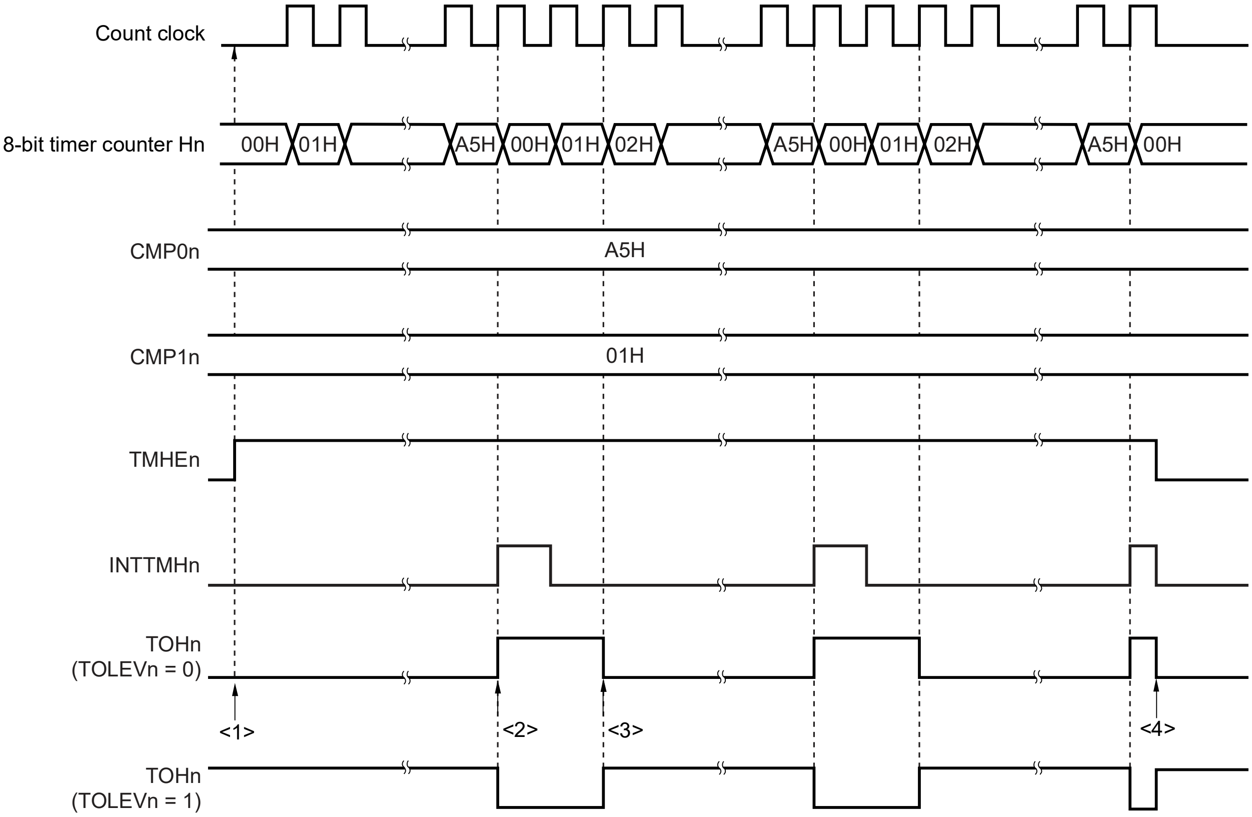
(2) Timing chart

The operation timing in PWM output mode is shown below.

Caution Make sure that the CMP1n register setting value (M) and CMP0n register setting value (N) are within the following range.
00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

Figure 8-12. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, TOHn output remains inactive (when TOLEVn = 0).

<2> When the values of 8-bit timer counter Hn and the CMP0n register match, the TOHn output level is inverted, the value of 8-bit timer counter Hn is cleared, and the INTTMHn signal is output.

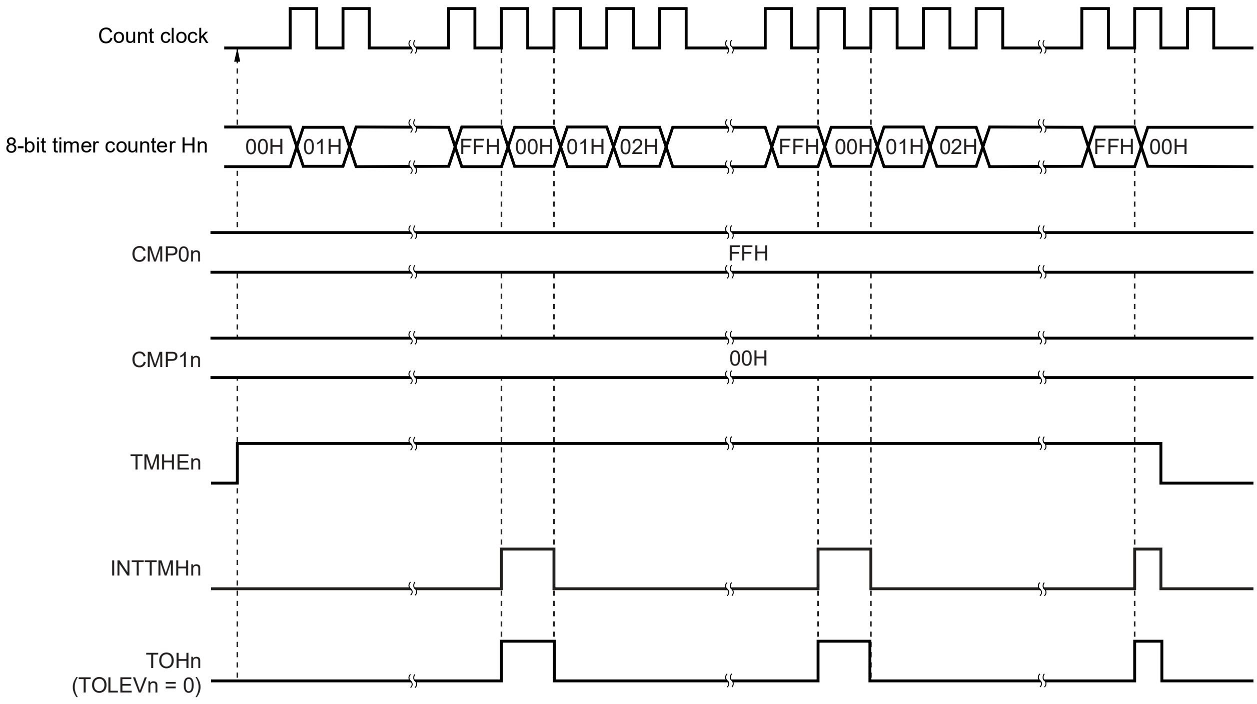
<3> When the values of 8-bit timer counter Hn and the CMP1n register match, the level of the TOHn output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMHn signal is not output.

<4> Clearing the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

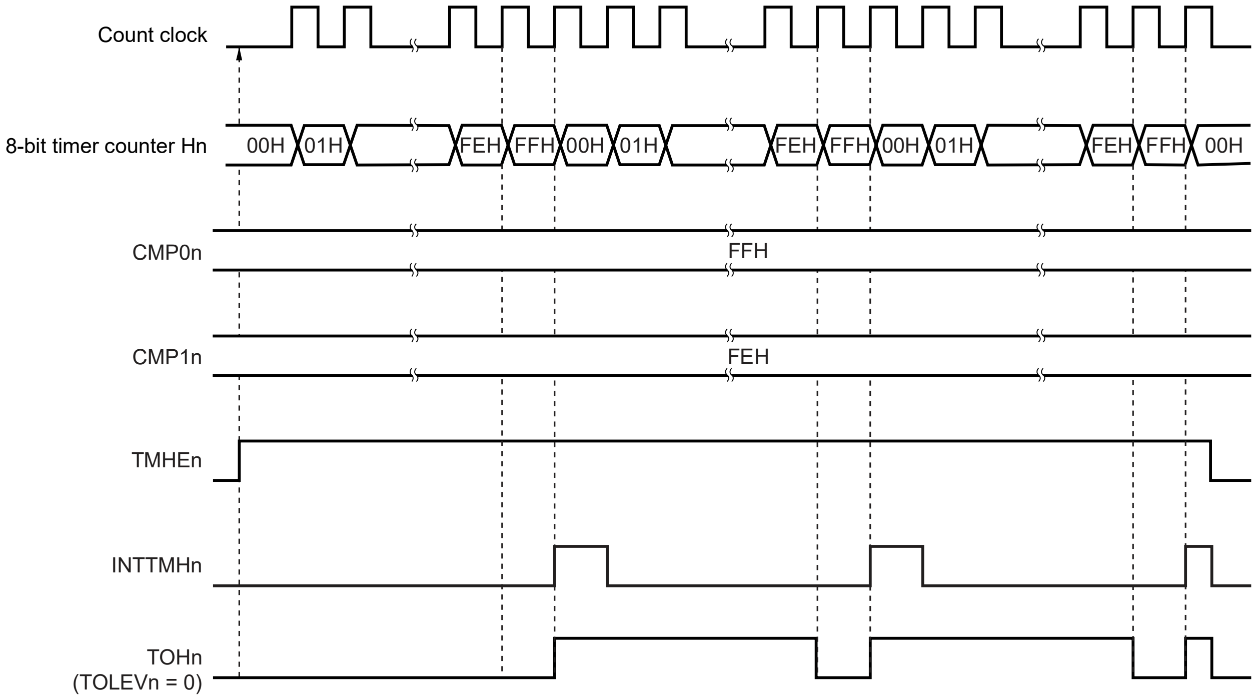
Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP0n = FFH, CMP1n = 00H



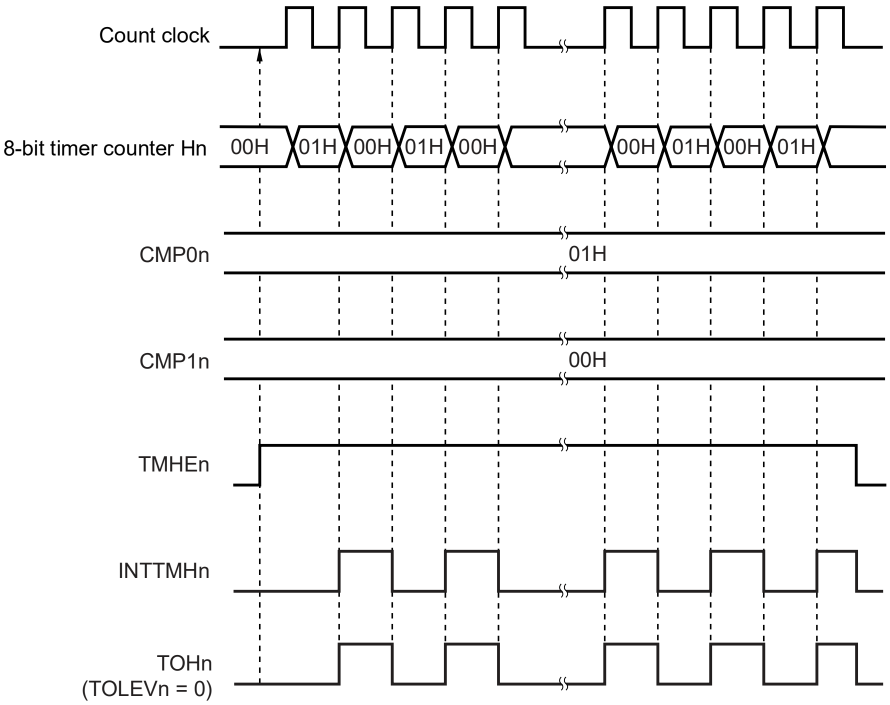
(c) Operation when CMP0n = FFH, CMP1n = FEH



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (3/4)

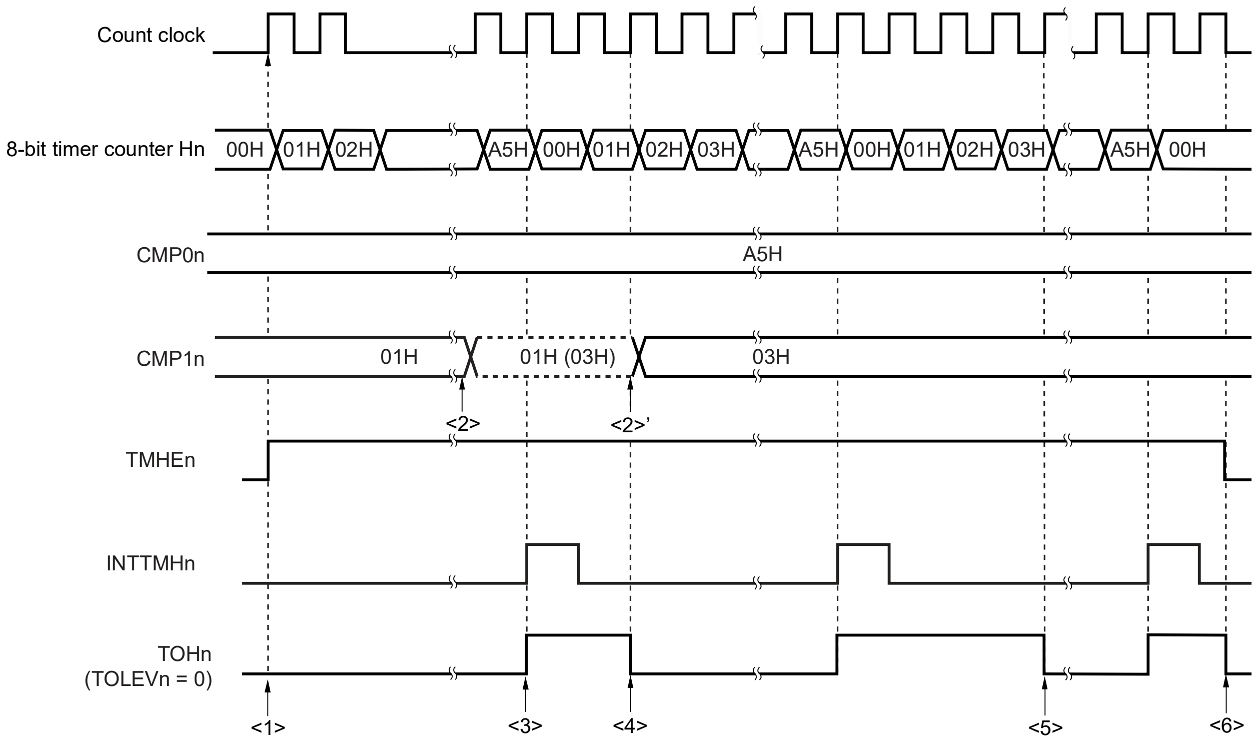
(d) Operation when CMP0n = 01H, CMP1n = 00H



Remark n = 0, 1

Figure 8-12. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP1n (CMP1n = 01H → 03H, CMP0n = A5H)



<1> The count operation is enabled by setting the TMHEn bit to 1. Start 8-bit timer counter Hn by masking one count clock to count up. At this time, the TOHn output remains inactive (when TOLEVn = 0).

<2> The CMP1n register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of 8-bit timer counter Hn and the CMP0n register match, the value of 8-bit timer counter Hn is cleared, the TOHn output becomes active, and the INTTMHn signal is output.

<4> If the CMP1n register value is changed, the value is latched and not transferred to the register. When the values of 8-bit timer counter Hn and the CMP1n register before the change match, the value is transferred to the CMP1n register and the CMP1n register value is changed (<2>').

However, three count clocks or more are required from when the CMP1n register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of 8-bit timer counter Hn and the CMP1n register after the change match, the TOHn output becomes inactive. 8-bit timer counter Hn is not cleared and the INTTMHn signal is not generated.

<6> Clearing the TMHEn bit to 0 during timer Hn operation makes the INTTMHn signal and TOHn output inactive.

Remark n = 0, 1

8.4.3 Carrier generator mode operation (8-bit timer H1 only)

The carrier clock generated by 8-bit timer H1 is output in the cycle set by 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

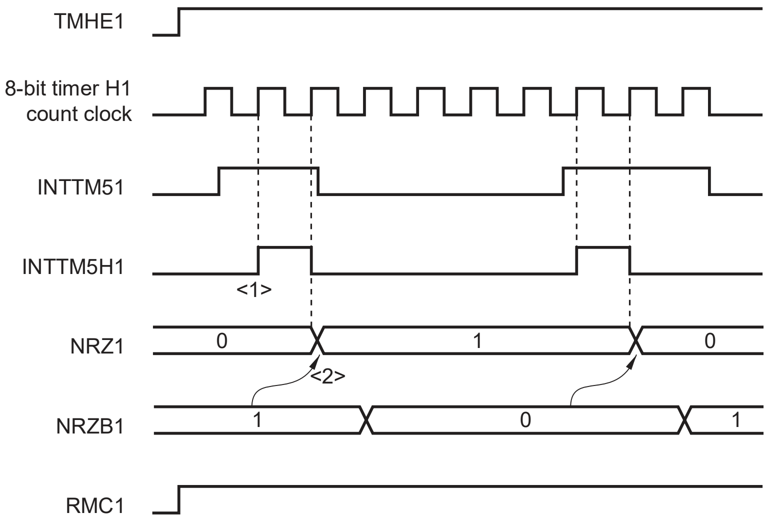
(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output
1	0	Low-level output
1	1	Carrier pulse output

To control the carrier pulse output during a count operation, the NRZ1 and NRZB1 bits of the TMCYC1 register have a master and slave bit configuration. The NRZ1 bit is read-only but the NRZB1 bit can be read and written. The INTTM51 signal is synchronized with the 8-bit timer H1 count clock and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal of the NRZ1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit. The timing for transfer from the NRZB1 bit to the NRZ1 bit is as shown below.

Figure 8-13. Transfer Timing



<1> The INTTM51 signal is synchronized with the count clock of 8-bit timer H1 and is output as the INTTM5H1 signal.

<2> The value of the NRZB1 bit is transferred to the NRZ1 bit at the second clock from the rising edge of the INTTM5H1 signal.

Cautions 1. Do not rewrite the NRZB1 bit again until at least the second clock after it has been rewritten, or else the transfer from the NRZB1 bit to the NRZ1 bit is not guaranteed.

2. When 8-bit timer/event counter 51 is used in the carrier generator mode, an interrupt is generated at the timing of <1>. When 8-bit timer/event counter 51 is used in a mode other than the carrier generator mode, the timing of the interrupt generation differs.

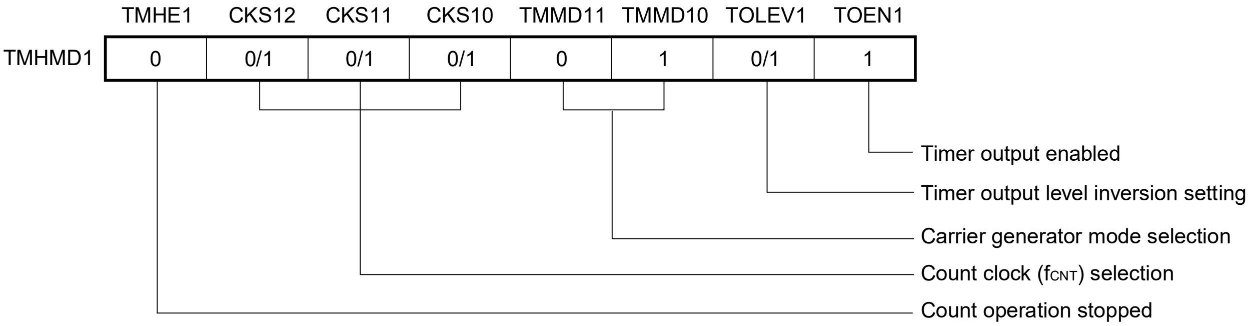
(3) Usage

Outputs an arbitrary carrier clock from the TOH1 pin.

<1> Set each register.

Figure 8-14. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)



	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1
TMHMD1	0	0/1	0/1	0/1	0	1	0/1	1

Timer output enabled
Timer output level inversion setting
Carrier generator mode selection
Count clock (f_{CNT}) selection
Count operation stopped

(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... Carrier output enable bit

(v) TCL51 and TMC51 register setting

- Refer to **7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

<2> When TMHE1 = 1, 8-bit timer H1 starts counting.

<3> When TCE51 of 8-bit timer mode control register 51 (TMC51) is set to 1, 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared, and at the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with the count clock of 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> When the NRZ1 bit is high level, a carrier clock is output from the TOH1 pin.

<9> By performing the procedures above, an arbitrary carrier clock is obtained. To stop the count operation, clear TMHE1 to 0.

If the setting value of the CMP01 register is N, the setting value of the CMP11 register is M, and the count clock frequency is f_{CNT}, the carrier clock output cycle and duty are as follows.

Carrier clock output cycle = (N + M + 2)/f_{CNT} Duty = High-level width : Carrier clock output width = (M + 1) : (N + M + 2)

Cautions 1. Be sure to set the CMP11 register when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to the CMP11 register).

2. Set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

(4) Timing chart

The carrier output control timing is shown below.

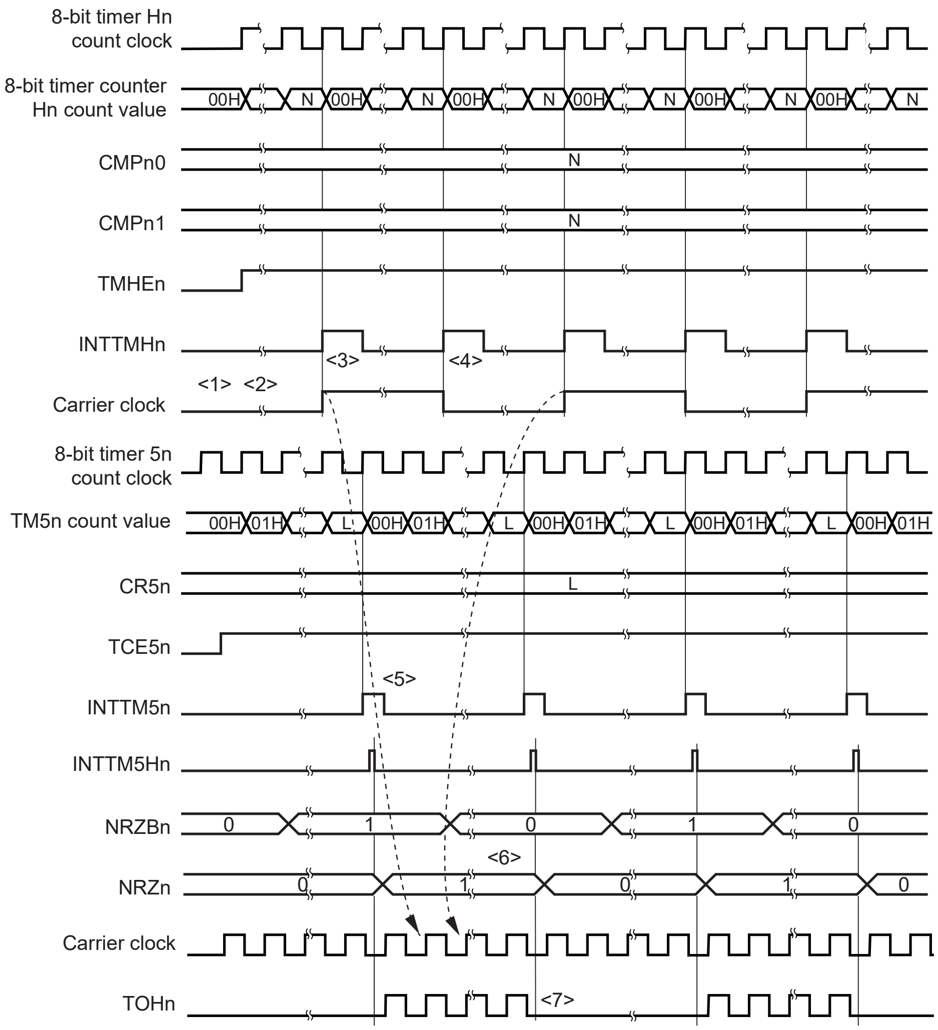
Cautions 1. Set the values of the CMP01 and CMP11 registers in a range of 01H to FFH.

2. In the carrier generator mode, three operating clocks (signal selected by CKS12 to CKS10 bits of TMHMD1 register) or more are required from when the CMP11 register value is changed to when the value is transferred to the register.

3. Be sure to set the RMC1 bit before the count operation is started.

Figure 8-15. Carrier Generator Mode Operation Timing (1/3)

(a) Operation when CMP01 = N, CMP11 = N



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer counter H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<3> When the count value of 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to 50% is generated.

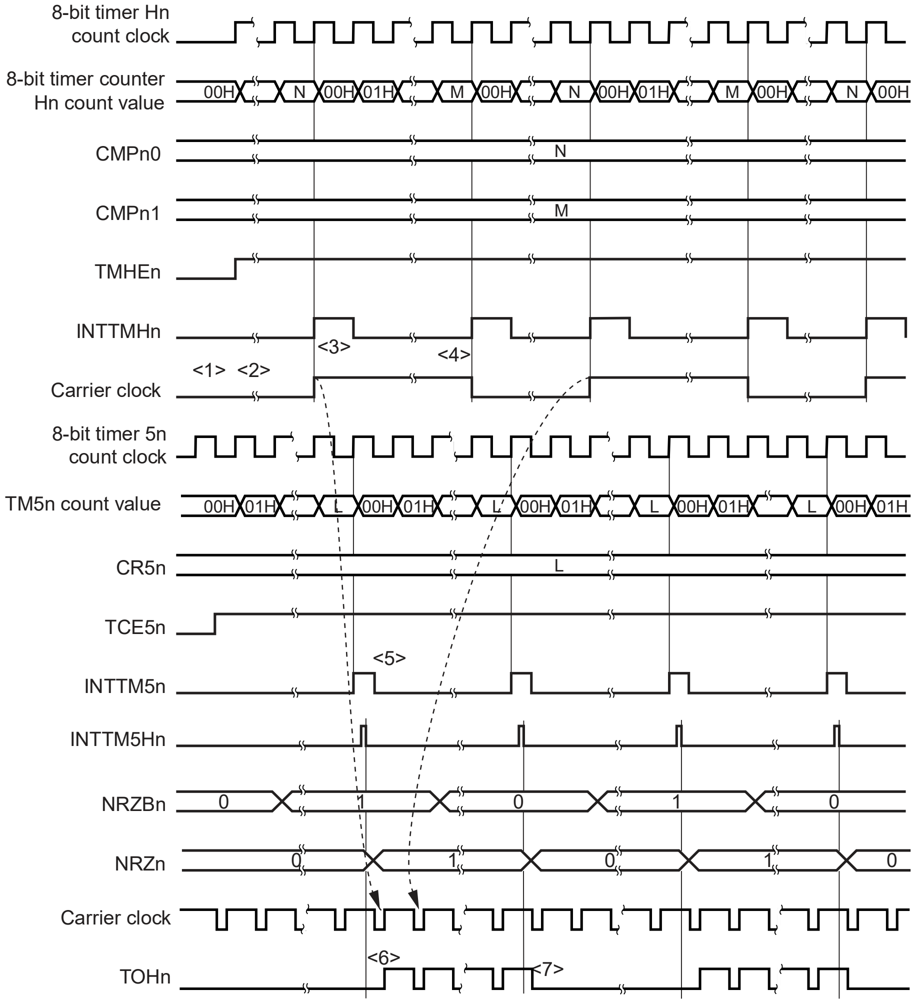
<5> When the INTTM51 signal is generated, it is synchronized with the count clock of 8-bit timer H1 and output as the INTTM5H1 signal.

<6> The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<7> When NRZ1 = 0 is set, the TOH1 output becomes low level.

Figure 8-15. Carrier Generator Mode Operation Timing (2/3)

(b) Operation when CMP01 = N, CMP11 = M



<1> When TMHE1 = 0 and TCE51 = 0, 8-bit timer counter H1 operation is stopped.

<2> When TMHE1 = 1 is set, 8-bit timer counter H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<3> When the count value of 8-bit timer counter H1 matches the CMP01 register value, the first INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register. 8-bit timer counter H1 is cleared to 00H.

<4> When the count value of 8-bit timer counter H1 matches the CMP11 register value, the INTTMH1 signal is generated, the carrier clock signal is inverted, and the compare register to be compared with 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register. 8-bit timer counter H1 is cleared to 00H. By performing procedures <3> and <4> repeatedly, a carrier clock with duty fixed to other than 50% is generated.

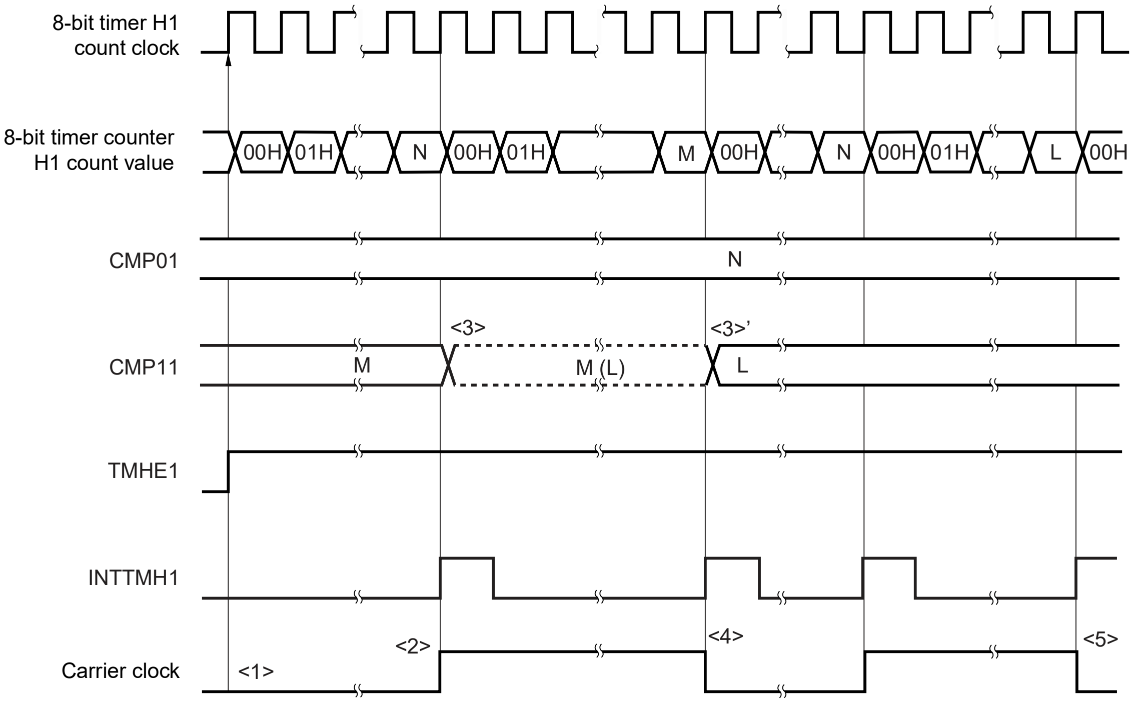
<5> When the INTTM51 signal is generated, it is synchronized with the count clock of 8-bit timer H1 and output as the INTTM5H1 signal.

<6> A carrier signal is output at the first rising edge of the carrier clock if NRZ1 is set to 1.

<7> When NRZ1 = 0, the TOH1 output is held at the high level and is not changed to low level while the carrier clock is high level (from <6> and <7>, the high-level width of the carrier clock waveform is guaranteed).

Figure 8-15. Carrier Generator Mode Operation Timing (3/3)

(c) Operation when CMP11 is changed



<1> When TMHE1 = 1 is set, 8-bit timer counter H1 starts a count operation. At that time, the carrier clock is held at the inactive level.

<2> When the count value of 8-bit timer counter H1 matches the CMP01 register value, 8-bit timer counter H1 is cleared and the INTTMH1 signal is output.

<3> The CMP11 register can be rewritten during 8-bit timer H1 operation, however, the changed value (L) is latched. The CMP11 register is changed when the count value of 8-bit timer counter H1 and the CMP11 register value before the change (M) match (<3>').

<4> When the count value of 8-bit timer counter H1 and the CMP11 register value before the change (M) match, the INTTMH1 signal is output, the carrier signal is inverted, and 8-bit timer counter H1 is cleared to 00H.

<5> The timing at which the count value of 8-bit timer counter H1 and the CMP11 register value match again is indicated by the value after the change (L).

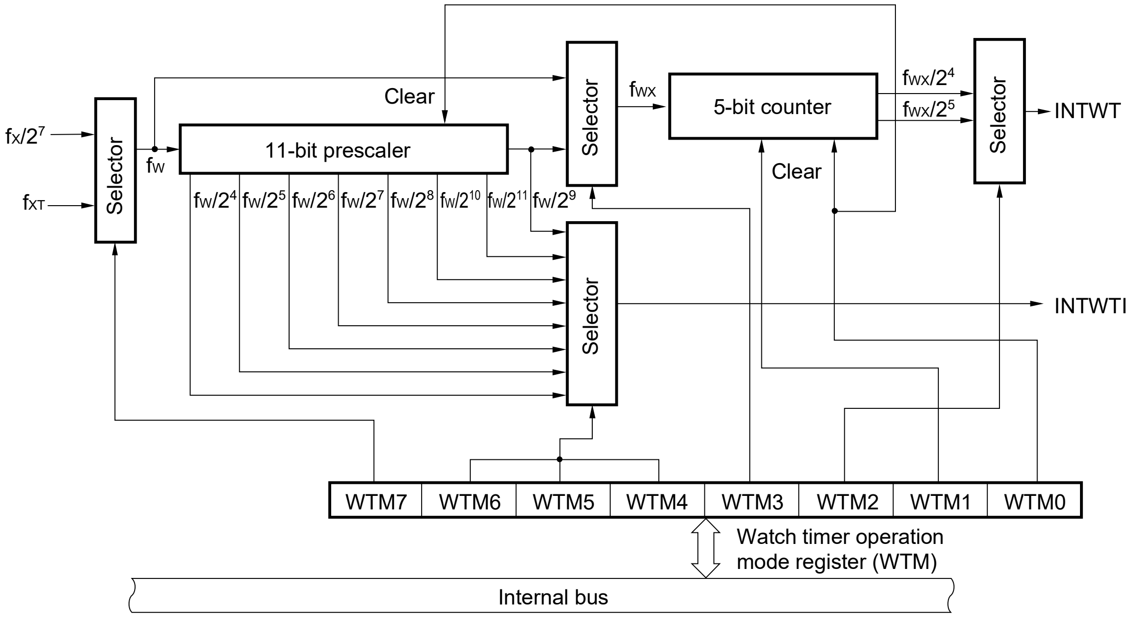
9.1 Functions of Watch Timer

The watch timer has the following functions.

- Watch timer
- Interval timer

The watch timer and the interval timer can be used simultaneously.
Figure 9-1 shows the watch timer block diagram.

Figure 9-1. Watch Timer Block Diagram


fx/2^7
fXT
Selector
fW
11-bit prescaler
Clear
Selector
fWX
5-bit counter
Clear
fWX/2^4
fWX/2^5
Selector
INTWT
fW/2^4 fW/2^5 fW/2^6 fW/2^7 fW/2^8 fW/2^10 fW/2^11 fW/2^9
Selector
INTWTI
WTM7 WTM6 WTM5 WTM4 WTM3 WTM2 WTM1 WTM0
Watch timer operation mode register (WTM)
Internal bus

Remark fX: High-speed system clock oscillation frequency
fXT: Subsystem clock oscillation frequency
fW: Watch timer clock frequency
fWX: fW or fW/2^9

(1) Watch timer

When the high-speed system clock or subsystem clock is used, interrupt requests (INTWT) are generated at preset intervals.

Table 9-1. Watch Timer Interrupt Time

Interrupt Time	When Operated at f_{XT} = 32.768 kHz	When Operated at f_X = 10 MHz
$2^4/f_W$	488 μs	205 μs
$2^5/f_W$	977 μs	410 μs
$2^{13}/f_W$	0.25 s	0.105 s
$2^{14}/f_W$	0.5 s	0.210 s

Remark f_X: High-speed system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

(2) Interval timer

Interrupt requests (INTWTI) are generated at preset time intervals.

Table 9-2. Interval Timer Interval Time

Interval Time	When Operated at f_{XT} = 32.768 kHz	When Operated at f_X = 10 MHz
$2^4/f_W$	488 μs	205 μs
$2^5/f_W$	977 μs	410 μs
$2^6/f_W$	1.95 ms	820 μs
$2^7/f_W$	3.91 ms	1.64 ms
$2^8/f_W$	7.81 ms	3.28 ms
$2^9/f_W$	15.6 ms	6.55 ms
$2^{10}/f_W$	31.3 ms	13.1 ms
$2^{11}/f_W$	62.5 ms	26.2 ms

Remark f_X: High-speed system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

9.2 Configuration of Watch Timer

The watch timer includes the following hardware.

Table 9-3. Watch Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	11 bits × 1
Control register	Watch timer operation mode register (WTM)

9.3 Register Controlling Watch Timer

The watch timer is controlled by the watch timer operation mode register (WTM).

- **Watch timer operation mode register (WTM)**
  This register sets the watch timer count clock, enables/disables operation, prescaler interval time, and 5-bit counter operation control.
  WTM can be set by a 1-bit or 8-bit memory manipulation instruction.
  $\overline{\text{RESET}}$ input clears WTM to 00H.

Figure 9-2. Format of Watch Timer Operation Mode Register (WTM)

Address: FF6FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	Watch timer count clock selection
0	$f_X/2^7$ (78.125 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval time selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
1	1	0	$2^{10}/f_W$
1	1	1	$2^{11}/f_W$

WTM3	WTM2	Interrupt time selection
0	0	$2^{14}/f_W$
0	1	$2^{13}/f_W$
1	0	$2^5/f_W$
1	1	$2^4/f_W$

WTM1	5-bit counter operation control
0	Clear after operation stop
1	Start

WTM0	Watch timer operation enable
0	Operation stop (clear both prescaler and timer)
1	Operation enable

Caution Do not change the count clock and interval time (by setting bits 4 to 7 (WTM4 to WTM7) of WTM) during watch timer operation.

Remarks 1. f_W: Watch timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X: High-speed system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. Figures in parentheses apply to operation with f_X = 10 MHz, f_{XT} = 32.768 kHz.

9.4 Watch Timer Operations

9.4.1 Watch timer operation

The watch timer generates an interrupt request (INTWT) at a specific time interval by using the high-speed system clock or subsystem clock.

When bit 0 (WTM0) and bit 1 (WTM1) of the watch timer operation mode register (WTM) are set to 1, the count operation starts. When these bits are cleared to 0, the 5-bit counter is cleared and the count operation stops.

When the interval timer is simultaneously operated, zero-second start can be achieved only for the watch timer by clearing WTM1 to 0. In this case, however, the 11-bit prescaler is not cleared. Therefore, an error up to $2^{11} \times 1/f_W$ seconds occurs in the first overflow (INTWT) after zero-second start.

The interrupt request is generated at the following time intervals.

Table 9-4. Watch Timer Interrupt Time

WTM3	WTM2	Interrupt Time Selection	When Operated at f_{XT} = 32.768 kHz (WTM7 = 1)	When Operated at f_X = 10 MHz (WTM7 = 0)
0	0	$2^{14}/f_W$	0.5 s	0.210 s
0	1	$2^{13}/f_W$	0.25 s	0.105 s
1	0	$2^{5}/f_W$	977 μs	410 μs
1	1	$2^{4}/f_W$	488 μs	205 μs

Remark f_X: High-speed system clock oscillation frequency
f_{XT}: Subsystem clock oscillation frequency
f_W: Watch timer clock frequency

9.4.2 Interval timer operation

The watch timer operates as interval timer which generates interrupt requests (INTWTI) repeatedly at an interval of the preset count value.

The interval time can be selected with bits 4 to 6 (WTM4 to WTM6) of the watch timer operation mode register (WTM). When bit 0 (WTM0) of the WTM is set to 1, the count operation starts. When this bit is cleared to 0, the count operation stops.

Table 9-5. Interval Timer Interval Time

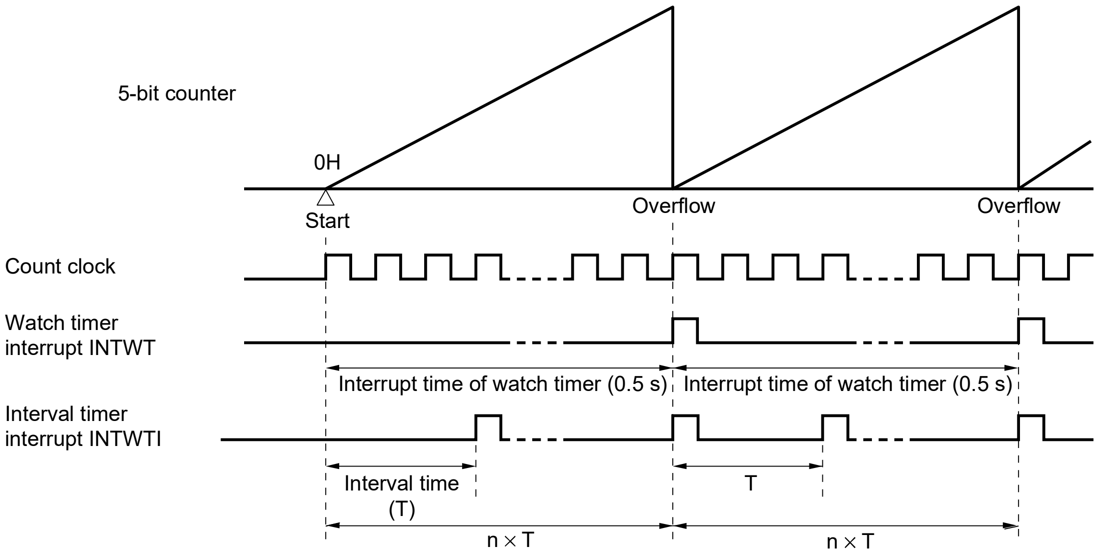
WTM6	WTM5	WTM4	Interval Time	When Operated at f_{XT} = 32.768 kHz (WTM7 = 1)	When Operated at f_X = 10 MHz (WTM7 = 0)
0	0	0	$2^4/f_W$	488 μs	205 μs
0	0	1	$2^5/f_W$	977 μs	410 μs
0	1	0	$2^6/f_W$	1.95 ms	820 μs
0	1	1	$2^7/f_W$	3.91 ms	1.64 ms
1	0	0	$2^8/f_W$	7.81 ms	3.28 ms
1	0	1	$2^9/f_W$	15.6 ms	6.55 ms
1	1	0	$2^{10}/f_W$	31.3 ms	13.1 ms
1	1	1	$2^{11}/f_W$	62.5 ms	26.2 ms

Remark f_X: High-speed system clock oscillation frequency

f_{XT}: Subsystem clock oscillation frequency

f_W: Watch timer clock frequency

Figure 9-3. Operation Timing of Watch Timer/Interval Timer


5-bit counter
0H
Start
Overflow
Overflow
Count clock
Watch timer interrupt INTWT
Interrupt time of watch timer (0.5 s)
Interrupt time of watch timer (0.5 s)
Interval timer interrupt INTWTI
Interval time (T)
T
n × T
n × T

Remark f_W: Watch timer clock frequency

n: The number of times of interval timer operations

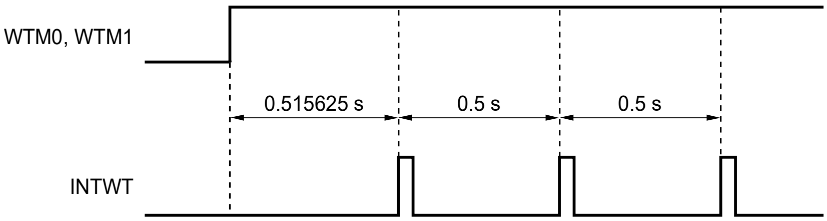
Figures in parentheses apply to operation with f_W = 32.768 kHz (WTM7 = 1, WTM3, WTM2 = 0, 0).

9.5 Cautions for Watch Timer

When operation of the watch timer and 5-bit counter is enabled by the watch timer mode control register (WTM) (by setting bits 0 (WTM0) and 1 (WTM1) of WTM to 1), the interval until the first interrupt request (INTWT) is generated after the register is set does not exactly match the specification made with bit 3 (WTM3) of WTM. This is because there is a delay of one 11-bit (max.) prescaler output cycle until the 5-bit counter starts counting. Subsequently, however, the INTWT signal is generated at the specified intervals.

Figure 9-4. Example of Generation of Watch Timer Interrupt Request (INTWT) (When Interrupt Period = 0.5 s)

It takes 0.515625 seconds (max.) for the first INTWT to be generated (2^9 × 1/32768 = 0.015625 s longer). INTWT is then generated every 0.5 seconds.



CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

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 RESF, refer to **CHAPTER 19 RESET FUNCTION**.

Table 10-1. Loop Detection Time of Watchdog Timer

Loop Detection Time	
During Ring-OSC Clock Operation	During High-Speed System Clock Operation
$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: High-speed system clock oscillation frequency

3. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 10 MHz.

The operation mode of the watchdog timer (WDT) is switched according to the option byte setting of the on-chip Ring-OSC as shown in Table 10-2.

Table 10-2. Option Byte Setting and Watchdog Timer Operation Mode

	Option Byte	
	Ring-OSC Cannot Be Stopped	Ring-OSC Can Be Stopped by Software
Watchdog timer clock source	Fixed to f_R[Note 1].	• Selectable by software (f_{XP}, f_R or stopped) • When reset is released: f_R
Operation after reset	Operation starts with the maximum interval ($f_R/2^{18}$).	Operation starts with the maximum interval ($f_R/2^{18}$).
Operation mode selection	The interval can be changed only once.	The clock selection/interval can be changed only once.
Features	The watchdog timer cannot be stopped.	The watchdog timer can be stopped in standby mode[Note 2].

Notes 1. As long as power is being supplied, Ring-OSC oscillation cannot be stopped (except in the reset period).

2. The conditions under which clock supply to the watchdog timer is stopped differ depending on the clock source of the watchdog timer.

<1> If the clock source is f_{XP}, clock supply to the watchdog timer is stopped under the following conditions.

- When f_{XP} is stopped
- In HALT/STOP mode
- During oscillation stabilization time

<2> If the clock source is f_R, clock supply to the watchdog timer is stopped under the following conditions.

- If the CPU clock is f_{XP} and if f_R is stopped by software before execution of the STOP instruction
- In HALT/STOP mode

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: High-speed system clock oscillation frequency

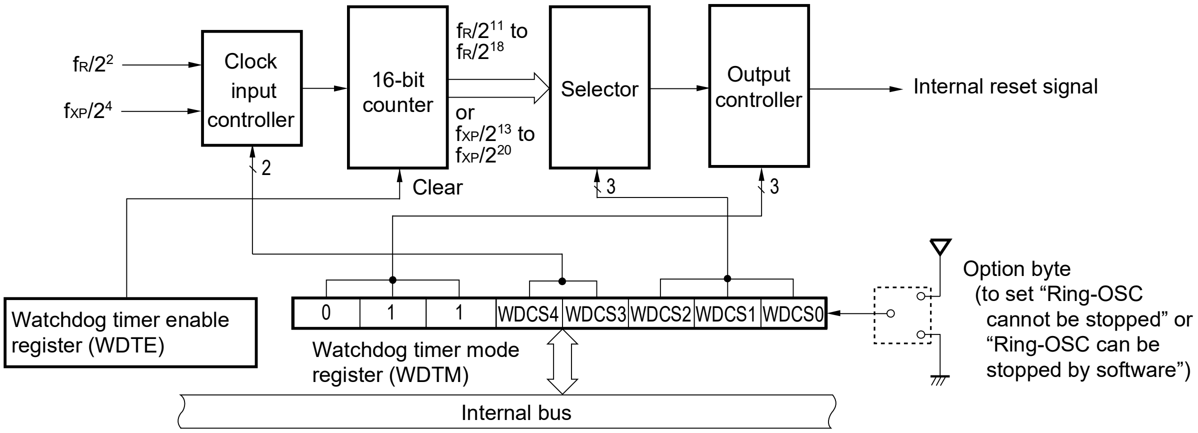
10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

Figure 10-1. Block Diagram of Watchdog Timer



10.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

- Watchdog timer mode register (WDTM)
- Watchdog timer enable register (WDTE)

(1) Watchdog timer mode register (WDTM)

This register sets the overflow time and operation clock of the watchdog timer.

This register can be set by an 8-bit memory manipulation instruction and can be read many times, but can be written only once after reset is released.

$\overline{\text{RESET}}$ input sets WDTM to 67H.

Figure 10-2. Format of Watchdog Timer Mode Register (WDTM)

Address: FF98H After reset: 67H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4[Note 1]	WDCS3[Note 1]	Operation clock selection
0	0	Ring-OSC clock (f_R)
0	1	High-speed system clock (f_{XP})
1	×	Watchdog timer operation stopped

WDCS2[Note 2]	WDCS1[Note 2]	WDCS0[Note 2]	Overflow time setting	
			During Ring-OSC clock operation	During high-speed system clock operation
0	0	0	$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (819.2 μs)
0	0	1	$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (1.64 ms)
0	1	0	$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (3.28 ms)
0	1	1	$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (6.55 ms)
1	0	0	$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (13.11 ms)
1	0	1	$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (26.21 ms)
1	1	0	$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (52.43 ms)
1	1	1	$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (104.86 ms)

Notes 1. If "Ring-OSC cannot be stopped" is specified by the option byte, this cannot be set. The Ring-OSC clock will be selected no matter what value is written.

2. Reset is released at the maximum cycle (WDCS2, 1, 0 = 1, 1, 1).

Cautions 1. If data is written to WDTM, a wait cycle is generated. Do not write data to WDTM when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

2. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when "Ring-OSC cannot be stopped" is selected by the option byte, other values are ignored).

3. After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing is attempted a second time, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

4. WDTM cannot be set by a 1-bit memory manipulation instruction.

5. If "Ring-OSC can be stopped by software" is selected by the option byte and the watchdog timer is stopped by setting WDCS4 to 1, the watchdog timer does not resume operation even if WDCS4 is cleared to 0. In addition, the internal reset signal is not generated.

Remarks 1. f_R: Ring-OSC clock oscillation frequency

2. f_{XP}: High-speed system clock oscillation frequency

3. ×: don't care

4. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 10 MHz.

(2) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

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

$\overline{\text{RESET}}$ input sets WDTE to 9AH.

Figure 10-3. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H After reset: 9AH R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. If a value other than ACH is written to WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated. If the source clock to the watchdog timer is stopped, however, an internal reset signal is generated when the source clock to the watchdog timer resumes operation.

3. The value read from WDTE is 9AH (this differs from the written value (ACH)).

The relationship between the watchdog timer operation and the internal reset signal generated by the watchdog timer is shown below.

Table 10-4. Relationship Between Watchdog Timer Operation and Internal Reset Signal Generated by Watchdog Timer

Watchdog Timer Operation Internal Reset Signal Generation Cause	“Ring-OSC Cannot Be Stopped by Software” Is Selected by Option Byte (Watchdog Timer Is Always Operating)	“Ring-OSC Can Be Stopped by Software” Is Selected by Option Byte		
		Watchdog Timer Is Operating	Watchdog Timer Stopped	
			WDCS4 Is Set to 1	Source Clock to Watchdog Timer Is Stopped
Watchdog timer overflows	Internal reset signal is generated.	Internal reset signal is generated.	–	–
Write to WDTM for the second time	Internal reset signal is generated.	Internal reset signal is generated.	Internal reset signal is not generated and the watchdog timer does not resume operation.	Internal reset signal is generated when the source clock to the watchdog timer resumes operation.
Write other than “ACH” to WDTE	Internal reset signal is generated.	Internal reset signal is generated.	Internal reset signal is not generated.	Internal reset signal is generated when the source clock to the watchdog timer resumes operation.
Access WDTE by 1-bit memory manipulation instruction				

10.4 Operation of Watchdog Timer

10.4.1 Watchdog timer operation when "Ring-OSC cannot be stopped" is selected by option byte

The operation clock of watchdog timer is fixed to the Ring-OSC.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1). The watchdog timer operation cannot be stopped.

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Ring-OSC clock
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2].
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. The operation clock (Ring-OSC clock) cannot be changed. If any value is written to bits 3 and 4 (WDCS3, WDCS4) of WDTM, it is ignored.

2. As soon as WDTM is written, the counter of the watchdog timer is cleared.

Caution **In this mode, operation of the watchdog timer absolutely cannot be stopped even during STOP instruction execution. For 8-bit timer H1 (TMH1), a division of the Ring-OSC can be selected as the count source, so clear the watchdog timer using the interrupt request of TMH1 before the watchdog timer overflows after STOP instruction execution. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.**

10.4.2 Watchdog timer operation when “Ring-OSC can be stopped by software” is selected by option byte

The operation clock of the watchdog timer can be selected as either the Ring-OSC clock or the high-speed system clock.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1).

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Ring-OSC clock
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2, 3].
   - Operation clock: Any of the following can be selected using bits 3 and 4 (WDCS3 and WDCS4).
     Ring-OSC clock (f_R)
     High-speed system clock (f_{XP})
     Watchdog timer operation stopped
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. As soon as WDTM is written, the counter of the watchdog timer is cleared.

2. Set bits 7, 6, and 5 to 0, 1, 1, respectively. Do not set the other values.

3. If the watchdog timer is stopped by setting WDCS4 and WDCS3 to 1 and ×, respectively, an internal reset signal is not generated even if the following processing is performed.

- WDTM is written a second time.
- A 1-bit memory manipulation instruction is executed to WDTE.
- A value other than ACH is written to WDTE.

Caution In this mode, watchdog timer operation is stopped during HALT/STOP instruction execution. After HALT/STOP mode is released, counting is started again using the operation clock of the watchdog timer set before HALT/STOP instruction execution by WDTM. At this time, the counter is not cleared to 0 but holds its value.

For the watchdog timer operation during STOP mode and HALT mode in each status, refer to **10.4.3 Watchdog timer operation in STOP mode** and **10.4.4 Watchdog timer operation in HALT mode**.

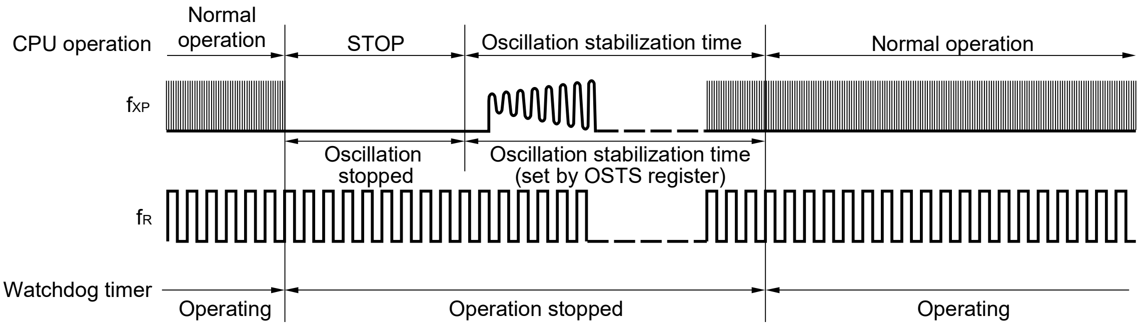
10.4.3 Watchdog timer operation in STOP mode (when “Ring-OSC can be stopped by software” is selected by option byte)

The watchdog timer stops counting during STOP instruction execution regardless of whether the high-speed system clock or Ring-OSC clock is being used.

(1) When the CPU clock and the watchdog timer operation clock are the high-speed system clock (f_{XP}) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting stops for the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) and then counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

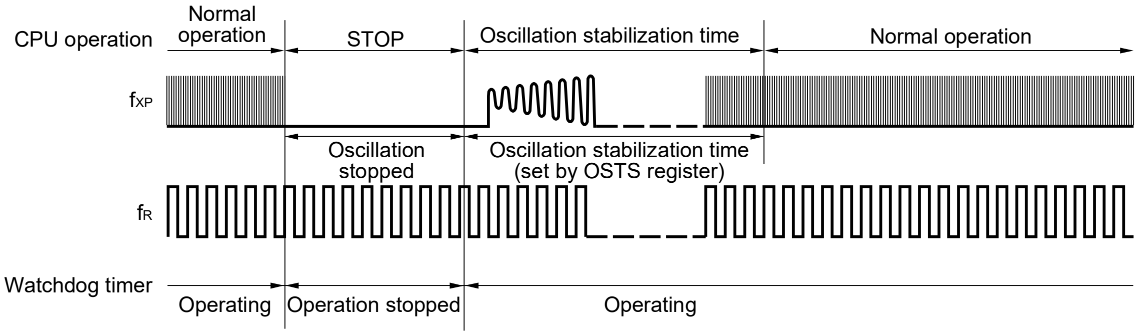
Figure 10-4. Operation in STOP Mode (CPU Clock and WDT Operation Clock: High-Speed System Clock)



(2) When the CPU clock is the high-speed system clock (f_{XP}) and the watchdog timer operation clock is the Ring-OSC clock (f_R) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 10-5. Operation in STOP Mode
(CPU Clock: High-Speed System Clock, WDT Operation Clock: Ring-OSC Clock)



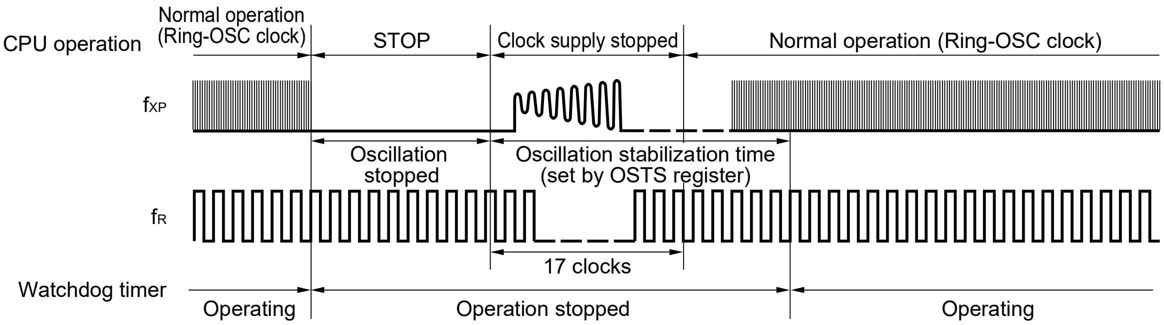
(3) When the CPU clock is the Ring-OSC clock (f_R) and the watchdog timer operation clock is the high-speed system clock (f_{XP}) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is stopped until the timing of <1> or <2>, whichever is earlier, and then counting is started using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

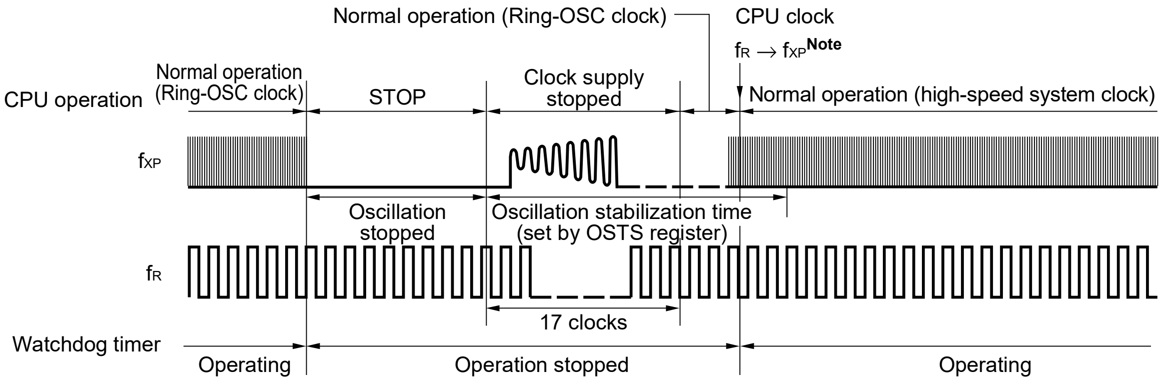
<1> The oscillation stabilization time set by the oscillation stabilization time select register (OSTS) elapses.
<2> The CPU clock is switched to the high-speed system clock (f_{XP}).

Figure 10-6. Operation in STOP Mode
(CPU Clock: Ring-OSC Clock, WDT Operation Clock: High-Speed System Clock)

<1> Timing when counting is started after the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) has elapsed


CPU operation
Normal operation (Ring-OSC clock)
STOP
Clock supply stopped
Normal operation (Ring-OSC clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

<2> Timing when counting is started after the CPU clock is switched to the high-speed system clock (f_{XP})

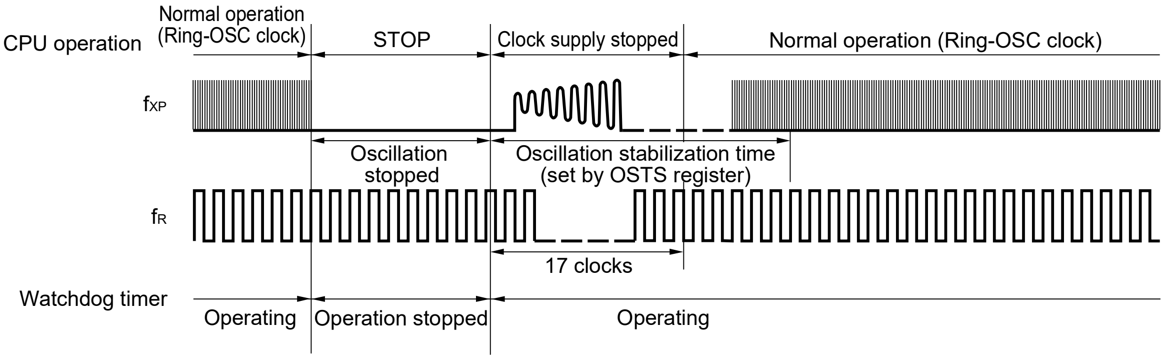

Normal operation (Ring-OSC clock)
CPU clock
fR → fXPNote
CPU operation
Normal operation (Ring-OSC clock)
STOP
Clock supply stopped
Normal operation (high-speed system clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

Note Confirm the oscillation stabilization time of f_{XP} using the oscillation stabilization time counter status register (OSTC).

(4) When CPU clock and watchdog timer operation clock are the Ring-OSC clocks (f_R) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

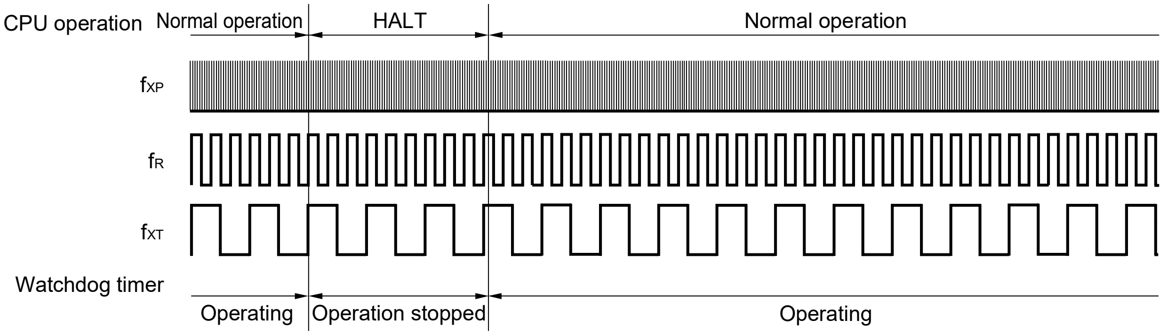
Figure 10-7. Operation in STOP Mode (CPU Clock and WDT Operation Clock: Ring-OSC Clock)


CPU operation
Normal operation (Ring-OSC clock)
STOP
Clock supply stopped
Normal operation (Ring-OSC clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

10.4.4 Watchdog timer operation in HALT mode (when “Ring-OSC can be stopped by software” is selected by option byte)

The watchdog timer stops counting during HALT instruction execution regardless of whether the CPU clock is the high-speed system clock (f_{XP}), Ring-OSC clock (f_R), or subsystem clock (f_{XT}), or whether the operation clock of the watchdog timer is the high-speed system clock (f_{XP}) or Ring-OSC clock (f_R). After HALT mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 10-8. Operation in HALT Mode



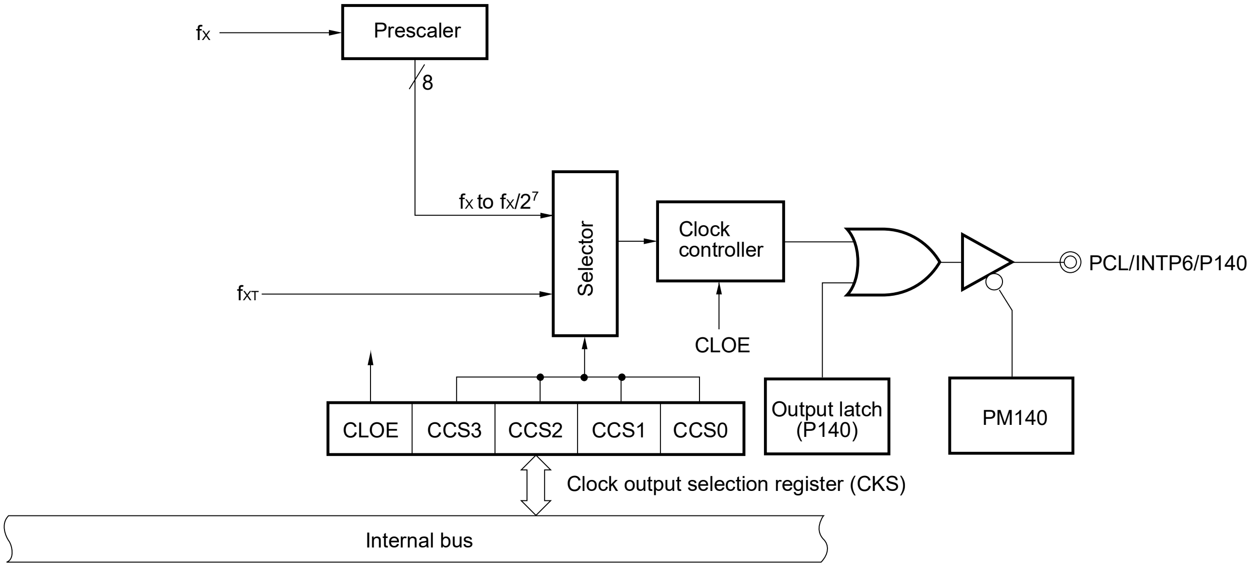
CHAPTER 11 CLOCK OUTPUT CONTROLLER

11.1 Functions of Clock Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

Figure 11-1 shows the block diagram of clock output controller.

Figure 11-1. Block Diagram of Clock Output Controller



11.2 Configuration of Clock Output Controller

The clock output controller includes the following hardware.

Table 11-1. Clock Output Controller Configuration

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 14 (PM14) Port register 14 (P14)

11.3 Registers Controlling Clock Output Controller

The following two registers are used to control the clock output controller.

- Clock output selection register (CKS)
- Port mode register 14 (PM14)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and sets the output clock.

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

$\overline{\text{RESET}}$ input clears CKS to 00H.

Figure 11-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (10 MHz)
0	0	0	1	$f_X/2$ (5 MHz)
0	0	1	0	$f_X/2^2$ (2.5 MHz)
0	0	1	1	$f_X/2^3$ (1.25 MHz)
0	1	0	0	$f_X/2^4$ (625 kHz)
0	1	0	1	$f_X/2^5$ (312.5 kHz)
0	1	1	0	$f_X/2^6$ (156.25 kHz)
0	1	1	1	$f_X/2^7$ (78.125 kHz)
1	0	0	0	f_{XT} (32.768 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: High-speed system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. Figures in parentheses are for operation with f_X = 10 MHz or f_{XT} = 32.768 kHz.

(2) Port mode register 14 (PM14)

This register sets port 14 input/output in 1-bit units.

When using the P140/INTP6/PCL pin for clock output, set PM140 and the output latch of P140 to 0.

PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM14 to FFH.

Figure 11-3. Format of Port Mode Register 14 (PM14)

Address: FF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

PM140	P140 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

11.4 Clock Output Controller Operations

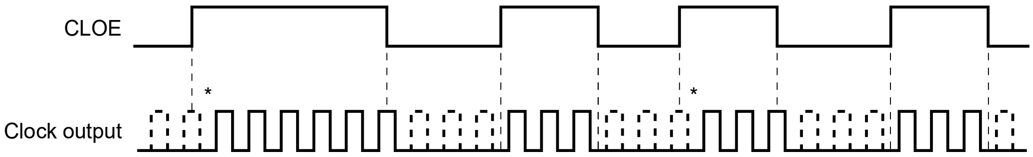
The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 11-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 11-4. Remote Control Output Application Example



CHAPTER 12 A/D CONVERTER

12.1 Functions of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

The A/D converter has the following two functions.

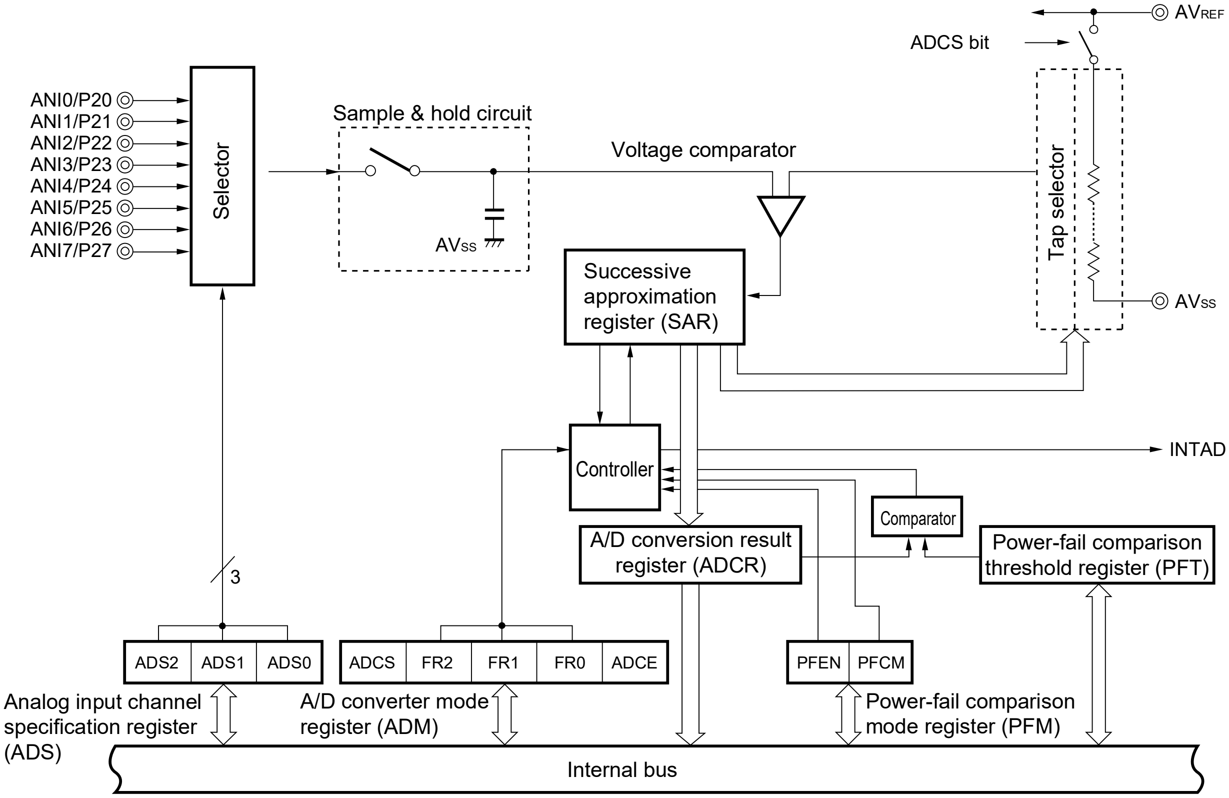
(1) 10-bit resolution A/D conversion

10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

(2) Power-fail detection function

This function is used to detect a voltage drop in a battery. The A/D conversion result (ADCR register value) and power-fail comparison threshold register (PFT) value are compared. INTAD is generated only when a comparative condition has been matched.

Figure 12-1. Block Diagram of A/D Converter


AVREF
ADCS bit
ANI0/P20
ANI1/P21
ANI2/P22
ANI3/P23
ANI4/P24
ANI5/P25
ANI6/P26
ANI7/P27
Selector
Sample & hold circuit
AVSS
Voltage comparator
Tap selector
AVSS
Successive approximation register (SAR)
Controller
INTAD
Comparator
A/D conversion result register (ADCR)
Power-fail comparison threshold register (PFT)
3
ADS2
ADS1
ADS0
ADCS
FR2
FR1
FR0
ADCE
PFEN
PFCM
Analog input channel specification register (ADS)
A/D converter mode register (ADM)
Power-fail comparison mode register (PFM)
Internal bus

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

Table 12-1. Registers of A/D Converter Used on Software

Item	Configuration
Registers	A/D conversion result register (ADCR) A/D converter mode register (ADM) Analog input channel specification register (ADS) Power-fail comparison mode register (PFM) Power-fail comparison threshold register (PFT)

(1) ANI0 to ANI7 pins

These are the analog input pins of the 8-channel A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin by the analog input channel specification register (ADS) can be used as input port pins.

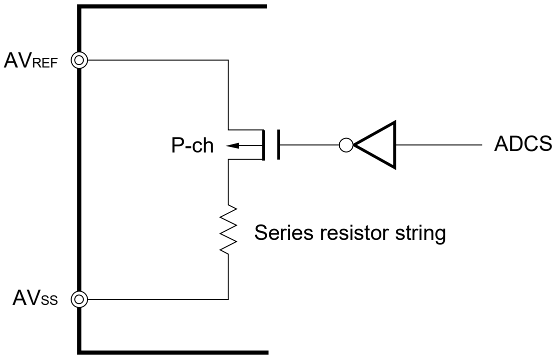
(2) Sample & hold circuit

The sample & hold circuit samples the input signal of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled analog input voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the analog input signal.

Figure 12-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled analog input voltage and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register compares the sampled analog voltage and the voltage of the series resistor string, and converts the result, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) A/D conversion result register (ADCR)

The result of A/D conversion is loaded from the successive approximation register (SAR) to this register each time A/D conversion is completed, and the ADCR register holds the result of A/D conversion in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) Controller

When A/D conversion has been completed or when the power-fail detection function is used, this controller compares the result of A/D conversion (value of the ADCR register) and the value of the power-fail comparison threshold register (PFT). It generates the interrupt INTAD only if a specified comparison condition is satisfied as a result.

(8) AV_{REF} pin

This pin inputs an analog power/reference voltage to the A/D converter. Always use this pin at the same potential as that of the V_{DD} pin even when the A/D converter is not used.

The signal input to ANI0 to ANI7 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(9) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

(10) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

(11) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(12) Power-fail comparison mode register (PFM)

This register is used to set the power-fail monitor mode.

(13) Power-fail comparison threshold register (PFT)

This register is used to set the threshold value that is to be compared with the value of the A/D conversion result register (ADCR).

12.3 Registers Used in A/D Converter

The following five registers are used to control the A/D converter.

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- A/D conversion result register (ADCR)
- Power-fail comparison mode register (PFM)
- Power-fail comparison threshold register (PFT)

(1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.
ADM can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears ADM to 00H.

Figure 12-3. Format of A/D Converter Mode Register (ADM)

Address: FF28H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2	FR1	FR0	0	0	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

FR2	FR1	FR0	Conversion time selection[Note 1]				
				fx = 2 MHz	fx = 8.38 MHz	fx = 10 MHz	fx = 16 MHz
0	0	0	288/fx	144 μs[Note 1]	34.3 μs	28.8 μs	18 μs
0	0	1	240/fx	120 μs[Note 1]	28.6 μs	24.0 μs	15 μs
0	1	0	192/fx	96 μs	22.9 μs	19.2 μs	12 μs[Note 1]
1	0	0	144/fx	72 μs	17.2 μs	14.4 μs	9 μs[Note 1]
1	0	1	120/fx	60 μs	14.3 μs	12.0 μs[Note 1]	7.5 μs[Note 1]
1	1	0	96/fx	48 μs	11.5 μs[Note 1]	9.6 μs[Note 1]	6 μs[Note 1]
Other than above			Setting prohibited				

ADCE	Boost reference voltage generator operation control[Note 2]
0	Stops operation of reference voltage generator
1	Enables operation of reference voltage generator

Notes 1. Set so that the A/D conversion time is 14 μs or longer but less than 100 μs.

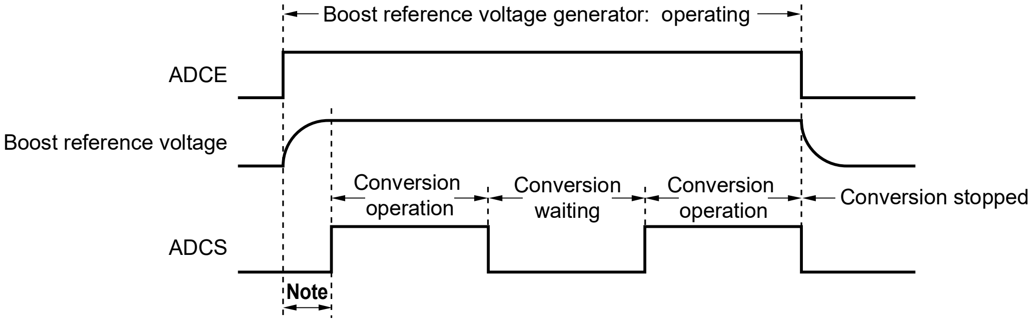
2. A booster circuit is incorporated to realize low-voltage operation. The operation of the circuit that generates the reference voltage for boosting is controlled by ADCE, and it takes 14 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 14 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result.

Table 12-2. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only reference voltage generator consumes power)
1	0	Conversion mode (reference voltage generator operation stopped[Note])
1	1	Conversion mode (reference voltage generator operates)

Note Data of first conversion cannot be used.

Figure 12-4. Timing Chart When Boost Reference Voltage Generator Is Used



Note The time from the rising of the ADCE bit to the rising of the ADCS bit must be 14 μs or longer to stabilize the reference voltage.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2 to values other than the identical data.

2. For the sampling time of the A/D converter and the A/D conversion start delay time, refer to (11) in 12.6 Cautions for A/D Converter.

3. If data is written to ADM, a wait cycle is generated. Do not write data to ADM when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

Remark fx: High-speed system clock oscillation frequency

(2) Analog input channel specification register (ADS)

This register specifies the input port of the analog voltage to be A/D converted.

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

$\overline{\text{RESET}}$ input clears ADS to 00H.

Figure 12-5. Format of Analog Input Channel Specification Register (ADS)

Address: FF29H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	Analog input channel specification
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

Cautions 1. Be sure to clear bits 3 to 7 of ADS to 0.

2. If data is written to ADS, a wait cycle is generated. Do not write data to ADS when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

(3) A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower six bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and is stored in ADCR in order starting from the most significant bit (MSB). FF09H indicates the higher 8 bits of the conversion result, and FF08H indicates the lower 2 bits of the conversion result.

ADCR can be read by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

Figure 12-6. Format of A/D Conversion Result Register (ADCR)

Address: FF08H, FF09H After reset: Undefined R

Symbol	FF09H								FF08H							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. Do not read data from ADCR when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

(4) Power-fail comparison mode register (PFM)

The power-fail comparison mode register (PFM) is used to compare the A/D conversion result (value of the ADCR register) and the value of the power-fail comparison threshold register (PFT).

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

$\overline{\text{RESET}}$ input clears PFM to 00H.

Figure 12-7. Format of Power-Fail Comparison Mode Register (PFM)

Address: FF2AH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	Power-fail comparison enable
0	Stops power-fail comparison (used as a normal A/D converter)
1	Enables power-fail comparison (used for power-fail detection)

PFCM		Power-fail comparison mode selection
0	Higher 8 bits of ADCR ≥ PFT	Interrupt request signal (INTAD) generation
	Higher 8 bits of ADCR < PFT	No INTAD generation
1	Higher 8 bits of ADCR ≥ PFT	No INTAD generation
	Higher 8 bits of ADCR < PFT	INTAD generation

Caution If data is written to PFM, a wait cycle is generated. Do not write data to PFM when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

(5) Power-fail comparison threshold register (PFT)

The power-fail comparison threshold register (PFT) is a register that sets the threshold value when comparing the values with the A/D conversion result.

8-bit data in PFT is compared to the higher 8 bits (FF09H) of the 10-bit A/D conversion result.

PFT can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PFT to 00H.

Figure 12-8. Format of Power-Fail Comparison Threshold Register (PFT)

Address: FF2BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

Caution If data is written to PFT, a wait cycle is generated. Do not write data to PFT when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

12.4 A/D Converter Operations

12.4.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<2> Set ADCE to 1 and wait for 14 μs or longer.

<3> Set ADCS to 1 and start the conversion operation.
(<4> to <10> are operations performed by hardware.)

<4> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation has ended.

<6> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<7> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<8> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

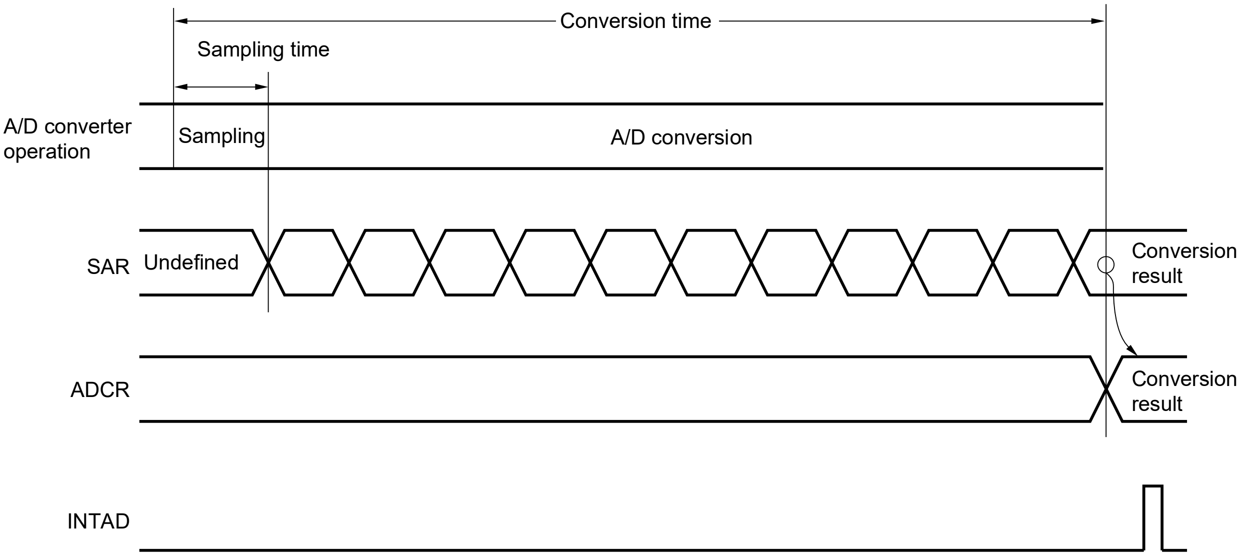
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<9> Comparison is continued in this way up to bit 0 of SAR.

<10> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<11> Repeat steps <4> to <10>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <3>. To restart A/D conversion from the status of ADCE = 0, however, start from <2>.

Figure 12-9. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to one of the ADM, analog input channel specification register (ADS), power-fail comparison mode register (PFM), or power-fail comparison threshold register (PFT) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input makes the A/D conversion result register (ADCR) undefined.

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the theoretical A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

or

$$(\mathrm{ADCR} - 0.5) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < (\mathrm{ADCR} + 0.5) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

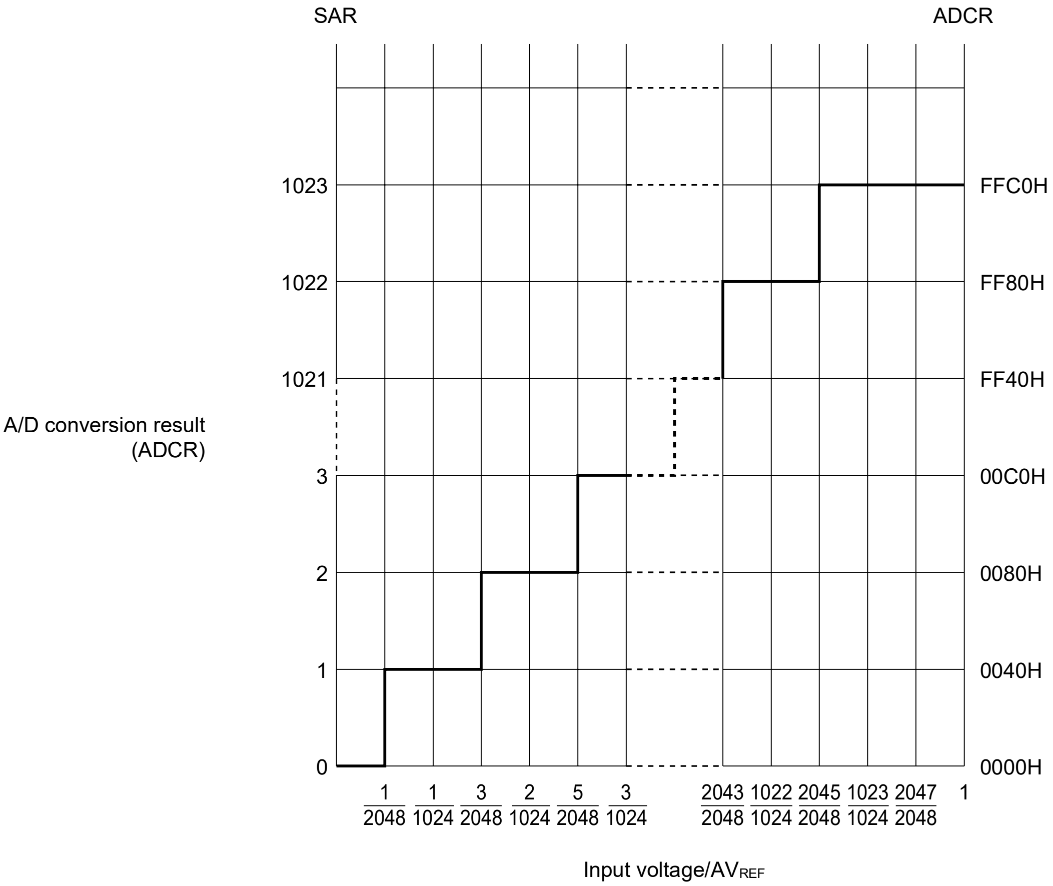
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-10 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-10. Relationship Between Analog Input Voltage and A/D Conversion Result



12.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI7 by the analog input channel specification register (ADS) and A/D conversion is executed.

In addition, the following two functions can be selected by setting bit 7 (PFEN) of the power-fail comparison mode register (PFM).

- Normal 10-bit A/D converter (PFEN = 0)
- Power-fail detection function (PFEN = 1)

(1) A/D conversion operation (when PFEN = 0)

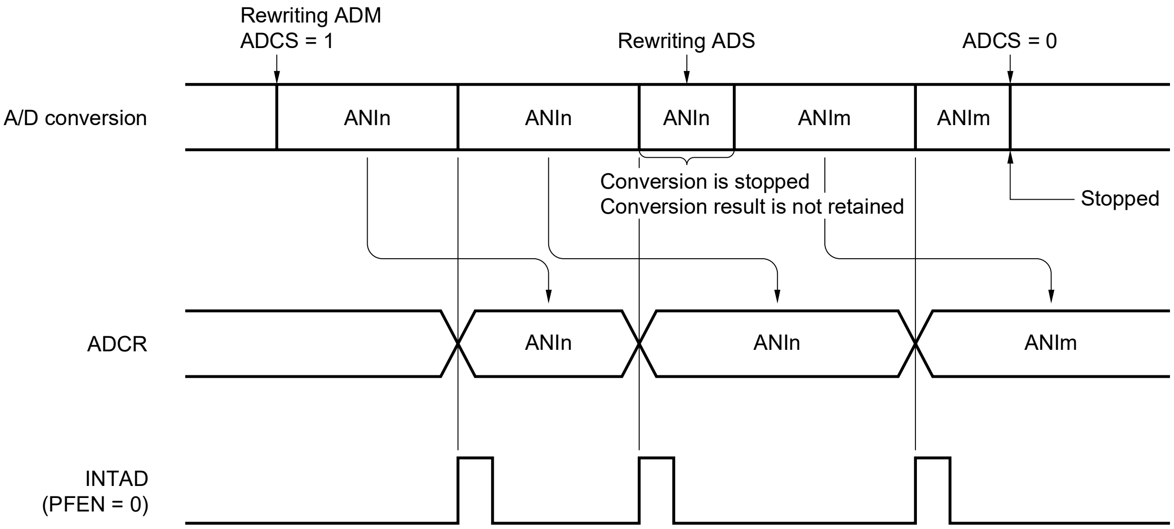
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 0, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has started and when one A/D conversion has been completed, the next A/D conversion operation is immediately started. The A/D conversion operations are repeated until new data is written to ADS.

If ADM, ADS, the power-fail comparison mode register (PFM), and the power-fail comparison threshold register (PFT) are rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result is undefined.

Figure 12-11. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(2) Power-fail detection function (when PFEN = 1)

By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 1, the A/D conversion operation of the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) is started.

When the A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), the values are compared with power-fail comparison threshold register (PFT), and an interrupt request signal (INTAD) is generated under the condition specified by bit 6 (PFCM) of PFM.

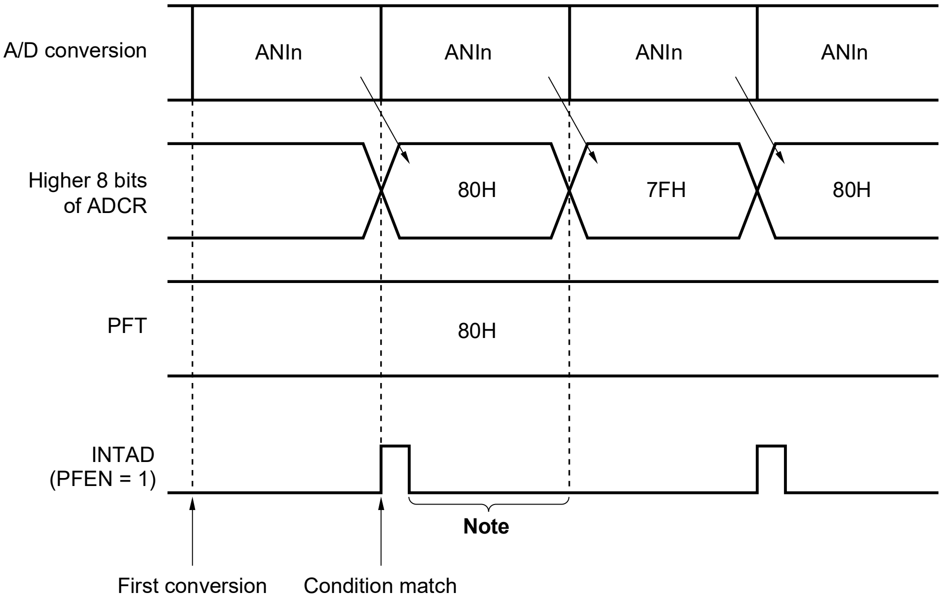
<1> When PFEN = 1 and PFCM = 0

The higher 8 bits of ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when the higher 8 bits of ADCR ≥ PFT.

<2> When PFEN = 1 and PFCM = 1

The higher 8 bits of ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when the higher 8 bits of ADCR < PFT.

Figure 12-12. Power-Fail Detection (When PFEN = 1 and PFCM = 0)



Note If the conversion result is not read before the end of the next conversion after INTAD is output, the result is replaced by the next conversion result.

Remark n = 0 to 7

The setting methods are described below.

- When used as A/D conversion operation

<1> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.

<2> Select the channel and conversion time using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.

<3> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.

<4> An interrupt request signal (INTAD) is generated.

<5> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<Change the channel>

<6> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS to start A/D conversion.

<7> An interrupt request signal (INTAD) is generated.

<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<Complete A/D conversion>

<9> Clear ADCS to 0.

<10> Clear ADCE to 0.

Cautions 1. Make sure the period of <1> to <3> is 14 μs or more.

2. It is no problem if the order of <1> and <2> is reversed.

3. <1> can be omitted. However, do not use the first conversion result after <3> in this case.

4. The period from <4> to <7> differs from the conversion time set using bits 5 to 3 (FR2 to FR0) of ADM. The period from <6> to <7> is the conversion time set using FR2 to FR0.

- When used as power-fail function

<1> Set bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 1.

<2> Set power-fail comparison condition using bit 6 (PFCM) of PFM.

<3> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.

<4> Select the channel and conversion time using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 3 (FR2 to FR0) of ADM.

<5> Set a threshold value to the power-fail comparison threshold register (PFT).

<6> Set bit 7 (ADCS) of ADM to 1.

<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<8> The higher 8 bits of ADCR and PFT are compared and an interrupt request signal (INTAD) is generated if the conditions match.

<Change the channel>

<9> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS.

<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<11> The higher 8 bits of ADCR and the power-fail comparison threshold register (PFT) are compared and an interrupt request signal (INTAD) is generated if the conditions match.

<Complete A/D conversion>

<12> Clear ADCS to 0.

<13> Clear ADCE to 0.

Cautions 1. Make sure the period of <3> to <6> is 14 μs or more.

2. It is no problem if order of <3>, <4>, and <5> is changed.

3. <3> must not be omitted if the power-fail function is used.

4. The period from <7> to <11> differs from the conversion time set using bits 5 to 3 (FR2 to FR0) of ADM. The period from <9> to <11> is the conversion time set using FR2 to FR0.

12.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$\begin{aligned}1\text{LSB} &= 1/2^{10} = 1/1024\\ &= 0.098\%\text{FSR}\end{aligned}$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12-13. Overall Error

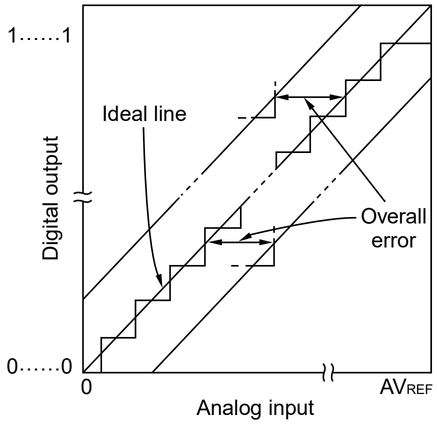
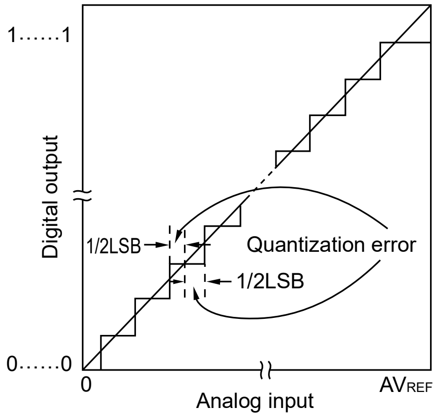


Figure 12-14. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 12-15. Zero-Scale Error

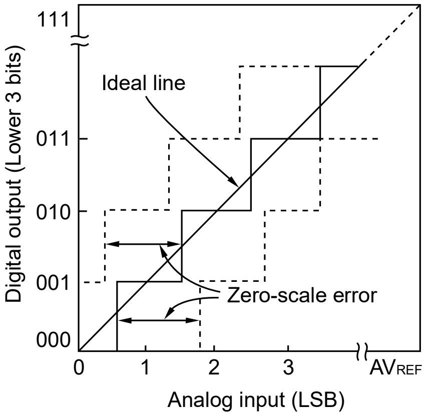


Figure 12-16. Full-Scale Error

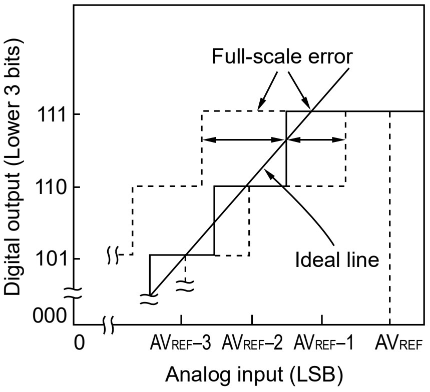


Figure 12-17. Integral Linearity Error

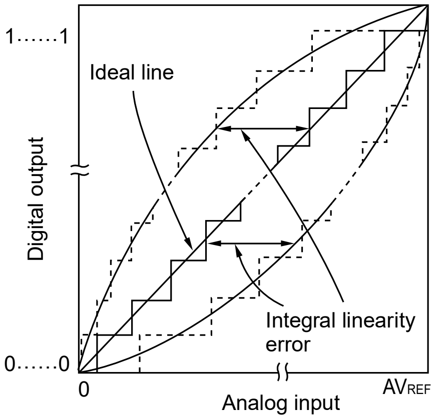
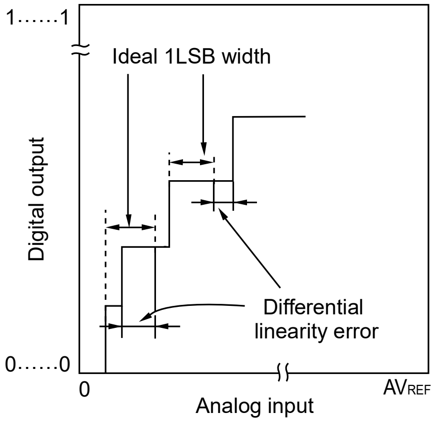


Figure 12-18. Differential Linearity Error



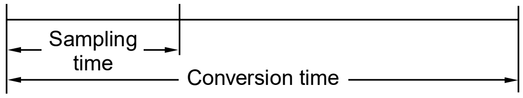
(8) Conversion time

This expresses the time since sampling has been started until digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



12.6 Cautions for A/D Converter

(1) Operating current in standby mode

The A/D converter stops operating in the standby mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 (refer to **Figure 12-2**).

(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR) write and ADCR read by instruction upon the end of conversion

ADCR read has priority. After the read operation, the new conversion result is written to ADCR.

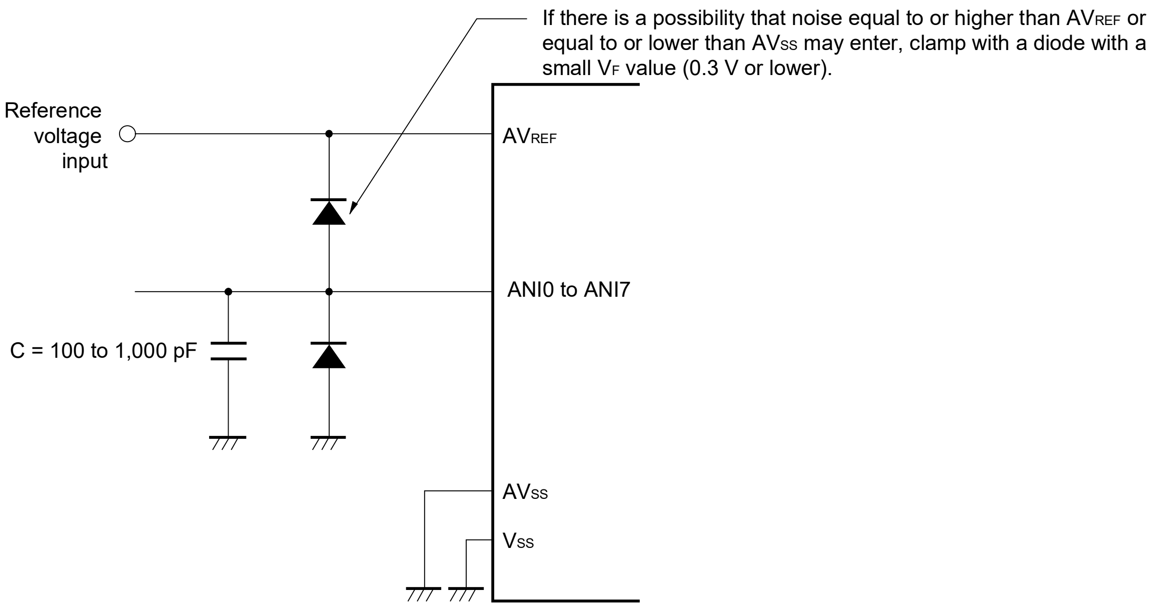
<2> Conflict between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write upon the end of conversion

ADM or ADS write has priority. ADCR write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AVREF pin and pins ANI0 to ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally, as shown in Figure 12-19, to reduce noise.

Figure 12-19. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI0 to ANI7
C = 100 to 1,000 pF
AVSS
VSS

(5) ANI0/P20 to ANI7/P27

<1> The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).
When A/D conversion is performed with any of ANI0 to ANI7 selected, do not access port 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI7 pins

In this A/D converter, the internal sampling capacitor is charged and sampling is performed for approx. one sixth of the conversion time.

Since only the leakage current flows other than during sampling and the current for charging the capacitor also flows during sampling, the input impedance fluctuates and has no meaning.

To perform sufficient sampling, however, it is recommended to make the output impedance of the analog input source 10 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI7 pins (refer to **Figure 12-19**).

(7) AVREF pin input impedance

A series resistor string of several tens of 10 kΩ is connected between the AVREF and AVSS pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AVREF and AVSS pins, resulting in a large reference voltage error.

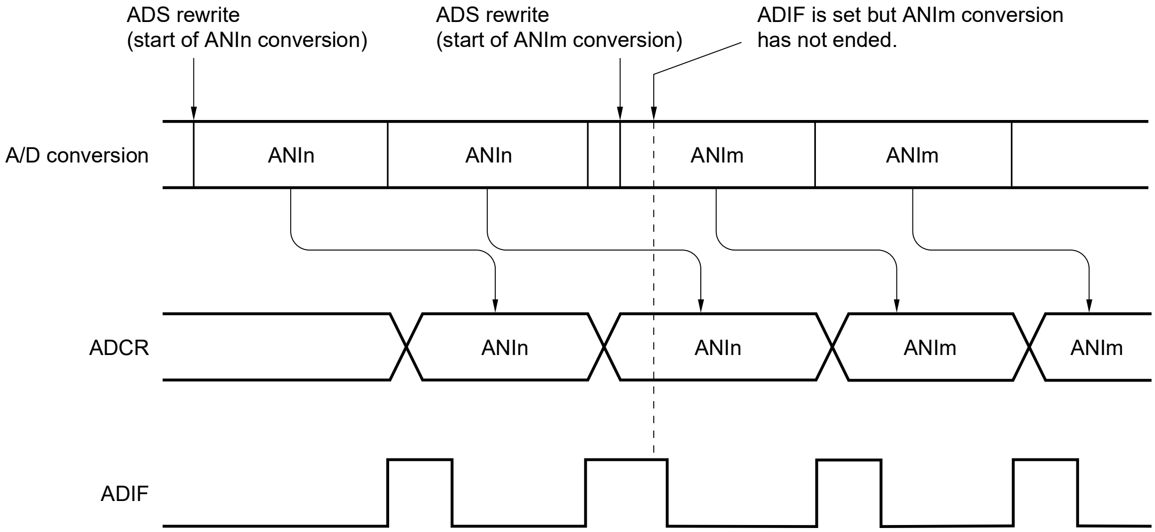
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 12-20. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(9) Conversion results just after A/D conversion start

The A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 14 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR) read operation

When a write operation is performed to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) A/D converter sampling time and A/D conversion start delay time

The A/D converter sampling time differs depending on the set value of the A/D converter mode register (ADM). The delay time exists until actual sampling is started after A/D converter operation is enabled.

When using a set in which the A/D conversion time must be strictly observed, care is required for the contents shown in Figure 12-21 and Table 12-3.

Figure 12-21. Timing of A/D Converter Sampling and A/D Conversion Start Delay

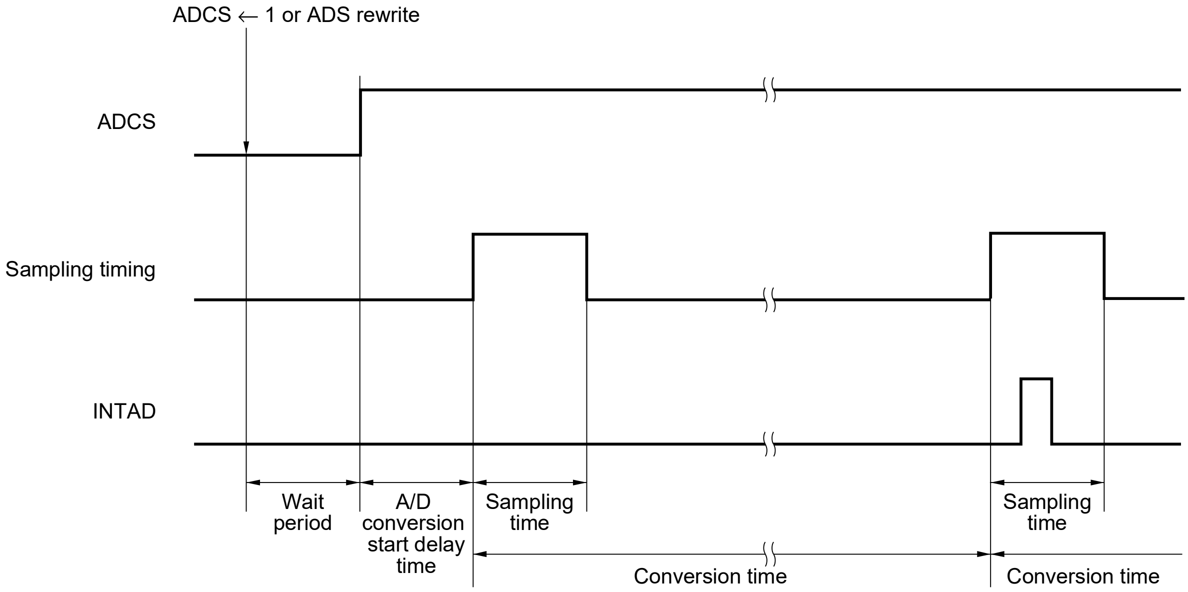


Table 12-3. A/D Converter Sampling Time and A/D Conversion Start Delay Time (ADM Set Value)

FR2	FR1	FR0	Conversion Time	Sampling Time	A/D Conversion Start Delay Time[Note]	
					MIN.	MAX.
0	0	0	288/fx	40/fx	32/fx	36/fx
0	0	1	240/fx	32/fx	28/fx	32/fx
0	1	0	192/fx	24/fx	24/fx	28/fx
1	0	0	144/fx	20/fx	16/fx	18/fx
1	0	1	120/fx	16/fx	14/fx	16/fx
1	1	0	96/fx	12/fx	12/fx	14/fx
Other than above			Setting prohibited	–	–	–

Note The A/D conversion start delay time is the time after wait period. For the wait function, refer to **CHAPTER 29 CAUTIONS FOR WAIT**.

Remark fx: High-speed system clock oscillation frequency

(12) Register generating wait cycle

Do not read data from the ADCR register and do not write data to the ADM, ADS, PFM, and PFT registers while the CPU is operating on the subsystem clock and while high-speed system clock oscillation is stopped.

(13) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12-22. Internal Equivalent Circuit of ANIn Pin

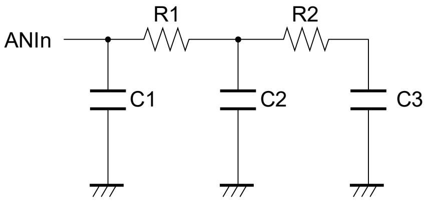


Table 12-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	R2	C1	C2	C3
2.7 V	12 kΩ	8 kΩ	8 pF	3 pF	0.6 pF
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	0.6 pF

Remarks 1. The resistance and capacitance values shown in Table 12-4 are not guaranteed values.

2. n = 0 to 7

CHAPTER 13 SERIAL INTERFACE UART0

13.1 Functions of Serial Interface UART0

Serial interface UART0 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, refer to **13.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.

For details, refer to **13.4.2 Asynchronous serial interface (UART) mode** and **13.4.3 Dedicated baud rate generator**.

- Two-pin configuration TxD0: Transmit data output pin
  RxD0: Receive data input pin
- Length of communication data can be selected from 7 or 8 bits.
- Dedicated on-chip 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Four operating clock inputs selectable
- Fixed to LSB-first communication

Cautions 1. If clock supply to serial interface UART0 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART0 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD0 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER0 = 0, RXE0 = 0, and TXE0 = 0.

2. Set POWER0 = 1 and then set TXE0 = 1 (transmission) or RXE0 = 1 (reception) to start communication.

3. TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

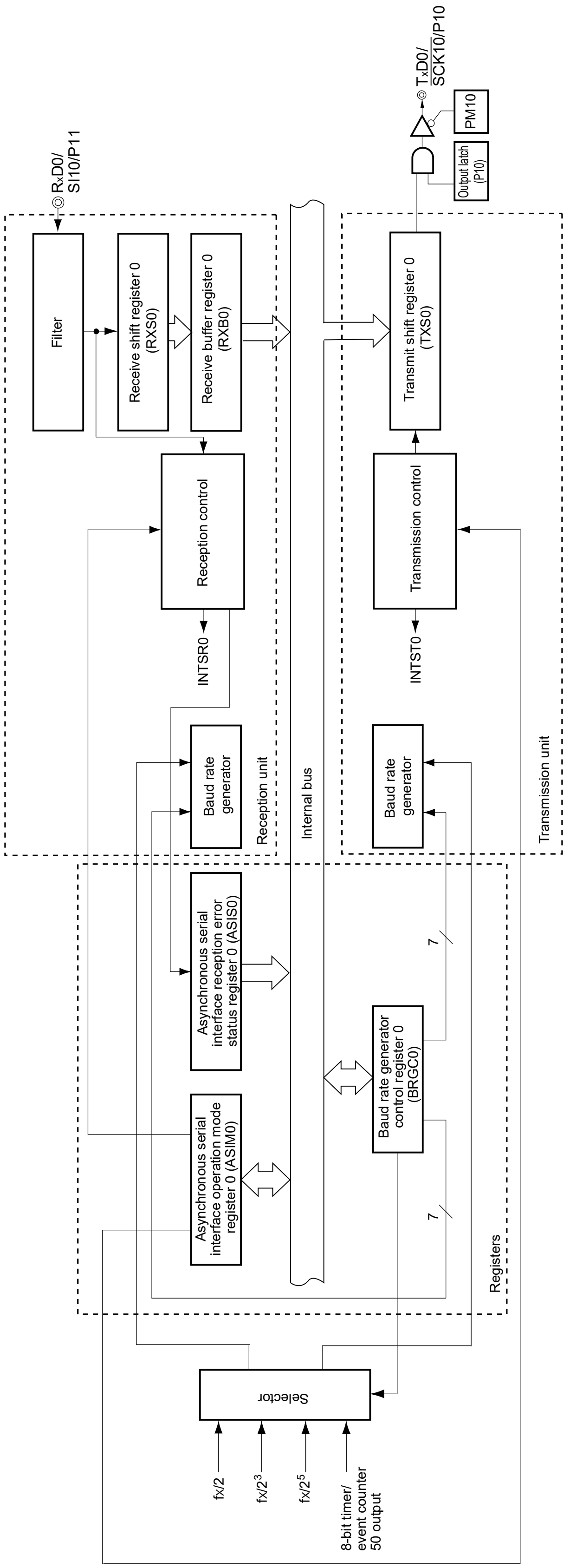
13.2 Configuration of Serial Interface UART0

Serial interface UART0 includes the following hardware.

Table 13-1. Configuration of Serial Interface UART0

Item	Configuration
Registers	Receive buffer register 0 (RXB0) Receive shift register 0 (RXS0) Transmit shift register 0 (TXS0)
Control registers	Asynchronous serial interface operation mode register 0 (ASIM0) Asynchronous serial interface reception error status register 0 (ASIS0) Baud rate generator control register 0 (BRGC0) Port mode register 1 (PM1) Port register 1 (P1)

Figure 13-1. Block Diagram of Serial Interface UART0



(1) Receive buffer register 0 (RXB0)

This 8-bit register stores parallel data converted by receive shift register 0 (RXS0).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 0 (RXS0).

If the data length is set to 7 bits, the receive data is transferred to bits 0 to 6 of RXB0 and the MSB of RXB0 is always 0.

If an overrun error (OVE0) occurs, the receive data is not transferred to RXB0.

RXB0 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

$\overline{\text{RESET}}$ input or POWER0 = 0 sets this register to FFH.

(2) Receive shift register 0 (RXS0)

This register converts the serial data input to the RxD0 pin into parallel data.

RXS0 cannot be directly manipulated by a program.

(3) Transmit shift register 0 (TXS0)

This register is used to set transmit data. Transmission is started when data is written to TXS0, and serial data is transmitted from the TxD0 pins.

TXS0 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

$\overline{\text{RESET}}$ input, POWER0 = 0, or TXE0 = 0 sets this register to FFH.

Caution Do not write the next transmit data to TXS0 before the transmission completion interrupt signal (INTST0) is generated.

13.3 Registers Controlling Serial Interface UART0

Serial interface UART0 is controlled by the following five registers.

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 0 (ASIM0)

This 8-bit register controls the serial communication operations of serial interface UART0.

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

$\overline{\text{RESET}}$ input sets ASIM0 to 01H.

Figure 13-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (1/2)

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1	Enables operation of the internal operation clock.

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Figure 13-2. Format of Asynchronous Serial Interface Operation Mode Register 0 (ASIM0) (2/2)

PS01	PS00	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL0	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL0	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE0) of asynchronous serial interface reception error status register 0 (ASIS0) is not set and the error interrupt does not occur.

Cautions 1. At startup, set POWER0 to 1 and then set TXE0 to 1. To stop the operation, clear TXE0 to 0, and then clear POWER0 to 0.

2. At startup, set POWER0 to 1 and then set RXE0 to 1. To stop the operation, clear RXE0 to 0, and then clear POWER0 to 0.

3. Set POWER0 to 1 and then set RXE0 to 1 while a high level is input to the RxD0 pin. If POWER0 is set to 1 and RXE0 is set to 1 while a low level is input, reception is started.

4. TXE0 and RXE0 are synchronized by the base clock (f_{XCLK0}) set by BRGC0. To enable transmission or reception again, set TXE0 or RXE0 to 1 at least two clocks of base clock after TXE0 or RXE0 has been cleared to 0. If TXE0 or RXE0 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

5. Clear the TXE0 and RXE0 bits to 0 before rewriting the PS01, PS00, and CL0 bits.

6. Make sure that TXE0 = 0 when rewriting the SL0 bit. Reception is always performed with “number of stop bits = 1”, and therefore, is not affected by the set value of the SL0 bit.

7. Be sure to set bit 0 to 1.

(2) Asynchronous serial interface reception error status register 0 (ASIS0)

This register indicates an error status on completion of reception by serial interface UART0. It includes three error flag bits (PE0, FE0, OVE0).

This register is read-only by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS0 to 00H if bit 7 (POWER0) and bit 5 (RXE0) of ASIM0 = 0. 00H is read when this register is read.

Figure 13-3. Format of Asynchronous Serial Interface Reception Error Status Register 0 (ASIS0)

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	Status flag indicating parity error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE0	Status flag indicating framing error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If the stop bit is not detected on completion of reception.

OVE0	Status flag indicating overrun error
0	If POWER0 = 0 and RXE0 = 0, or if ASIS0 register is read.
1	If receive data is set to the RXB0 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE0 bit differs depending on the set values of the PS01 and PS00 bits of asynchronous serial interface operation mode register 0 (ASIM0).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 0 (RXB0) but discarded.

4. If data is read from ASIS0, a wait cycle is generated. Do not read data from ASIS0 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

(3) Baud rate generator control register 0 (BRGC0)

This register selects the base clock of serial interface UART0 and the division value of the 5-bit counter.

BRGC0 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC0 to 1FH.

Figure 13-4. Format of Baud Rate Generator Control Register 0 (BRGC0)

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	Base clock (f_{XCLK0}) selection[Note 1]
0	0	TM50 output[Note 2]
0	1	$f_X/2$ (5 MHz)
1	0	$f_X/2^3$ (1.25 MHz)
1	1	$f_X/2^5$ (312.5 kHz)

MDL04	MDL03	MDL02	MDL01	MDL00	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	0	1	0	26	$f_{XCLK0}/26$
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

Notes 1. Be sure to set the base clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Base clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Base clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Base clock ≤ 5 MHz

2. Note the following points when selecting the TM50 output as the base clock.

- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the base clock to make the duty = 50%.
- Mode in which the base clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the base clock is the Ring-OSC clock, the operation of serial interface UART0 is not guaranteed.

2. Make sure that bit 6 (TXE0) and bit 5 (RXE0) of the ASIM0 register = 0 when rewriting the MDL04 to MDL00 bits.

3. The baud rate value is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits

2. f_X: High-speed system clock oscillation frequency

3. k: Value set by the MDL04 to MDL00 bits (k = 8, 9, 10, ..., 31)

4. ×: don’t care

5. Figures in parentheses apply to operation at f_X = 10 MHz.

6. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(4) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P10/TxD0/$\overline{\text{SCK10}}$ pin for serial interface data output, clear PM10 to 0 and set the output latch of P10 to 1.

When using the P11/RxD0/SI10 pin for serial interface data input, set PM11 to 1. The output latch of P11 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM1 to FFH.

Figure 13-5. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

13.4 Operation of Serial Interface UART0

Serial interface UART0 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

13.4.1 Operation stop mode

In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER0, TXE0, and RXE0) of ASIM0 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 0 (ASIM0).
ASIM0 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM0 to 01H.

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE0	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).

RXE0	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The input from the RxD0 pin is fixed to high level when POWER0 = 0.

2. Asynchronous serial interface reception error status register 0 (ASIS0), transmit shift register 0 (TXS0), and receive buffer register 0 (RXB0) are reset.

Caution Clear POWER0 to 0 after clearing TXE0 and RXE0 to 0 to set the operation stop mode.
To start the operation, set POWER0 to 1, and then set TXE0 and RXE0 to 1.

Remark To use the RxD0/SI10/P11 and TxD0/$\overline{\text{SCK10}}$/P10 pins as general-purpose port pins, refer to **CHAPTER 4 PORT FUNCTIONS**.

13.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC0 register (refer to **Figure 13-4**).
<2> Set bits 1 to 4 (SL0, CL0, PS00, and PS01) of the ASIM0 register (refer to **Figure 13-2**).
<3> Set bit 7 (POWER0) of the ASIM0 register to 1.
<4> Set bit 6 (TXE0) of the ASIM0 register to 1. → Transmission is enabled.
Set bit 5 (RXE0) of the ASIM0 register to 1. → Reception is enabled.
<5> Write data to the TXS0 register. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 13-2. Relationship Between Register Settings and Pins

POWER0	TXE0	RXE0	PM10	P10	PM11	P11	UART0 Operation	Pin Function	
								TxD0/$\overline{\text{SCK10}}$/P10	RxD0/SI10/P11
0	0	0	×[Note]	×[Note]	×[Note]	×[Note]	Stop	$\overline{\text{SCK10}}$/P10	SI10/P11
1	0	1	×[Note]	×[Note]	1	×	Reception	$\overline{\text{SCK10}}$/P10	RxD0
	1	0	0	1	×[Note]	×[Note]	Transmission	TxD0	SI10/P11
	1	1	0	1	1	×	Transmission/ reception	TxD0	RxD0

Note Can be set as port function.

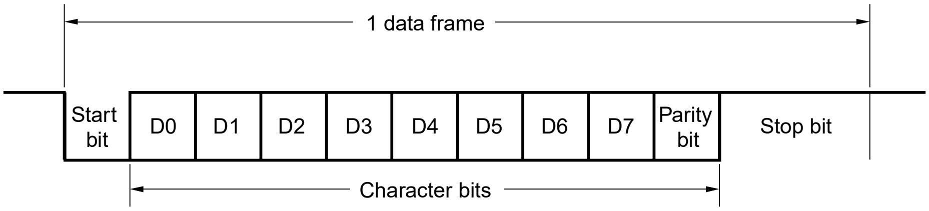
Remark ×: don't care
POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
PM1×: Port mode register
P1×: Port output latch

(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

Figures 13-6 and 13-7 show the format and waveform example of the normal transmit/receive data.

Figure 13-6. Format of Normal UART Transmit/Receive Data



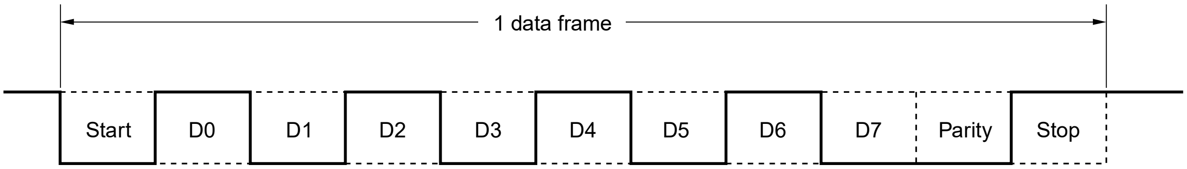
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

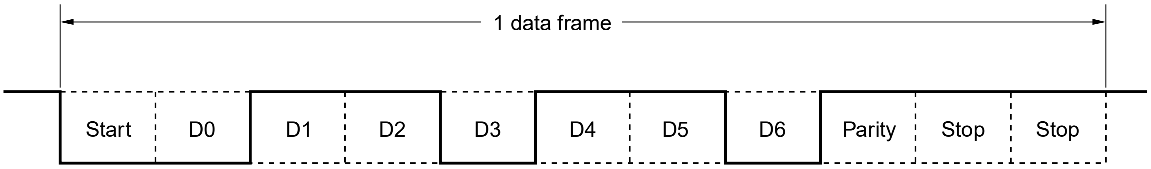
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 0 (ASIM0).

Figure 13-7. Example of Normal UART Transmit/Receive Data Waveform

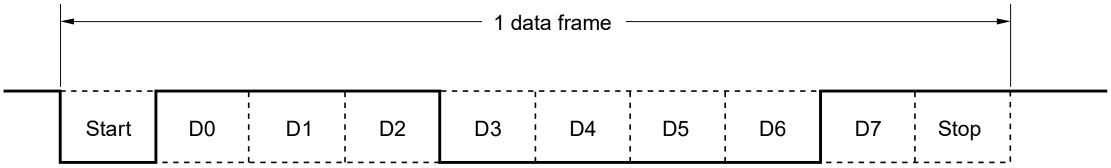
1. **Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



2. **Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H**



3. **Data length: 8 bits, Parity: None, Stop bit: 1 bit, Communication data: 87H**



(b) 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 transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 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”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Transmission

The TxD0 pin outputs a high level when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1. If bit 6 (TXE0) of ASIM0 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 0 (TXS0). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the start bit is output from the TxD0 pin, followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM0 are appended and a transmission completion interrupt request (INTST0) is generated.

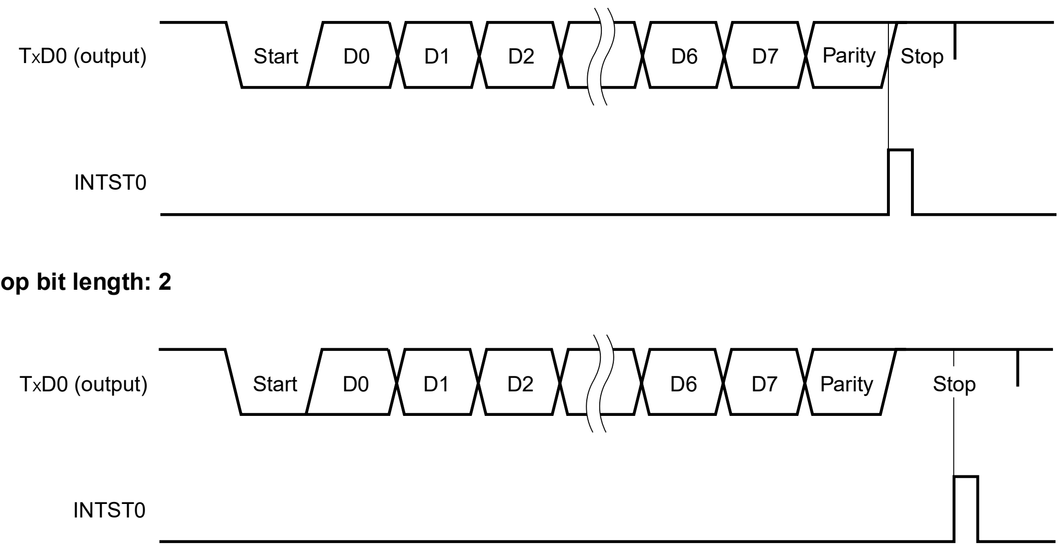
Transmission is stopped until the data to be transmitted next is written to TXS0.

Figure 13-8 shows the timing of the transmission completion interrupt request (INTST0). This interrupt occurs as soon as the last stop bit has been output.

Caution After transmit data is written to TXS0, do not write the next transmit data before the transmission completion interrupt signal (INTST0) is generated.

Figure 13-8. Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2

(d) Reception

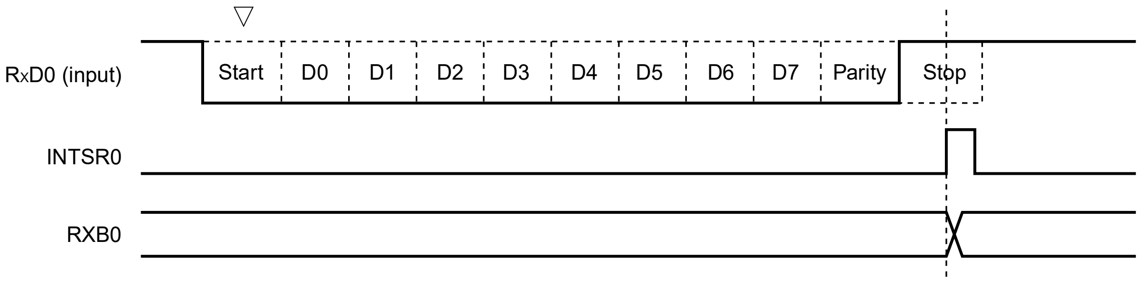
Reception is enabled and the RxD0 pin input is sampled when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is set to 1 and then bit 5 (RXE0) of ASIM0 is set to 1.

The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD0 pin input is detected. When the set value of baud rate generator control register 0 (BRGC0) has been counted, the RxD0 pin input is sampled again (▽ in Figure 13-9). If the RxD0 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 0 (RXS0) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR0) is generated and the data of RXS0 is written to receive buffer register 0 (RXB0). If an overrun error (OVE0) occurs, however, the receive data is not written to RXB0.

Even if a parity error (PE0) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR0) is generated after completion of reception.

Figure 13-9. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 0 (RXB0) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 0 (ASIS0) before reading RXB0.

(e) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 0 (ASIS0) is set as a result of data reception, a reception error interrupt request (INTSR0) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS0 in the reception error interrupt servicing (INTSR0) (refer to **Figure 13-3**).

The contents of ASIS0 are reset to 0 when ASIS0 is read.

Table 13-3. Cause of Reception Error

Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 0 (RXB0).

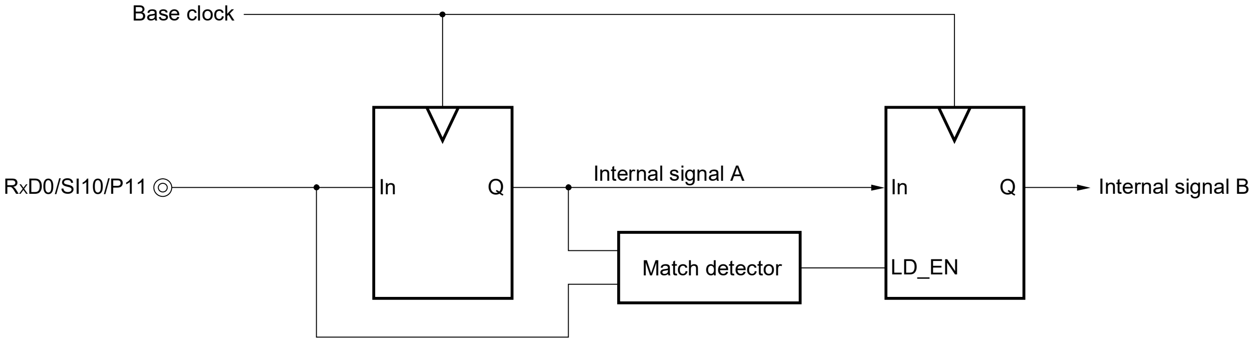
(f) Noise filter of receive data

The RxD0 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 13-10, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 13-10. Noise Filter Circuit



13.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and a 5-bit programmable counter, and generates a serial clock for transmission/reception of UART0.

Separate 5-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock

  The clock selected by bits 7 and 6 (TPS01 and TPS00) of baud rate generator control register 0 (BRGC0) is supplied to each module when bit 7 (POWER0) of asynchronous serial interface operation mode register 0 (ASIM0) is 1. This clock is called the base clock and its frequency is called f_{XCLK0}. The base clock is fixed to low level when POWER0 = 0.

- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 6 (TXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.

  It starts counting when POWER0 = 1 and TXE0 = 1.

  The counter is cleared to 0 when the first data transmitted is written to transmit shift register 0 (TXS0).

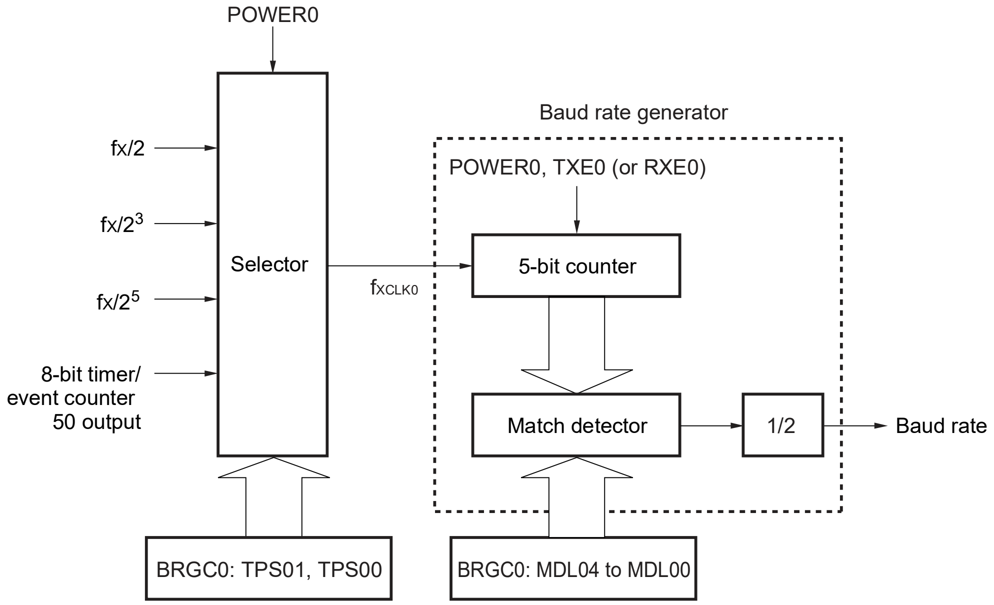
- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER0) or bit 5 (RXE0) of asynchronous serial interface operation mode register 0 (ASIM0) is 0.

  It starts counting when the start bit has been detected.

  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 13-11. Configuration of Baud Rate Generator



Remark		
Remark	POWER0:	Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
	TXE0:	Bit 6 of ASIM0
	RXE0:	Bit 5 of ASIM0
	BRGC0:	Baud rate generator control register 0

(2) Generation of serial clock

A serial clock can be generated by using baud rate generator control register 0 (BRGC0).

Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS01 and TPS00) of BRGC0.

Bits 4 to 0 (MDL04 to MDL00) of BRGC0 can be used to select the division value of the 5-bit counter.

(a) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\dfrac{f_{XCLK0}}{2 \times k}$ [bps]

Remark f_{XCLK0}: Frequency of base clock selected by the TPS01 and TPS00 bits of the BRGC0 register
k: Value set by the MDL04 to MDL00 bits of the BRGC0 register (k = 8, 9, 10, ..., 31)

(b) Error of baud rate

The baud rate error can be calculated by the following expression.

- Error (%) = $\left(\dfrac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1\right) \times 100$ [%]

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz
Set value of MDL04 to MDL00 bits of BRGC0 register = 10000B (k = 16)
Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)
= 2,500,000/(2 × 16) = 78,125 [bps]

Error = (78,125/76,800 – 1) × 100
= 1.725 [%]

(3) Example of setting baud rate

Table 13-4. Set Data of Baud Rate Generator

Baud Rate [bps]	f_X = 10.0 MHz				f_X = 8.38 MHz				f_X = 4.19 MHz			
	TPS01, TPS00	k	Calculated Value	ERR[%]	TPS01, TPS00	k	Calculated Value	ERR[%]	TPS01, TPS00	k	Calculated Value	ERR[%]
2400	–	–	–	–	–	–	–	–	3	27	2425	1.03
4800	–	–	–	–	3	27	4850	1.03	3	14	4676	–2.58
9600	3	16	9766	1.73	3	14	9353	–2.58	2	27	9699	1.03
10400	3	15	10417	0.16	3	13	10072	–3.15	2	25	10475	0.72
19200	3	8	19531	1.73	2	27	19398	1.03	2	14	18705	–2.58
31250	2	20	31250	0	2	17	30809	–1.41	–	–	–	–
38400	2	16	39063	1.73	2	14	38796	–2.58	2	27	38796	1.03
76800	2	8	78125	1.73	1	27	77593	1.03	1	14	74821	–2.58
115200	1	22	113636	–1.36	1	18	116389	1.03	1	9	116389	1.03
153600	1	16	156250	1.73	1	14	149643	–2.58	–	–	–	–
230400	1	11	227273	–1.36	1	9	232778	1.03	–	–	–	–

Remark TPS01, TPS00: Bits 7 and 6 of baud rate generator control register 0 (BRGC0) (setting of base clock (f_{XCLK0}))

k: Value set by the MDL04 to MDL00 bits of BRGC0 (k = 8, 9, 10, ..., 31)

f_X: High-speed system clock oscillation frequency

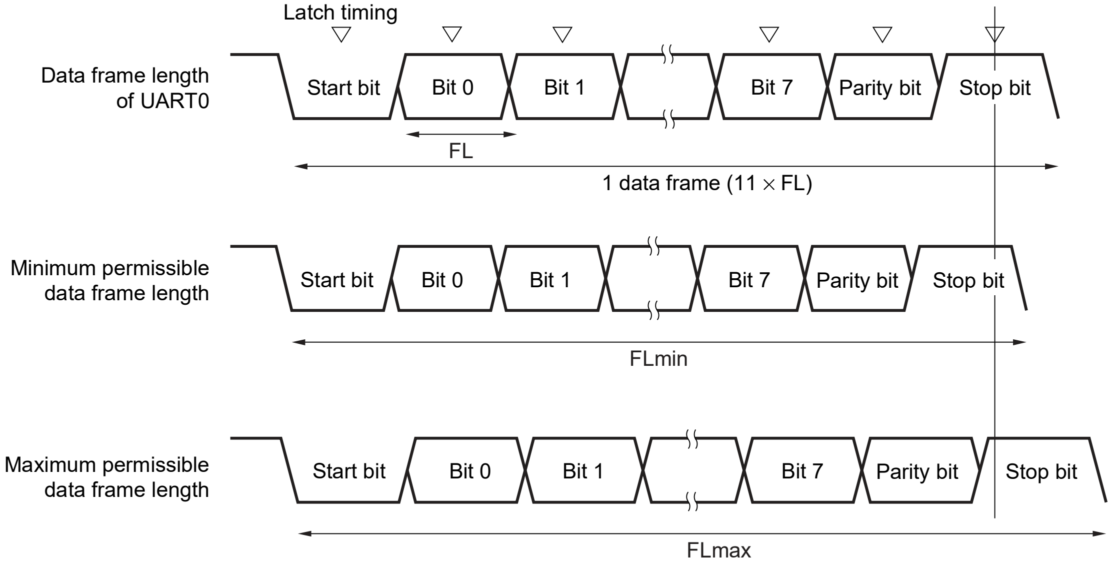
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 13-12. Permissible Baud Rate Range During Reception



As shown in Figure 13-12, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART0

k: Set value of BRGC0

FL: 1-bit data length

Margin of latch timing: 2 clocks

Minimum permissible data frame length: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k}\text{FL}$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\text{ Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\text{ FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\text{ FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\text{ Brate}$$

The permissible baud rate error between UART0 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 13-5. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
16	+4.14%	–4.19%
24	+4.34%	–4.38%
31	+4.44%	–4.47%

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 higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC0

CHAPTER 14 SERIAL INTERFACE UART6

14.1 Functions of Serial Interface UART6

Serial interface UART6 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, refer to **14.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

This mode supports the LIN (Local Interconnect Network)-bus. The functions of this mode are outlined below.

For details, refer to **14.4.2 Asynchronous serial interface (UART) mode** and **14.4.3 Dedicated baud rate generator**.

- Two-pin configuration TxD6: Transmit data output pin
  RxD6: Receive data input pin
- Data length of communication data can be selected from 7 or 8 bits.
- Dedicated internal 8-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Twelve operating clock inputs selectable
- MSB- or LSB-first communication selectable
- Inverted transmission operation
- Synchronous break field transmission from 13 to 20 bits
- More than 11 bits can be identified for synchronous break field reception (SBF reception flag provided).

Cautions 1. The TxD6 output inversion function inverts only the transmission side and not the reception side. To use this function, the reception side must be ready for reception of inverted data.

2. If clock supply to serial interface UART6 is not stopped (e.g., in the HALT mode), normal operation continues. If clock supply to serial interface UART6 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD6 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER6 = 0, RXE6 = 0, and TXE6 = 0.

3. If data is continuously transmitted, the communication timing from the stop bit to the next start bit is extended two operating clocks of the macro. However, this does not affect the result of communication because the reception side initializes the timing when it has detected a start bit. Do not use the continuous transmission function if the interface is incorporated in LIN.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

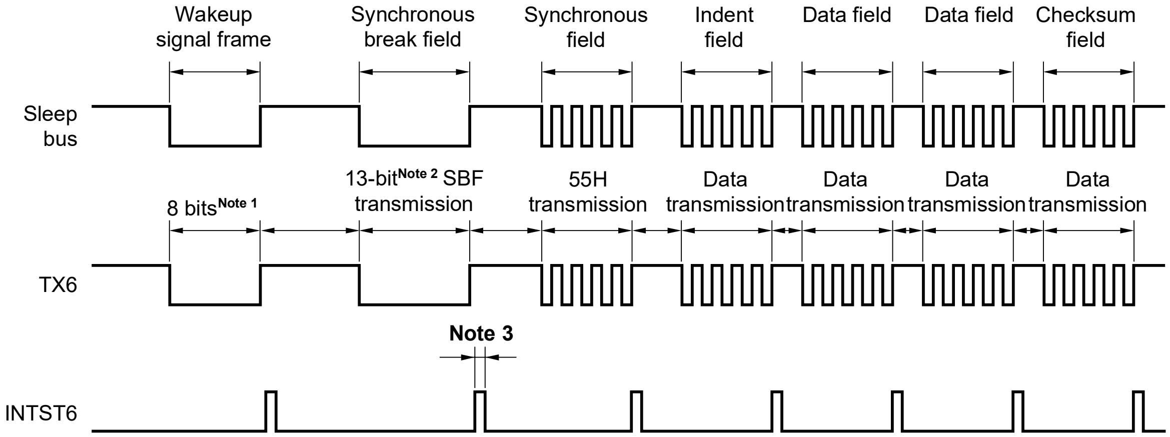
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±15% or less.

Figures 14-1 and 14-2 outline the transmission and reception operations of LIN.

Figure 14-1. LIN Transmission Operation



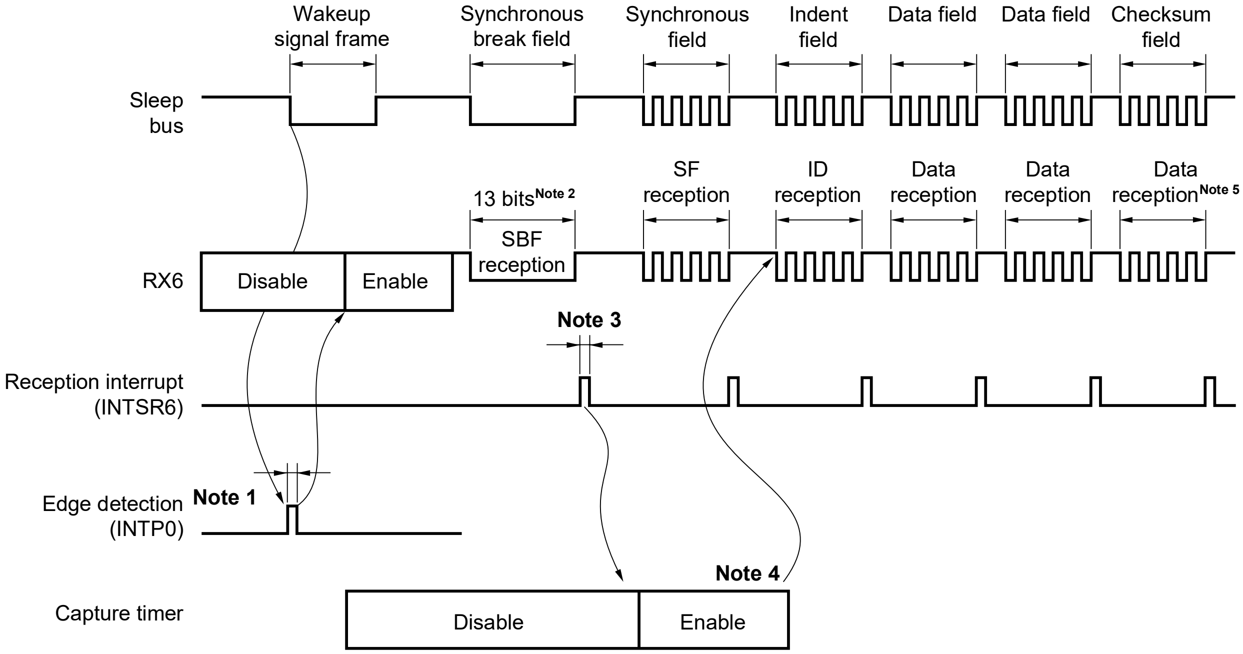
Notes 1. The wakeup signal frame is substituted by 80H transmission in the 8-bit mode.

2. The synchronous break field is output by hardware. The output width is the bit length set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6). If more precise output width adjustment is necessary, use baud rate generator control register 6 (BRGC6) (refer to **14.4.2 (2) (h) SBF transmission**).

3. INTST6 is output on completion of each transmission. It is also output when SBF is transmitted.

Remark The interval between each field is controlled by software.

Figure 14-2. LIN Reception Operation



Notes 1. The wakeup signal is detected at the edge of the pin, and enables UART6 and sets the SBF reception mode.

2. Reception continues until the STOP bit is detected. When an SBF with low-level data of 11 bits or more has been detected, it is assumed that SBF reception has been completed correctly, and an interrupt signal is output. If an SBF with low-level data of less than 11 bits has been detected, it is assumed that an SBF reception error has occurred. The interrupt signal is not output and the SBF reception mode is restored.

3. If SBF reception has been completed correctly, an interrupt signal is output. This SBF reception completion interrupt enables the capture timer. Detection of errors OVE6, PE6, and FE6 is suppressed, and error detection processing of UART communication and data transfer of the shift register and RXB6 is not performed. The shift register holds the reset value FFH.

4. Calculate the baud rate error from the bit length of the synchronous field, disable UART6 after SF reception, and then re-set baud rate generator control register 6 (BRGC6).

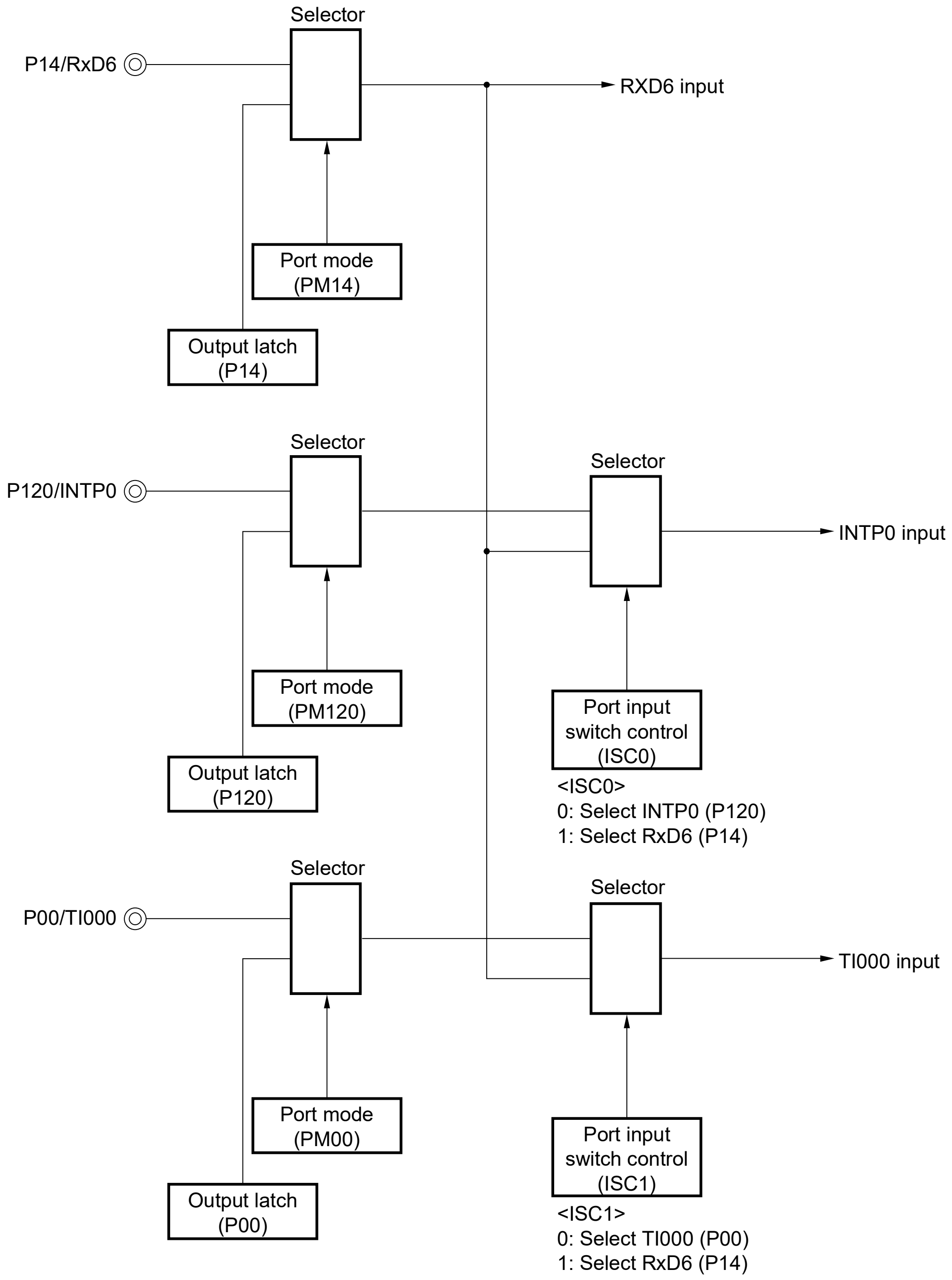
5. Distinguish the checksum field by software. Also perform processing by software to initialize UART6 after reception of the checksum field and to set the SBF reception mode again.

To perform a LIN receive operation, use a configuration like the one shown in Figure 14-3.

The wakeup signal transmitted from the LIN master is received by detecting the edge of the external interrupt (INTP0). The length of the synchronous field transmitted from the LIN master can be measured using the external event capture operation of 16-bit timer/event counter 00, and the baud rate error can be calculated.

The input source of the reception port input (RxD6) can be input to the external interrupt (INTP0) and 16-bit timer/event counter 00 by port input switch control (ISC0/ISC1), without connecting RxD6 and INTP0/TI000 externally.

Figure 14-3. Port Configuration for LIN Reception Operation



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (refer to **Figure 14-11**)

The peripheral functions used in the LIN communication operation are shown below.

<Peripheral functions used>

- External interrupt (INTP0); wakeup signal detection
  Use: Detects the wakeup signal edges and detects start of communication.
- 16-bit timer/event counter 00 (TI000); baud rate error detection
  Use: Detects the baud rate error (measures the TI000 input edge interval in the capture mode) by detecting the sync field (SF) length and divides it by the number of bits.
- Serial interface UART6

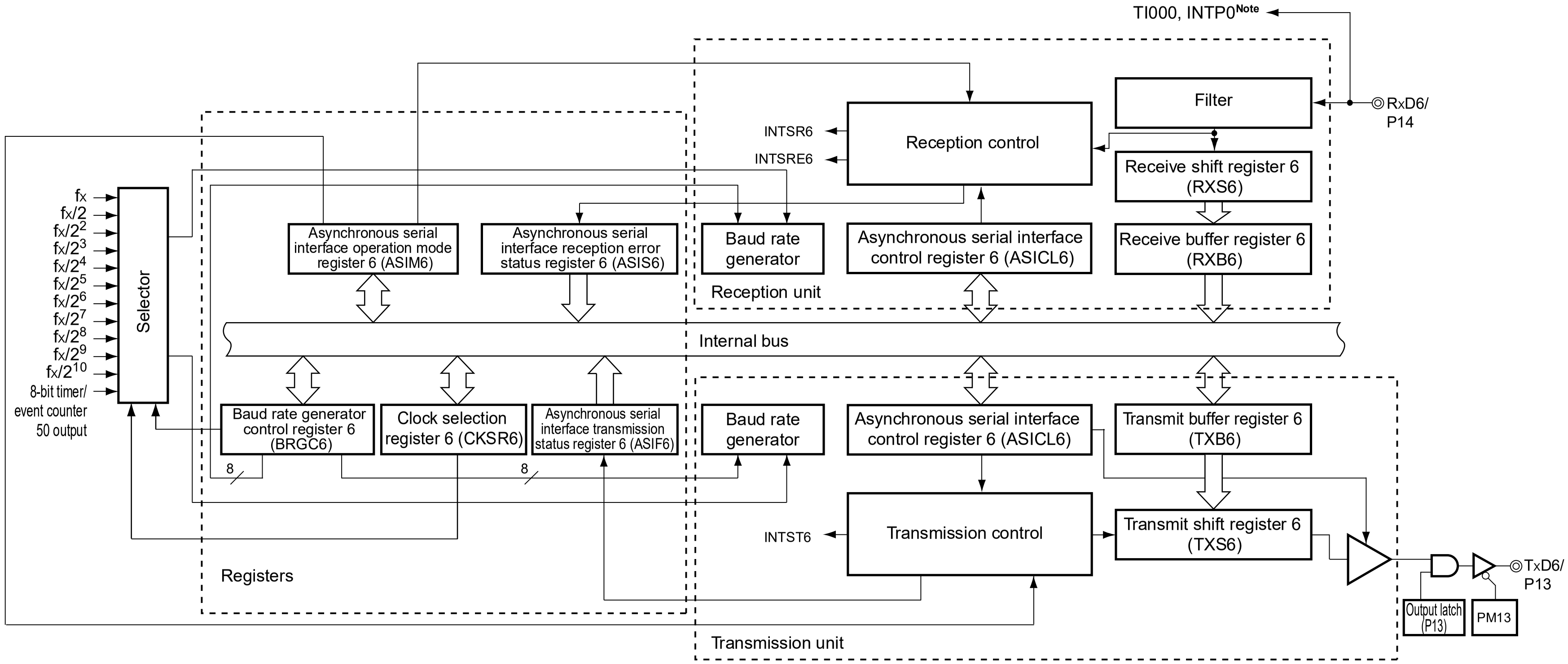
14.2 Configuration of Serial Interface UART6

Serial interface UART6 includes the following hardware.

Table 14-1. Configuration of Serial Interface UART6

Item	Configuration
Registers	Receive buffer register 6 (RXB6) Receive shift register 6 (RXS6) Transmit buffer register 6 (TXB6) Transmit shift register 6 (TXS6)
Control registers	Asynchronous serial interface operation mode register 6 (ASIM6) Asynchronous serial interface reception error status register 6 (ASIS6) Asynchronous serial interface transmission status register 6 (ASIF6) Clock selection register 6 (CKSR6) Baud rate generator control register 6 (BRGC6) Asynchronous serial interface control register 6 (ASICL6) Input switch control register (ISC) Port mode register 1 (PM1) Port register 1 (P1)

Figure 14-4. Block Diagram of Serial Interface UART6



Note Selectable with input switch control register (ISC).

(1) Receive buffer register 6 (RXB6)

This 8-bit register stores parallel data converted by receive shift register 6 (RXS6).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 6 (RXS6). If the data length is set to 7 bits, data is transferred as follows.

- In LSB-first reception, the receive data is transferred to bits 0 to 6 of RXB6 and the MSB of RXB6 is always 0.
- In MSB-first reception, the receive data is transferred to bits 1 to 7 of RXB6 and the LSB of RXB6 is always 0.

If an overrun error (OVE6) occurs, the receive data is not transferred to RXB6.

RXB6 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

$\overline{\text{RESET}}$ input sets RXB6 to FFH.

(2) Receive shift register 6 (RXS6)

This register converts the serial data input to the RxD6 pin into parallel data.

RXS6 cannot be directly manipulated by a program.

(3) Transmit buffer register 6 (TXB6)

This buffer register is used to set transmit data. Transmission is started when data is written to TXB6.

This register can be read or written by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TXB6 to FFH.

Cautions 1. Do not write data to TXB6 when bit 1 (TXBF6) of asynchronous serial interface transmission status register 6 (ASIF6) is 1.

2. Do not refresh (write the same value to) TXB6 by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) are 1 or when bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 are 1).

(4) Transmit shift register 6 (TXS6)

This register transmits the data transferred from TXB6 from the TxD6 pin as serial data. Data is transferred from TXB6 immediately after TXB6 is written for the first transmission, or immediately before INTST6 occurs after one frame was transmitted for continuous transmission. Data is transferred from TXB6 and transmitted from the TxD6 pin at the falling edge of the base clock.

TXS6 cannot be directly manipulated by a program.

14.3 Registers Controlling Serial Interface UART6

Serial interface UART6 is controlled by the following nine registers.

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 6 (ASIM6)

This 8-bit register controls the serial communication operations of serial interface UART6.

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

$\overline{\text{RESET}}$ input sets ASIM6 to 01H.

Remark ASIM6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (1/2)

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].
1[Note 3]	Enables operation of the internal operation clock

TXE6	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission

Notes 1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to high level when POWER6 = 0.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

3. Operation of the 8-bit counter output is enabled at the second base clock after 1 is written to the POWER6 bit.

Figure 14-5. Format of Asynchronous Serial Interface Operation Mode Register 6 (ASIM6) (2/2)

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception

PS61	PS60	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL6	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL6	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

ISRM6	Enables/disables occurrence of reception completion interrupt in case of error
0	“INTSRE6” occurs in case of error (at this time, INTSR6 does not occur).
1	“INTSR6” occurs in case of error (at this time, INTSRE6 does not occur).

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE6) of asynchronous serial interface reception error status register 6 (ASIS6) is not set and the error interrupt does not occur.

Cautions 1. At startup, set POWER6 to 1 and then set TXE6 to 1. To stop the operation, clear TXE6 to 0, and then clear POWER6 to 0.

2. At startup, set POWER6 to 1 and then set RXE6 to 1. To stop the operation, clear RXE6 to 0, and then clear POWER6 to 0.

3. Set POWER6 to 1 and then set RXE6 to 1 while a high level is input to the RxD6 pin. If POWER6 is set to 1 and RXE6 is set to 1 while a low level is input, reception is started.

4. Clear the TXE6 and RXE6 bits to 0 before rewriting the PS61, PS60, and CL6 bits.

5. Fix the PS61 and PS60 bits to 0 when mounting the device on LIN.

6. Make sure that TXE6 = 0 when rewriting the SL6 bit. Reception is always performed with “the number of stop bits = 1”, and therefore, is not affected by the set value of the SL6 bit.

7. Make sure that RXE6 = 0 when rewriting the ISRM6 bit.

(2) Asynchronous serial interface reception error status register 6 (ASIS6)

This register indicates an error status on completion of reception by serial interface UART6. It includes three error flag bits (PE6, FE6, OVE6).

This register is read-only by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIS6 to 00H if bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 0. 00H is read when this register is read.

Figure 14-6. Format of Asynchronous Serial Interface Reception Error Status Register 6 (ASIS6)

Address: FF53H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	Status flag indicating parity error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the parity of transmit data does not match the parity bit on completion of reception

FE6	Status flag indicating framing error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If the stop bit is not detected on completion of reception

OVE6	Status flag indicating overrun error
0	If POWER6 = 0 and RXE6 = 0, or if ASIS6 register is read
1	If receive data is set to the RXB6 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE6 bit differs depending on the set values of the PS61 and PS60 bits of asynchronous serial interface operation mode register 6 (ASIM6).

2. The first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 6 (RXB6) but discarded.

4. If data is read from ASIS6, a wait cycle is generated. Do not read data from ASIS6 when the CPU is operating on the subsystem clock and the high-speed system clock is stopped. For details, refer to CHAPTER 29 CAUTIONS FOR WAIT.

(3) Asynchronous serial interface transmission status register 6 (ASIF6)

This register indicates the status of transmission by serial interface UART6. It includes two status flag bits (TXBF6 and TXSF6).

Transmission can be continued without disruption even during an interrupt period, by writing the next data to the TXB6 register after data has been transferred from the TXB6 register to the TXS6 register.

This register is read-only by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears ASIF6 to 00H if bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 0.

Figure 14-7. Format of Asynchronous Serial Interface Transmission Status Register 6 (ASIF6)

Address: FF55H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	Transmit buffer data flag
0	If POWER6 = 0 or TXE6 = 0, or if data is transferred to transmit shift register 6 (TXS6)
1	If data is written to transmit buffer register 6 (TXB6) (if data exists in TXB6)

TXSF6	Transmit shift register data flag
0	If POWER6 = 0 or TXE6 = 0, or if the next data is not transferred from transmit buffer register 6 (TXB6) after completion of transfer
1	If data is transferred from transmit buffer register 6 (TXB6) (if data transmission is in progress)

Cautions 1. To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is “0”. If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is “1”, the transmit data cannot be guaranteed.

2. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is “0” after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is “1”, the transmit data cannot be guaranteed.

(4) Clock selection register 6 (CKSR6)

This register selects the base clock of serial interface UART6.

CKSR6 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CKSR6 to 00H.

Remark CKSR6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 14-8. Format of Clock Selection Register 6 (CKSR6)

Address: FF56H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60	Base clock (f_{XCLK6}) selection[Note 1]
0	0	0	0	f_X (10 MHz)
0	0	0	1	$f_X/2$ (5 MHz)
0	0	1	0	$f_X/2^2$ (2.5 MHz)
0	0	1	1	$f_X/2^3$ (1.25 MHz)
0	1	0	0	$f_X/2^4$ (625 kHz)
0	1	0	1	$f_X/2^5$ (312.5 kHz)
0	1	1	0	$f_X/2^6$ (156.25 kHz)
0	1	1	1	$f_X/2^7$ (78.13 kHz)
1	0	0	0	$f_X/2^8$ (39.06 kHz)
1	0	0	1	$f_X/2^9$ (19.53 kHz)
1	0	1	0	$f_X/2^{10}$ (9.77 kHz)
1	0	1	1	TM50 output[Note 2]
Other than above				Setting prohibited

Notes 1. Be sure to set the base clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Base clock ≤ 10 MHz
- V_{DD} = 3.3 to 4.0 V: Base clock ≤ 8.38 MHz
- V_{DD} = 2.7 to 3.3 V: Base clock ≤ 5 MHz

2. Note the following points when selecting the TM50 output as the base clock.

- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the base clock to make the duty = 50%.
- Mode in which the base clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).

It is not necessary to enable the TO50 pin as a timer output pin in any mode.

Cautions 1. When the Ring-OSC clock is selected as the clock to be supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the count clock. If the base clock is the Ring-OSC clock, the operation of serial interface UART6 is not guaranteed.

2. Make sure POWER6 = 0 when rewriting TPS63 to TPS60.

Remarks 1. Figures in parentheses apply to operation with f_X = 10 MHz.

2. f_X: High-speed system clock oscillation frequency

3. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(5) Baud rate generator control register 6 (BRGC6)

This register sets the division value of the 8-bit counter of serial interface UART6.

BRGC6 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets BRGC6 to FFH.

Remark BRGC6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1).

Figure 14-9. Format of Baud Rate Generator Control Register 6 (BRGC6)

Address: FF57H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	Output clock selection of 8-bit counter
0	0	0	0	0	×	×	×	×	Setting prohibited
0	0	0	0	1	0	0	0	8	$f_{XCLK6}/8$
0	0	0	0	1	0	0	1	9	$f_{XCLK6}/9$
0	0	0	0	1	0	1	0	10	$f_{XCLK6}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	1	1	1	1	0	0	252	$f_{XCLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{XCLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{XCLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{XCLK6}/255$

Cautions 1. Make sure that bit 6 (TXE6) and bit 5 (RXE6) of the ASIM6 register = 0 when rewriting the MDL67 to MDL60 bits.

2. The baud rate is the output clock of the 8-bit counter divided by 2.

Remarks 1. f_{XCLK6}: Frequency of base clock selected by the TPS63 to TPS60 bits of CKSR6 register

2. k: Value set by MDL67 to MDL60 bits (k = 8, 9, 10, ..., 255)

3. ×: don’t care

(6) Asynchronous serial interface control register 6 (ASICL6)

This register controls the serial communication operations of serial interface UART6.

ASICL6 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets ASICL6 to 16H.

Caution ASICL6 can be refreshed (the same value is written) by software during a communication operation (when bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1 or bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1). Note, however, that communication is started by the refresh operation because bit 6 (SBRT6) of ASICL6 is cleared to 0 when communication is completed (when an interrupt signal is generated).

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (1/2)

Address: FF58H After reset: 16H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6

SBRF6	SBF reception status flag
0	If POWER6 = 0 and RXE6 = 0 or if SBF reception has been completed correctly
1	SBF reception in progress

SBRT6	SBF reception trigger
0	–
1	SBF reception trigger

SBTT6	SBF transmission trigger
0	–
1	SBF transmission trigger

Note Bit 7 is read-only.

Figure 14-10. Format of Asynchronous Serial Interface Control Register 6 (ASICL6) (2/2)

SBL62	SBL61	SBL60	SBF transmission output width control
1	0	1	SBF is output with 13-bit length.
1	1	0	SBF is output with 14-bit length.
1	1	1	SBF is output with 15-bit length.
0	0	0	SBF is output with 16-bit length.
0	0	1	SBF is output with 17-bit length.
0	1	0	SBF is output with 18-bit length.
0	1	1	SBF is output with 19-bit length.
1	0	0	SBF is output with 20-bit length.

DIR6	First-bit specification
0	MSB
1	LSB

TXDLV6	Enables/disables inverting TxD6 output
0	Normal output of TxD6
1	Inverted output of TxD6

Cautions 1. In the case of an SBF reception error, return the mode to the SBF reception mode and hold the status of the SBRF6 flag.

2. Before setting the SBRT6 bit, make sure that bit 7 (POWER6) and bit 5 (RXE6) of ASIM6 = 1.

3. The read value of the SBRT6 bit is always 0. SBRT6 is automatically cleared to 0 after SBF reception has been correctly completed.

4. Before setting the SBTT6 bit to 1, make sure that bit 7 (POWER6) and bit 6 (TXE6) of ASIM6 = 1.

5. The read value of the SBTT6 bit is always 0. SBTT6 is automatically cleared to 0 at the end of SBF transmission.

6. Before rewriting the DIR6 and TXDLV6 bits, clear the TXE6 and RXE6 bits to 0.

7. When using the 78K0/KD1+ to evaluate the program of a mask ROM version of the 78K0/KD1, set the SBTT6, SBL62, SBL61, and SBL60 bits to 0, 1, 0, 1, respectively.

(7) Input switch control register (ISC)

The input switch control register (ISC) is used to receive a status signal transmitted from the master during LIN (Local Interconnect Network) reception. The input source is switched by setting ISC.

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

$\overline{\text{RESET}}$ input clears ISC to 00H.

Figure 14-11. Format of Input Switch Control Register (ISC)

Address: FF4FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	TI000 input source selection
0	TI000 (P00)
1	RxD6 (P14)

ISC0	INTP0 input source selection
0	INTP0 (P120)
1	RxD6 (P14)

(8) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P13/TxD6 pin for serial interface data output, clear PM13 to 0 and set the output latch of P13 to 1.

When using the P14/RxD6 pin for serial interface data input, set PM14 to 1. The output latch of P14 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM1 to FFH.

Figure 14-12. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 Operation of Serial Interface UART6

Serial interface UART6 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

14.4.1 Operation stop mode

In this mode, serial communication cannot be executed; therefore, the power consumption can be reduced. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER6, TXE6, and RXE6) of ASIM6 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 6 (ASIM6).
ASIM6 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets ASIM6 to 01H.

Address: FF50H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6

POWER6	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE6	Enables/disables transmission
0	Disables transmission operation (synchronously resets the transmission circuit).

RXE6	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The output of the TxD6 pin goes high and the input from the RxD6 pin is fixed to high level when POWER6 = 0.

2. Asynchronous serial interface reception error status register 6 (ASIS6), asynchronous serial interface transmission status register 6 (ASIF6), bit 7 (SBRF6) and bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6), and receive buffer register 6 (RXB6) are reset.

Caution Clear POWER6 to 0 after clearing TXE6 and RXE6 to 0 to set the operation stop mode.
To start the operation, set POWER6 to 1, and then set TXE6 and RXE6 to 1.

Remark To use the RxD6/P14 and TxD6/P13 pins as general-purpose port pins, refer to **CHAPTER 4 PORT FUNCTIONS**.

14.4.2 Asynchronous serial interface (UART) mode

In this mode, data of 1 byte is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 6 (ASIM6)
- Asynchronous serial interface reception error status register 6 (ASIS6)
- Asynchronous serial interface transmission status register 6 (ASIF6)
- Clock selection register 6 (CKSR6)
- Baud rate generator control register 6 (BRGC6)
- Asynchronous serial interface control register 6 (ASICL6)
- Input switch control register (ISC)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the CKSR6 register (refer to **Figure 14-8**).
<2> Set the BRGC6 register (refer to **Figure 14-9**).
<3> Set bits 0 to 4 (ISRM6, SL6, CL6, PS60, PS61) of the ASIM6 register (refer to **Figure 14-5**).
<4> Set bits 0 and 1 (TXDLV6, DIR6) of the ASICL6 register (refer to **Figure 14-10**).
<5> Set bit 7 (POWER6) of the ASIM6 register to 1.
<6> Set bit 6 (TXE6) of the ASIM6 register to 1. → Transmission is enabled.
Set bit 5 (RXE6) of the ASIM6 register to 1. → Reception is enabled.
<7> Write data to transmit buffer register 6 (TXB6). → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

POWER6	TXE6	RXE6	PM13	P13	PM14	P14	UART6 Operation	Pin Function	
								TxD6/P13	RxD6/P14
0	0	0	×[Note]	×[Note]	×[Note]	×[Note]	Stop	P13	P14
1	0	1	×[Note]	×[Note]	1	×	Reception	P13	RxD6
	1	0	0	1	×[Note]	×[Note]	Transmission	TxD6	P14
	1	1	0	1	1	×	Transmission/ reception	TxD6	RxD6

Note Can be set as port function.

Remark ×: don't care

POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)

TXE6: Bit 6 of ASIM6

RXE6: Bit 5 of ASIM6

PM1×: Port mode register

P1×: Port output latch

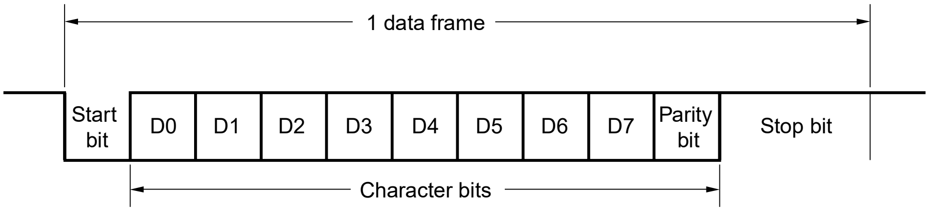
(2) Communication operation

(a) Format and waveform example of normal transmit/receive data

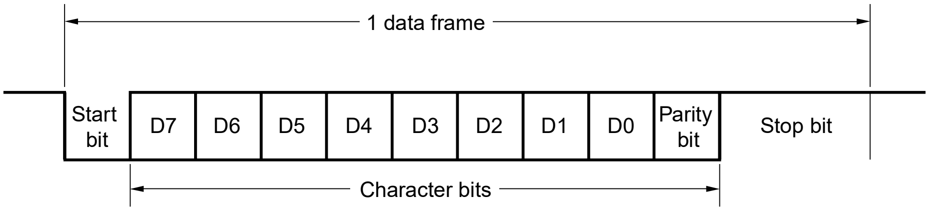
Figures 14-13 and 14-14 show the format and waveform example of the normal transmit/receive data.

Figure 14-13. Format of Normal UART Transmit/Receive Data

1. LSB-first transmission/reception



2. MSB-first transmission/reception



One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

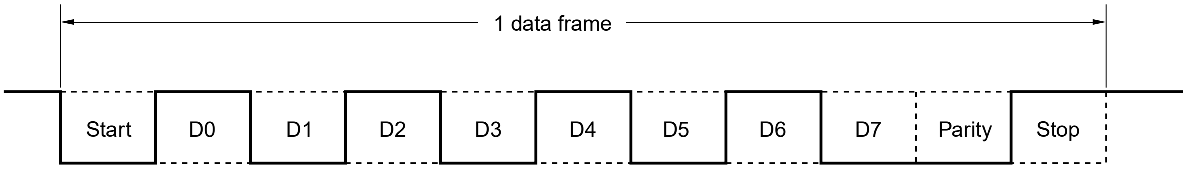
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 6 (ASIM6).

Whether data is communicated with the LSB or MSB first is specified by bit 1 (DIR6) of asynchronous serial interface control register 6 (ASICL6).

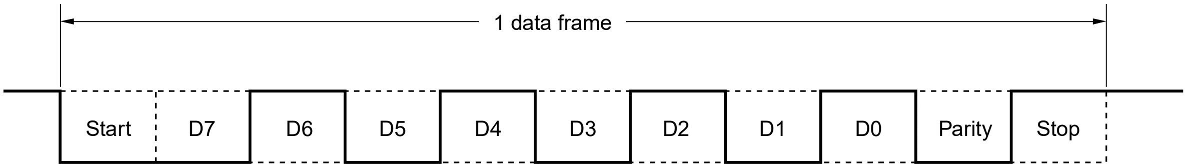
Whether the TxD6 pin outputs normal or inverted data is specified by bit 0 (TXDLV6) of ASICL6.

Figure 14-14. Example of Normal UART Transmit/Receive Data Waveform

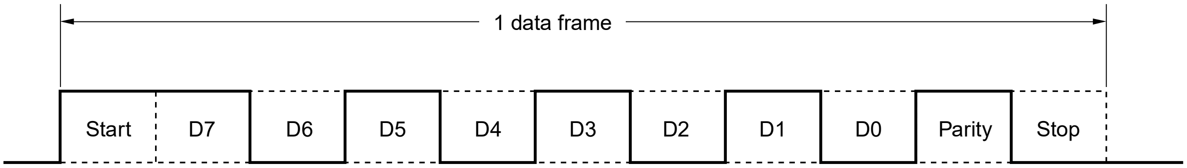
1. **Data length: 8 bits, LSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



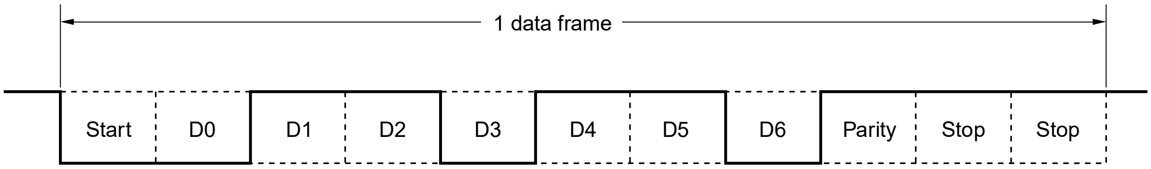
2. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



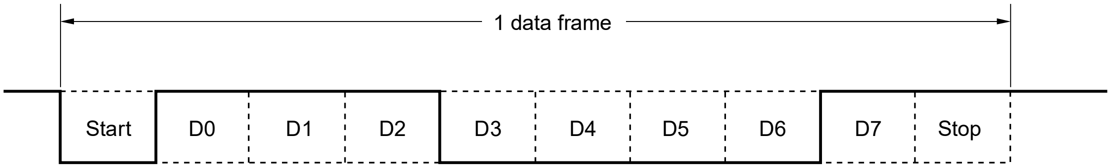
3. **Data length: 8 bits, MSB first, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H, TxD6 pin inverted output**



4. **Data length: 7 bits, LSB first, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H**



5. **Data length: 8 bits, LSB first, Parity: None, Stop bit: 1 bit, Communication data: 87H**



(b) 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 transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

Caution Fix the PS61 and PS60 bits to 0 when the device is incorporated in LIN.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 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”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Normal transmission

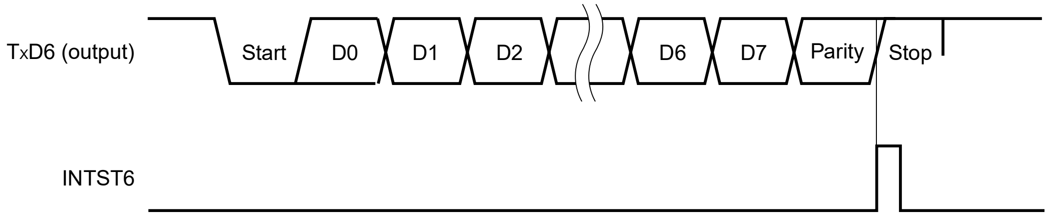
The TxD6 pin outputs a high level when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1. If bit 6 (TXE6) of ASIM6 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit buffer register 6 (TXB6). The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the data in TXB6 is transferred to transmit shift register 6 (TXS6). After that, the data is sequentially output from TXS6 to the TxD6 pin. When transmission is completed, the parity and stop bits set by ASIM6 are appended and a transmission completion interrupt request (INTST6) is generated. Transmission is stopped until the data to be transmitted next is written to TXB6.

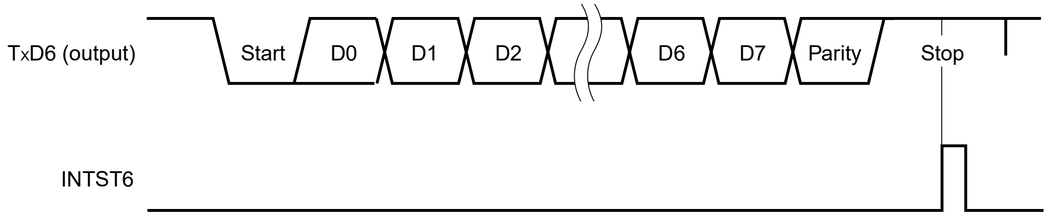
Figure 14-15 shows the timing of the transmission completion interrupt request (INTST6). This interrupt occurs as soon as the last stop bit has been output.

Figure 14-15. Normal Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIF6 register change from “10” to “11”, and to “01” during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is incorporated in a LIN, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is “0”. If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is “1”, the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

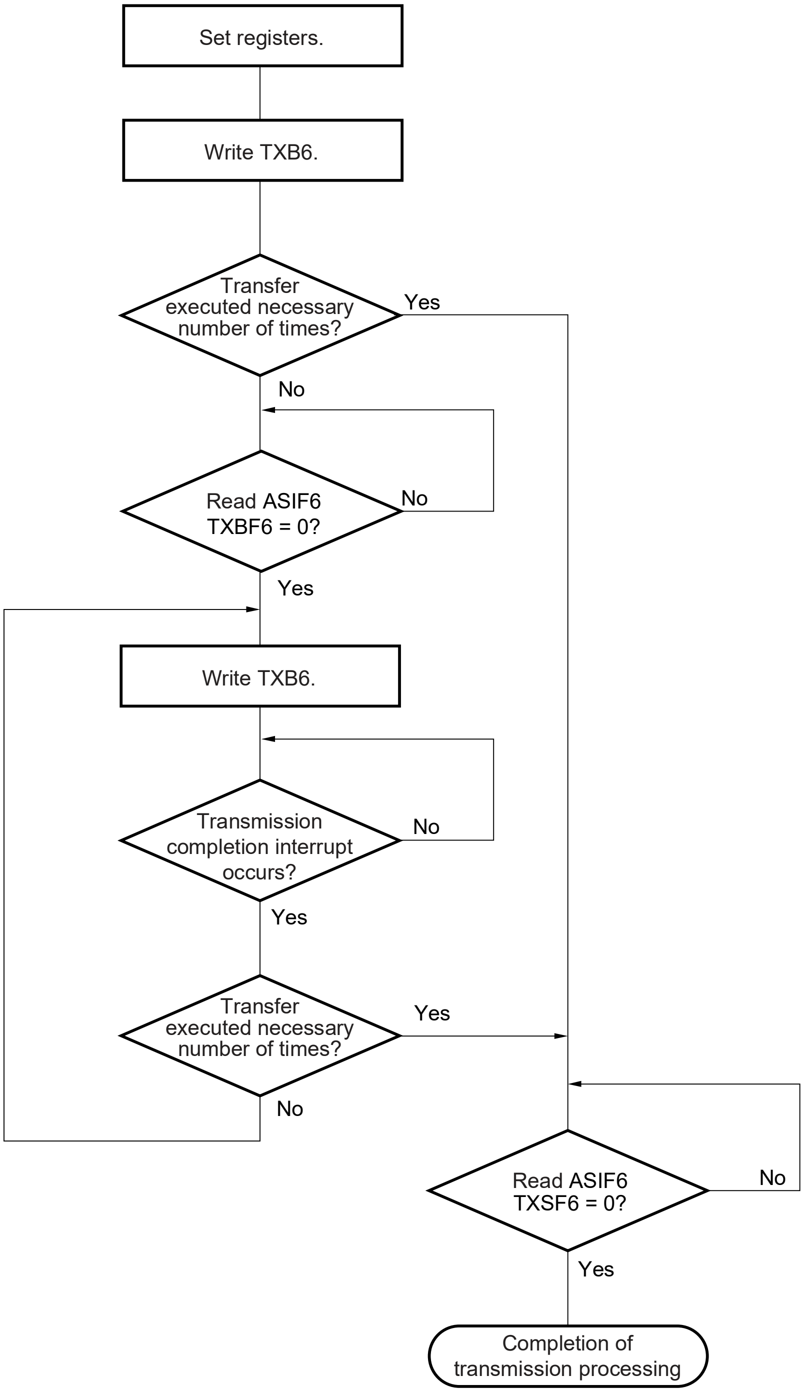
TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is “0” after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is “1”, the transmit data cannot be guaranteed.

2. During continuous transmission, an overrun error may occur, which means that the next transmission was completed before execution of INTST6 interrupt servicing after transmission of one data frame. An overrun error can be detected by developing a program that can count the number of transmit data and by referencing the TXSF6 flag.

Figure 14-16 shows an example of the continuous transmission processing flow.

Figure 14-16. Example of Continuous Transmission Processing Flow


Set registers.
Write TXB6.
Transfer executed necessary number of times?
Yes
No
Read ASIF6 TXBF6 = 0?
No
Yes
Write TXB6.
Transmission completion interrupt occurs?
No
Yes
Transfer executed necessary number of times?
Yes
No
Read ASIF6 TXSF6 = 0?
No
Yes
Completion of transmission processing

Remark TXB6: Transmit buffer register 6

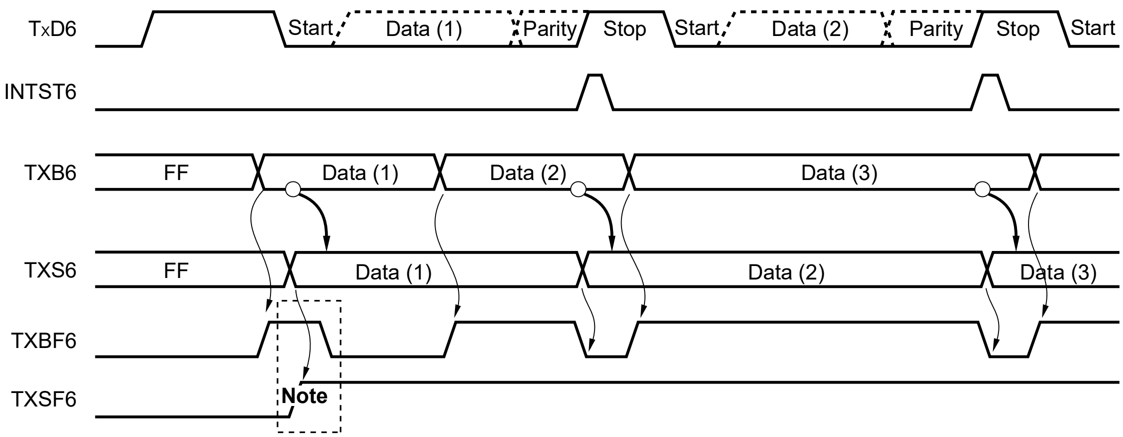
ASIF6: Asynchronous serial interface transmission status register 6

TXBF6: Bit 1 of ASIF6 (transmit buffer data flag)

TXSF6: Bit 0 of ASIF6 (transmit shift register data flag)

Figure 14-17 shows the timing of starting continuous transmission, and Figure 14-18 shows the timing of ending continuous transmission.

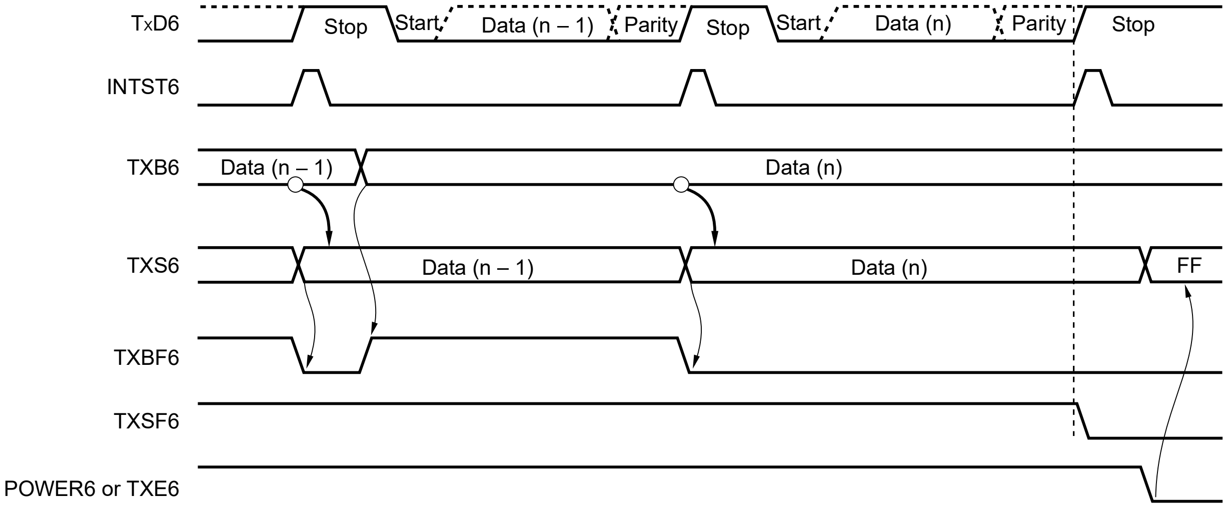
Figure 14-17. Timing of Starting Continuous Transmission



Note When ASIF6 is read, there is a period in which TXBF6 and TXSF6 = 1, 1. Therefore, judge whether writing is enabled using only the TXBF6 bit.

Remark		
	TxD6:	TxD6 pin (output)
	INTST6:	Interrupt request signal
	TXB6:	Transmit buffer register 6
	TXS6:	Transmit shift register 6
	ASIF6:	Asynchronous serial interface transmission status register 6
	TXBF6:	Bit 1 of ASIF6
	TXSF6:	Bit 0 of ASIF6

Figure 14-18. Timing of Ending Continuous Transmission



Remark TxD6: TxD6 pin (output)

INTST6: Interrupt request signal

TXB6: Transmit buffer register 6

TXS6: Transmit shift register 6

ASIF6: Asynchronous serial interface transmission status register 6

TXBF6: Bit 1 of ASIF6

TXSF6: Bit 0 of ASIF6

POWER6: Bit 7 of asynchronous serial interface operation mode register (ASIM6)

TXE6: Bit 6 of asynchronous serial interface operation mode register (ASIM6)

(e) Normal reception

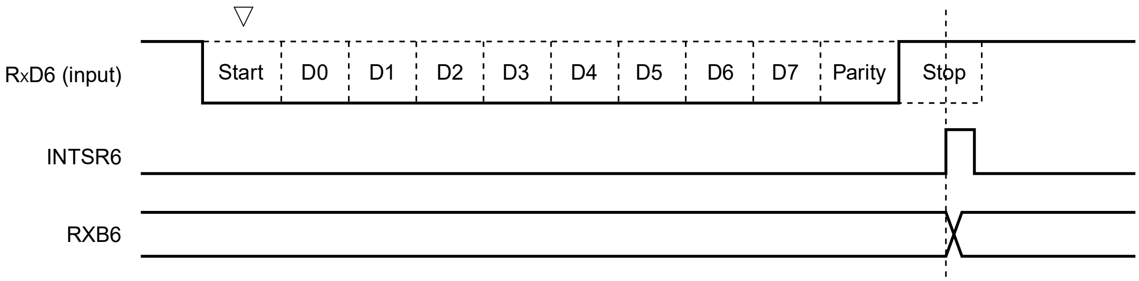
Reception is enabled and the RXD6 pin input is sampled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1.

The 8-bit counter of the baud rate generator starts counting when the falling edge of the RxD6 pin input is detected. When the set value of baud rate generator control register 6 (BRGC6) has been counted, the RxD6 pin input is sampled again (▽ in Figure 14-19). If the RxD6 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in the receive shift register (RXS6) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR6) is generated and the data of RXS6 is written to receive buffer register 6 (RXB6). If an overrun error (OVE6) occurs, however, the receive data is not written to RXB6.

Even if a parity error (PE6) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSR6/INTSRE6) is generated on completion of reception.

Figure 14-19. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 6 (RXB6) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 6 (ASIS6) before reading RXB6.

(f) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 6 (ASIS6) is set as a result of data reception, a reception error interrupt request (INTSR6/INTSRE6) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS6 in the reception error interrupt servicing (INTSR6/INTSRE6) (refer to **Figure 14-6**).

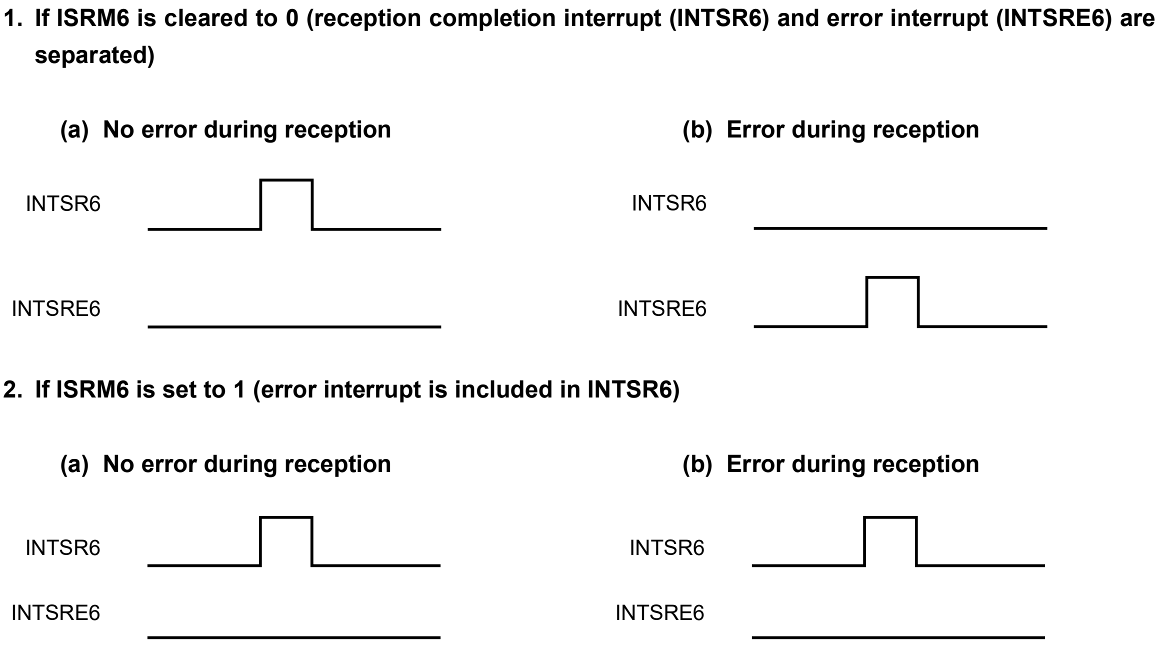
The contents of ASIS6 are reset to 0 when ASIS6 is read.

Table 14-3. Cause of Reception Error

Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 6 (RXB6).

The error interrupt can be separated into reception completion interrupt (INTSR6) and error interrupt (INTSRE6) by clearing bit 0 (ISRM6) of asynchronous serial interface operation mode register 6 (ASIM6) to 0.

Figure 14-20. Reception Error Interrupt



1. If ISRM6 is cleared to 0 (reception completion interrupt (INTSR6) and error interrupt (INTSRE6) are separated)

(a) No error during reception

(b) Error during reception

2. If ISRM6 is set to 1 (error interrupt is included in INTSR6)

(a) No error during reception

(b) Error during reception

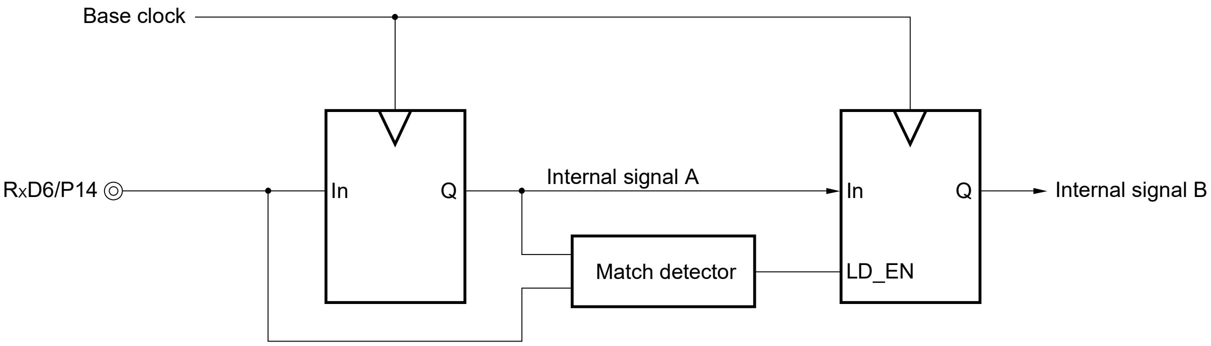
(g) Noise filter of receive data

The RxD6 signal is sampled with the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 14-21, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 14-21. Noise Filter Circuit



(h) SBF transmission

When the device is incorporated in LIN, the SBF (Synchronous Break Field) transmission control function is used for transmission. For the transmission operation of LIN, refer to **Figure 14-1 LIN Transmission Operation**.

An SBF length of a low level of 13 bits or more is set by bits 4 to 2 (SBL62 to SBL60) of asynchronous serial interface control register 6 (ASICL6). If more precise output width adjustment is necessary, use the baud rate value of the normal UART transmission function.

[Setting method]

Transmit 00H by setting the number of character bits of the data to 8 bits and the parity bit to 0 parity or even parity. This enables a low-level transmission of a data frame consisting of 10 bits (1 bit (start bit) + 8 bits (character bits) + 1 bit (parity bit)).

Adjust the baud rate value to make this 10-bit low level the targeted 13-bit SBF length (SBL62, SBL61, SBL60 = 1, 0, 1).

Example If LIN is to be transmitted under the following conditions

- Base clock of UART6 = 5 MHz (set by clock selection register 6 (CKSR6))
- Target baud rate value = 19200 bps

To realize the above baud rate value, the length of a 13-bit SBF is as follows if baud rate generator control register 6 (BRGC6) is set to 130.

- 13-bit SBF length = 0.2 μs × 130 × 2 × 13 = 676 μs

To realize a 13-bit SBF length in 10 bits, set a value 1.3 times the targeted baud rate to BRGC6. In this example, set 169 to BRGC6. The transmission length of a 10-bit low level in this case is as follows, and matches the 13-bit SBF length.

- 10-bit low-level transmission length = 0.2 μs × 169 × 2 × 10 = 676 μs

If the number of bits set by BRGC6 runs short, adjust the number of bits by setting the base clock of UART6.

Figure 14-22. Example of Setting Procedure of SBF Transmission (Flowchart)

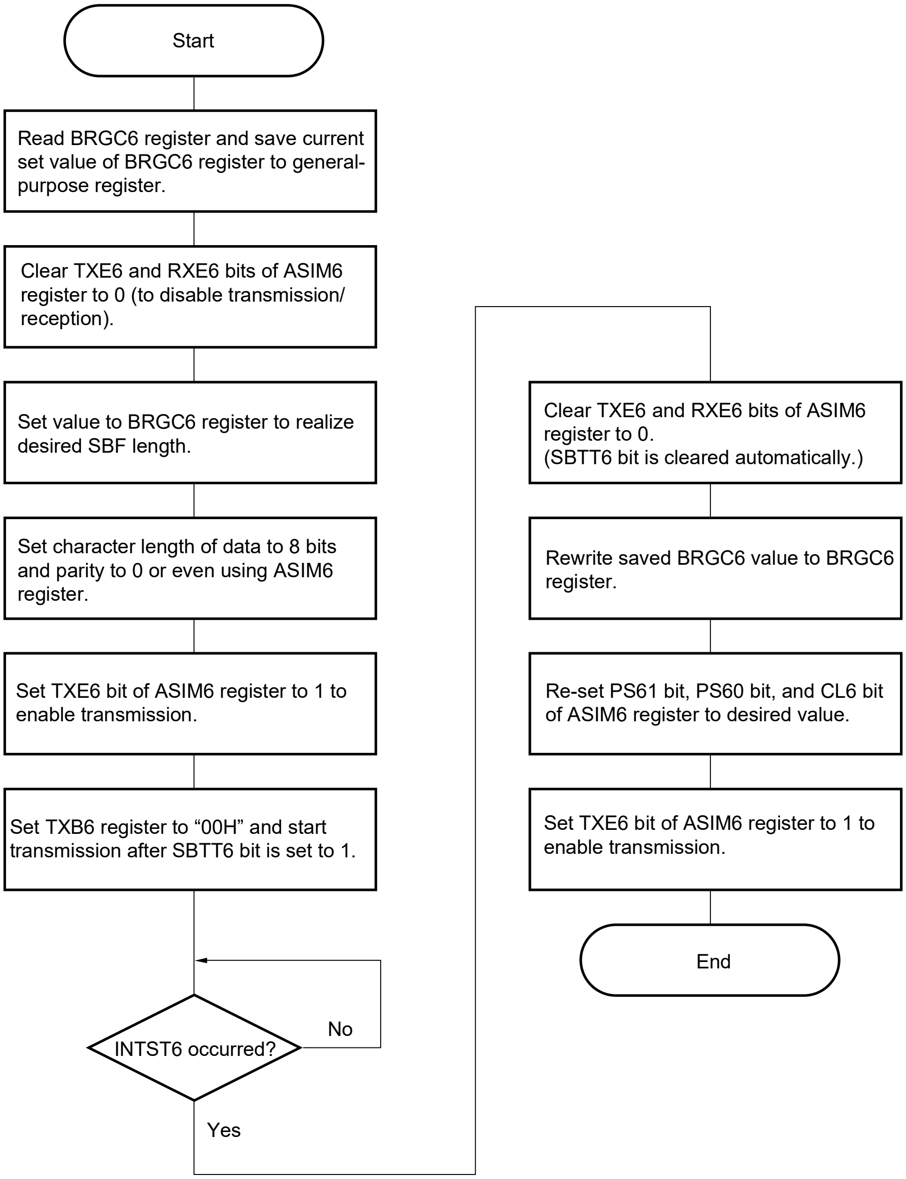
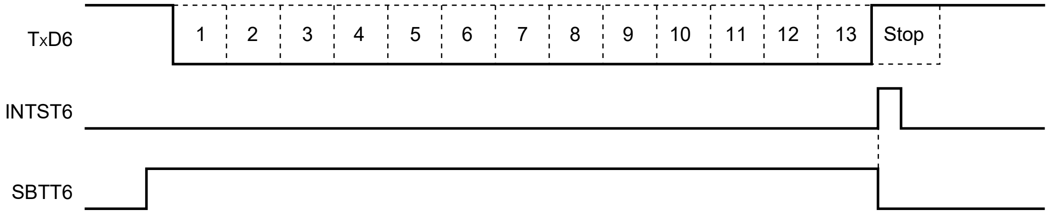

Start
Read BRGC6 register and save current set value of BRGC6 register to general-purpose register.
Clear TXE6 and RXE6 bits of ASIM6 register to 0 (to disable transmission/ reception).
Set value to BRGC6 register to realize desired SBF length.
Set character length of data to 8 bits and parity to 0 or even using ASIM6 register.
Set TXE6 bit of ASIM6 register to 1 to enable transmission.
Set TXB6 register to “00H” and start transmission after SBTT6 bit is set to 1.
INTST6 occurred?
No
Yes
Clear TXE6 and RXE6 bits of ASIM6 register to 0.
(SBTT6 bit is cleared automatically.)
Rewrite saved BRGC6 value to BRGC6 register.
Re-set PS61 bit, PS60 bit, and CL6 bit of ASIM6 register to desired value.
Set TXE6 bit of ASIM6 register to 1 to enable transmission.
End

Figure 14-23. SBF Transmission



Remark TxD6: TxD6 pin (output)

INTST6: Transmission completion interrupt request

SBTT6: Bit 5 of asynchronous serial interface control register 6 (ASICL6)

(i) SBF reception

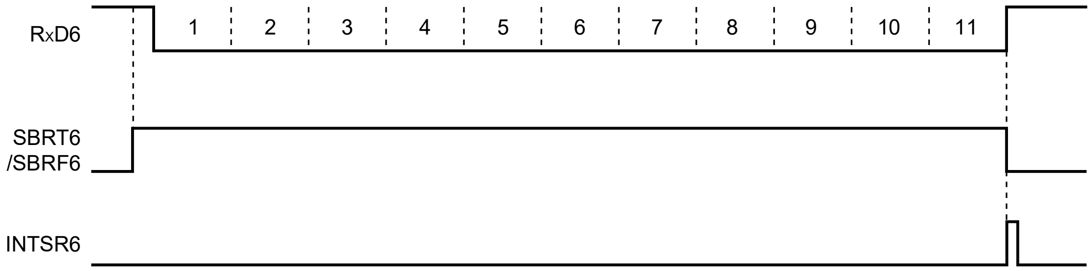
When the device is incorporated in LIN, the SBF (Synchronous Break Field) reception control function is used for reception. For the reception operation of LIN, refer to **Figure 14-2 LIN Reception Operation**.

Reception is enabled when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is set to 1 and then bit 5 (RXE6) of ASIM6 is set to 1. SBF reception is enabled when bit 6 (SBRT6) of asynchronous serial interface control register 6 (ASICL6) is set to 1. In the SBF reception enabled status, the RxD6 pin is sampled and the start bit is detected in the same manner as the normal reception enable status.

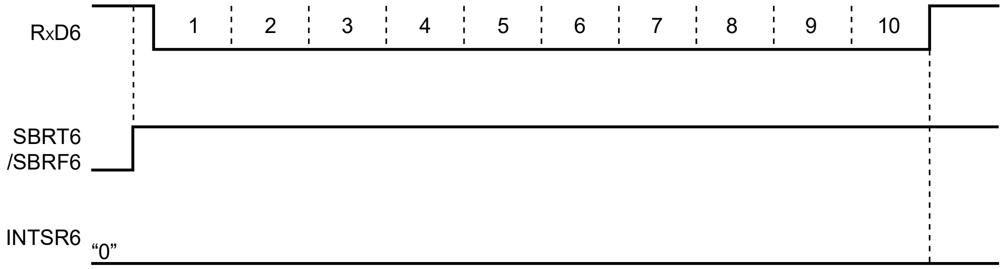
When the start bit has been detected, reception is started, and serial data is sequentially stored in receive shift register 6 (RXS6) at the set baud rate. When the stop bit is received and if the width of SBF is 11 bits or more, a reception completion interrupt request (INTSR6) is generated as normal processing. At this time, the SBRF6 and SBRT6 bits are automatically cleared, and SBF reception ends. Detection of errors, such as OVE6, PE6, and FE6 (bits 0 to 2 of asynchronous serial interface reception error status register 6 (ASIS6)) is suppressed, and error detection processing of UART communication is not performed. In addition, data transfer between receive shift register 6 (RXS6) and receive buffer register 6 (RXB6) is not performed, and the reset value of FFH is retained. If the width of SBF is 10 bits or less, an interrupt does not occur as error processing after the stop bit has been received, and the SBF reception mode is restored. In this case, the SBRF6 and SBRT6 bits are not cleared.

Figure 14-24. SBF Reception

1. Normal SBF reception (stop bit is detected with a width of more than 10.5 bits)



2. SBF reception error (stop bit is detected with a width of 10.5 bits or less)



Remark	RxD6:	RxD6 pin (input)
	SBRT6:	Bit 6 of asynchronous serial interface control register 6 (ASICL6)
	SBRF6:	Bit 7 of ASICL6
	INTSR6:	Reception completion interrupt request

14.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and an 8-bit programmable counter, and generates a serial clock for transmission/reception of UART6.

Separate 8-bit counters are provided for transmission and reception.

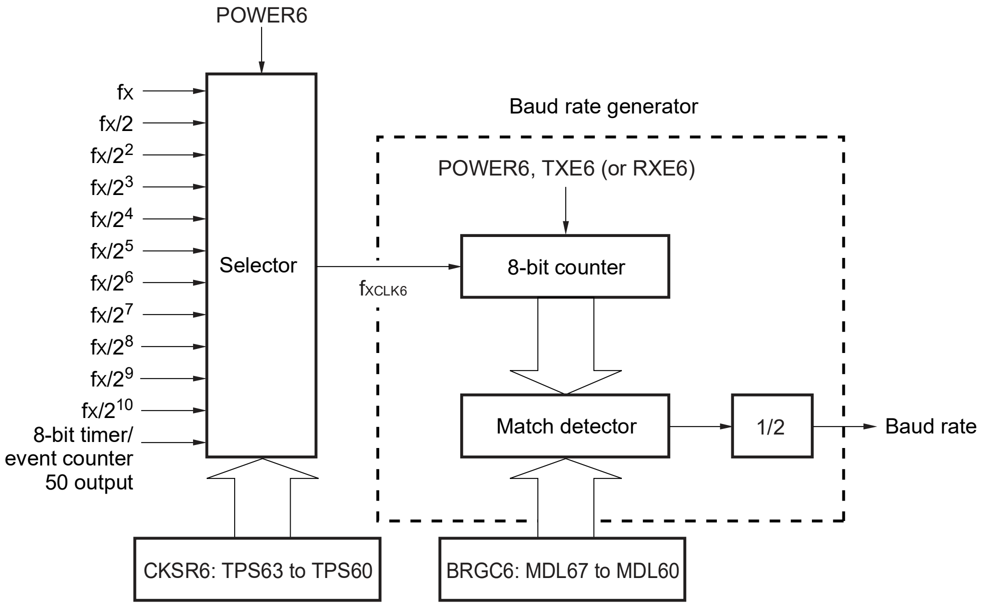
(1) Configuration of baud rate generator

- Base clock
  The clock selected by bits 3 to 0 (TPS63 to TPS60) of clock selection register 6 (CKSR6) is supplied to each module when bit 7 (POWER6) of asynchronous serial interface operation mode register 6 (ASIM6) is 1. This clock is called the base clock and its frequency is called f_{XCLK6}. The base clock is fixed to low level when POWER6 = 0.

- Transmission counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 6 (TXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.
  It starts counting when POWER6 = 1 and TXE6 = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit buffer register 6 (TXB6).
  If data are continuously transmitted, the counter is cleared to 0 again when one frame of data has been completely transmitted. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until POWER6 or TXE6 is cleared to 0.

- Reception counter
  This counter stops operation, cleared to 0, when bit 7 (POWER6) or bit 5 (RXE6) of asynchronous serial interface operation mode register 6 (ASIM6) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 14-25. Configuration of Baud Rate Generator



Remark POWER6: Bit 7 of asynchronous serial interface operation mode register 6 (ASIM6)

TXE6: Bit 6 of ASIM6

RXE6: Bit 5 of ASIM6

CKSR6: Clock selection register 6

BRGC6: Baud rate generator control register 6

(2) Generation of serial clock

A serial clock can be generated by using clock selection register 6 (CKSR6) and baud rate generator control register 6 (BRGC6).

Select the clock to be input to the 8-bit counter by using bits 3 to 0 (TPS63 to TPS60) of CKSR6.

Bits 7 to 0 (MDL67 to MDL60) of BRGC6 can be used to select the division value of the 8-bit counter.

(a) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

Remark f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register
k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 8, 9, 10, ..., 255)

(b) Error of baud rate

The baud rate error can be calculated by the following expression.

- 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 at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 10 MHz = 10,000,000 Hz
Set value of MDL67 to MDL60 bits of BRGC6 register = 00100001B (k = 33)
Target baud rate = 153600 bps

Baud rate = 10 M/(2 × 33)
= 10000000/(2 × 33) = 151,515 [bps]

Error = (151515/153600 – 1) × 100
= –1.357 [%]

(3) Example of setting baud rate

Table 14-4. Set Data of Baud Rate Generator

Baud Rate [bps]	f_X = 10.0 MHz				f_X = 8.38 MHz				f_X = 4.19 MHz			
	TPS63 to TPS60	k	Calculated Value	ERR[%]	TPS63 to TPS60	k	Calculated Value	ERR[%]	TPS63 to TPS60	k	Calculated Value	ERR[%]
600	6H	130	601	0.16	6H	109	601	0.11	5H	109	601	0.11
1200	5H	130	1202	0.16	5H	109	1201	0.11	4H	109	1201	0.11
2400	4H	130	2404	0.16	4H	109	2403	0.11	3H	109	2403	0.11
4800	3H	130	4808	0.16	3H	109	4805	0.11	2H	109	4805	0.11
9600	2H	130	9615	0.16	2H	109	9610	0.11	1H	109	9610	0.11
10400	2H	120	10417	0.16	2H	101	10371	0.28	1H	101	10475	–0.28
19200	1H	130	19231	0.16	1H	109	19220	0.11	0H	109	19220	0.11
31250	1H	80	31250	0.00	0H	134	31268	0.06	0H	67	31268	0.06
38400	0H	130	38462	0.16	0H	109	38440	0.11	0H	55	38090	–0.80
76800	0H	65	76923	0.16	0H	55	76182	–0.80	0H	27	77593	1.03
115200	0H	43	116279	0.94	0H	36	116389	1.03	0H	18	116389	1.03
153600	0H	33	151515	–1.36	0H	27	155185	1.03	0H	14	149643	–2.58
230400	0H	22	227272	–1.36	0H	18	232778	1.03	0H	9	232778	1.03

Remark TPS63 to TPS60: Bits 3 to 0 of clock selection register 6 (CKSR6) (setting of base clock (f_{XCLK6}))

k: Value set by MDL67 to MDL60 bits of baud rate generator control register 6 (BRGC6) (k = 8, 9, 10, ..., 255)

f_X: High-speed system clock oscillation frequency

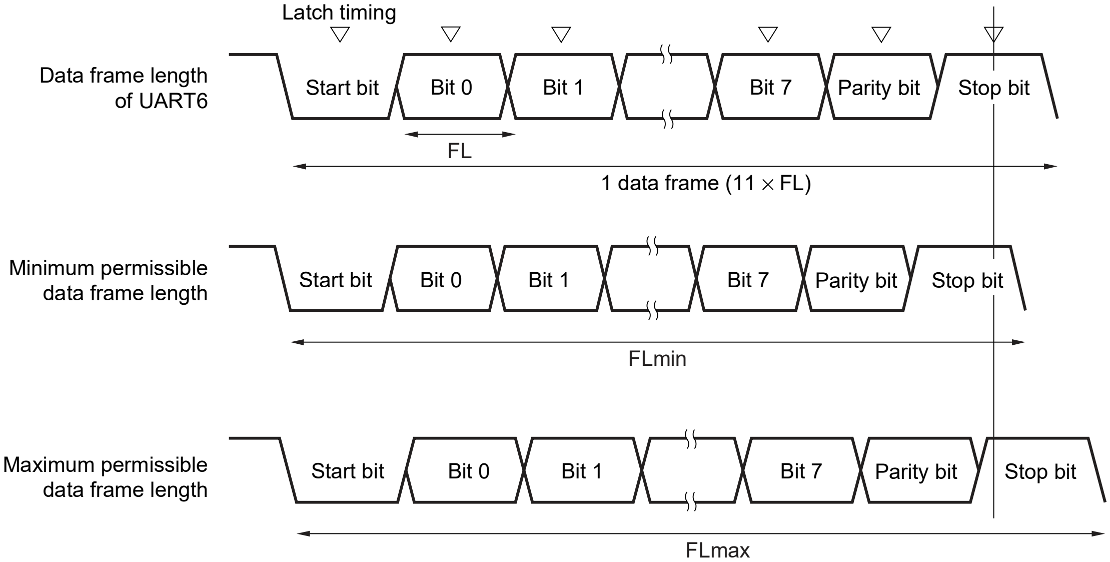
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-26. Permissible Baud Rate Range During Reception



As shown in Figure 14-26, the latch timing of the receive data is determined by the counter set by baud rate generator control register 6 (BRGC6) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART6
k: Set value of BRGC6
FL: 1-bit data length
Margin of latch timing: 2 clocks

Minimum permissible data frame length: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k}\text{FL}$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\text{Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}$$

The permissible baud rate error between UART6 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 14-5. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

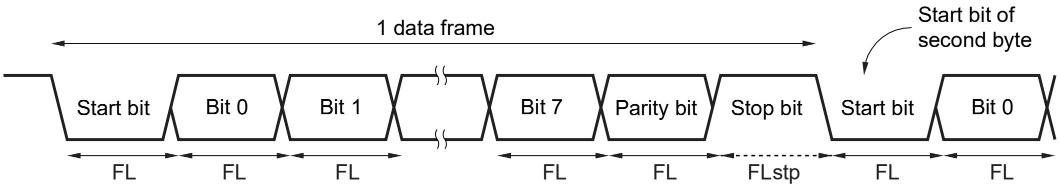
Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC6

(5) Data frame length during continuous transmission

When data is continuously transmitted, the data frame length from a stop bit to the next start bit is extended by two clocks of base clock from the normal value. However, the result of communication is not affected because the timing is initialized on the reception side when the start bit is detected.

Figure 14-27. Data Frame Length During Continuous Transmission



Where the 1-bit data length is FL, the stop bit length is FLstp, and base clock frequency is f_{XCLK6}, the following expression is satisfied.

$$FLstp = FL + 2/f_{XCLK6}$$

Therefore, the data frame length during continuous transmission is:

$$\text{Data frame length} = 11 \times FL + 2/f_{XCLK6}$$

CHAPTER 15 SERIAL INTERFACE CSI10

15.1 Functions of Serial Interface CSI10

Serial interface CSI10 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, refer to **15.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{\text{SCK10}}$) and two serial data lines (SI10 and SO10).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode can be used connecting peripheral ICs and display controllers with a clocked serial interface.

For details, refer to **15.4.2 3-wire serial I/O mode**

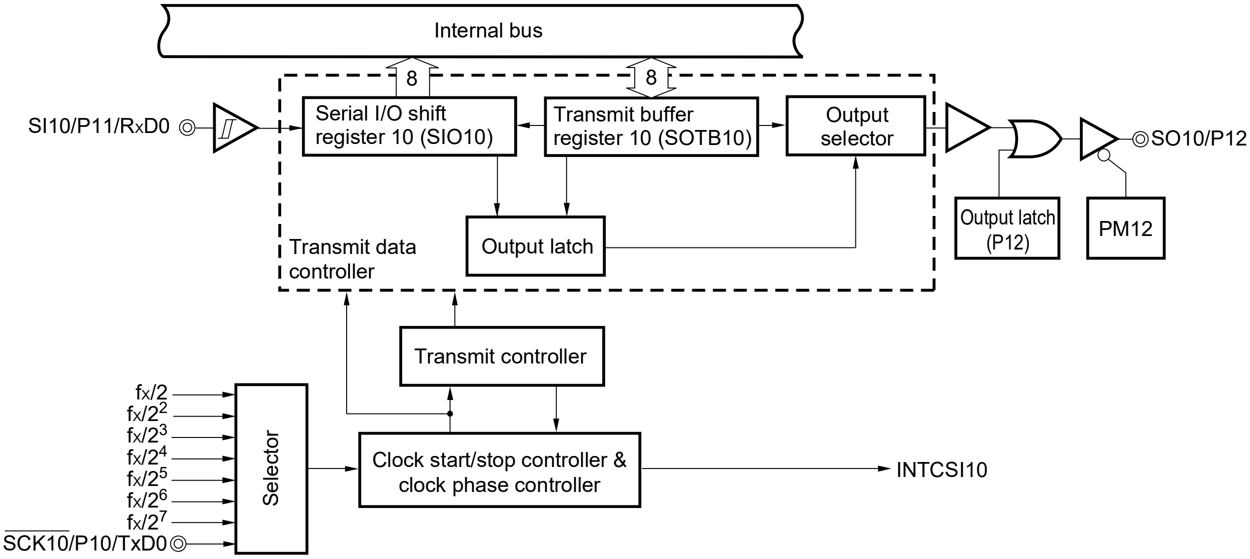
15.2 Configuration of Serial Interface CSI10

Serial interface CSI10 includes the following hardware.

Table 15-1. Configuration of Serial Interface CSI10

Item	Configuration
Registers	Transmit buffer register 10 (SOTB10) Serial I/O shift register 10 (SIO10)
Control registers	Serial operation mode register 10 (CSIM10) Serial clock selection register 10 (CSIC10) Port mode register 0 (PM0) or port mode register 1 (PM1) Port register 0 (P0) or port register 1 (P1)

Figure 15-1. Block Diagram of Serial Interface CSI10



(1) Transmit buffer register 10 (SOTB10)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB10 when bit 7 (CSIE10) and bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) are 1.

The data written to SOTB10 is converted from parallel data into serial data by serial I/O shift register 10, and output to the serial output pin (SO10).

SOTB10 can be written or read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes SOTB10 undefined.

Caution Do not access SOTB10 when CSOT10 = 1 (during serial communication).

(2) Serial I/O shift register 10 (SIO10)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO10 if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

During reception, the data is read from the serial input pin (SI10) to SIO10.

$\overline{\text{RESET}}$ input clears SIO10 to 00H.

Caution Do not access SIO10 when CSOT10 = 1 (during serial communication).

15.3 Registers Controlling Serial Interface CSI10

Serial interface CSI10 is controlled by the following four registers.

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Serial operation mode register 10 (CSIM10)

CSIM10 is used to select the operation mode and enable or disable operation.
CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Figure 15-2. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD10[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 6]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. When using P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, or P12/SO10 as a general-purpose port, refer to **CHAPTER 4 PORT FUNCTIONS**, **Caution 3** of **Figure 15-3**, and **Table 15-2**.

3. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

4. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

5. The SO10 output is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

6. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

Caution Be sure to clear bit 5 to 0.

(2) Serial clock selection register 10 (CSIC10)

CSIC10 specifies the timing of the data transmission/reception and sets the serial clock.

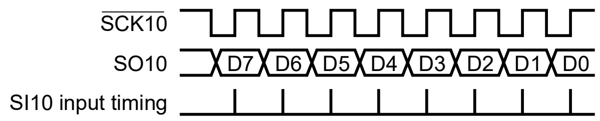
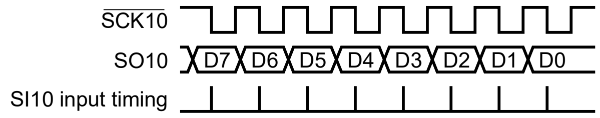
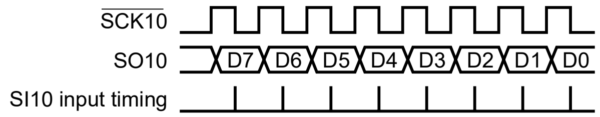
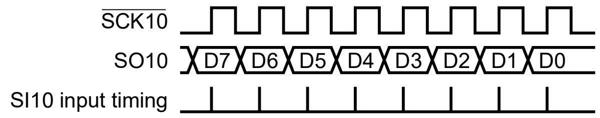
CSIC10 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CSIC10 to 00H.

Figure 15-3. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	Specification of data transmission/reception timing	Type
0	0	 $\overline{\text{SCK10}}$ SO10 D7 D6 D5 D4 D3 D2 D1 D0 SI10 input timing	1
0	1	 $\overline{\text{SCK10}}$ SO10 D7 D6 D5 D4 D3 D2 D1 D0 SI10 input timing	2
1	0	 $\overline{\text{SCK10}}$ SO10 D7 D6 D5 D4 D3 D2 D1 D0 SI10 input timing	3
1	1	 $\overline{\text{SCK10}}$ SO10 D7 D6 D5 D4 D3 D2 D1 D0 SI10 input timing	4

CKS102	CKS101	CKS100	CSI10 serial clock selection[Note]	Mode
0	0	0	$f_X/2$ (5 MHz)	Master mode
0	0	1	$f_X/2^2$ (2.5 MHz)	Master mode
0	1	0	$f_X/2^3$ (1.25 MHz)	Master mode
0	1	1	$f_X/2^4$ (625 kHz)	Master mode
1	0	0	$f_X/2^5$ (312.5 kHz)	Master mode
1	0	1	$f_X/2^6$ (156.25 kHz)	Master mode
1	1	0	$f_X/2^7$ (78.13 kHz)	Master mode
1	1	1	External clock input to $\overline{\text{SCK10}}$	Slave mode

Note Be sure to set the serial clock so that the following condition is satisfied.

- V_{DD} = 2.7 to 5.5 V: Serial clock ≤ 5 MHz

Cautions 1. When the Ring-OSC clock is selected as the clock supplied to the CPU, the clock of the Ring-OSC oscillator is divided and supplied as the serial clock. At this time, the operation of serial interface CSI10 is not guaranteed.

2. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

3. Clear CKP10 to 0 to use P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 as general-purpose port pins.

4. The phase type of the data clock is type 1 after reset.

Remarks 1. Figures in parentheses are for operation with fx = 10 MHz

2. fx: High-speed system clock oscillation frequency

(3) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using P10/$\overline{\text{SCK10}}$/TxD0 as the clock output pins of the serial interface, set PM10 to 0 and the output latch of P10 to 1. When using P12/SO10 as the data output pins, clear PM12 and the output latches of P12 to 0.

When using P10/$\overline{\text{SCK10}}$/TxD0 as the clock input pins of the serial interface, and P11/SI10/RxD0 as the data input pins, set PM10 and PM11 to 1. At this time, the output latches of P10 and P11 may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM1 to FFH.

Figure 15-4. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 Operation of Serial Interface CSI10

Serial interface CSI10 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

15.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, and P12/SO10 pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode register 10 (CSIM10).
To set the operation stop mode, clear bit 7 (CSIE10) of CSIM10 to 0.

(a) Serial operation mode register 10 (CSIM10)

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. When using P10/$\overline{\text{SCK10}}$/TxD0, P11/SI10/RxD0, or P12/SO10 as a general-purpose port, refer to **CHAPTER 4 PORT FUNCTIONS**, **Caution 3** of **Figure 15-3**, and **Table 15-2**.

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

15.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode can be used for connecting peripheral ICs and display controllers that have a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{\text{SCK10}}$), serial output (SO10), and serial input (SI10) lines.

(1) Registers used

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC10 register (refer to **Figure 15-3**).
<2> Set bits 0, 4, and 6 (CSOT10, DIR10, and TRMD10) of the CSIM10 register (refer to **Figure 15-2**).
<3> Set bit 7 (CSIE10) of the CSIM10 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 10 (SOTB10). → Data transmission/reception is started.
Read data from serial I/O shift register 10 (SIO10). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 15-2. Relationship Between Register Settings and Pins

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 Operation	Pin Function		
									SI10/RxD0/ P11	SO10/P12	$\overline{\text{SCK10}}$/ TxD0/P10
0	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Stop	RxD0/P11	P12	TxD0/ P10[Note 2]
1	0	1	×	×[Note 1]	×[Note 1]	1	×	Slave reception[Note 3]	SI10	P12	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	×[Note 1]	×[Note 1]	0	0	1	×	Slave transmission[Note 3]	RxD0/P11	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	1	×	0	0	1	×	Slave transmission/ reception[Note 3]	SI10	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	0	1	×	×[Note 1]	×[Note 1]	0	1	Master reception	SI10	P12	$\overline{\text{SCK10}}$ (output)
1	1	×[Note 1]	×[Note 1]	0	0	0	1	Master transmission	RxD0/P11	SO10	$\overline{\text{SCK10}}$ (output)
1	1	1	×	0	0	0	1	Master transmission/ reception	SI10	SO10	$\overline{\text{SCK10}}$ (output)

Notes 1. Can be set as port function.

2. To use P10/$\overline{\text{SCK10}}$/TxD0 as port pins, clear CKP10 to 0.

3. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remark

×:	don't care
CSIE10:	Bit 7 of serial operation mode register 10 (CSIM10)
TRMD10:	Bit 6 of CSIM10
CKP10:	Bit 4 of serial clock selection register 10 (CSIC10)
CKS102, CKS101, CKS100:	Bits 2 to 0 of CSIC10
PM1×:	Port mode register
P1×:	Port output latch

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1. Transmission/reception is started when a value is written to transmit buffer register 10 (SOTB10). In addition, data can be received when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

Reception is started when data is read from serial I/O shift register 10 (SIO10).

After communication has been started, bit 0 (CSOT10) of CSIM10 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF10) is set, and CSOT10 is cleared to 0. Then the next communication is enabled.

Caution Do not access the control register and data register when CSOT10 = 1 (during serial communication).

Figure 15-5. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

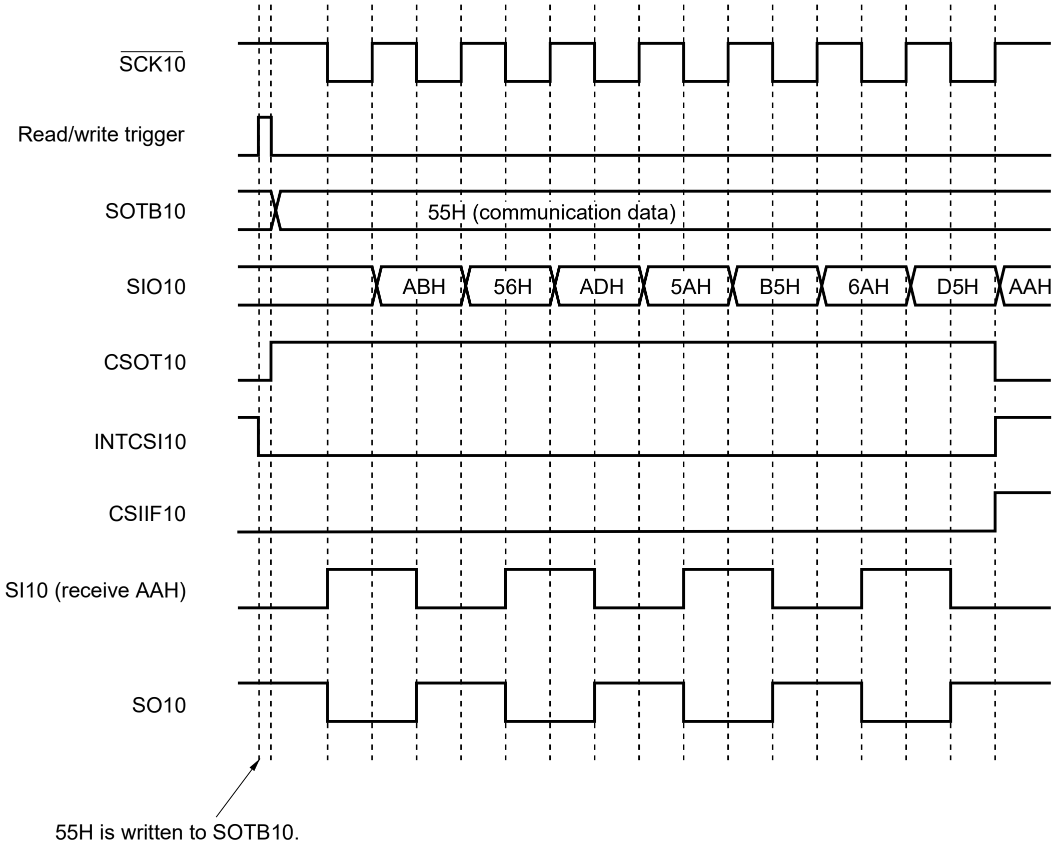


Figure 15-5. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

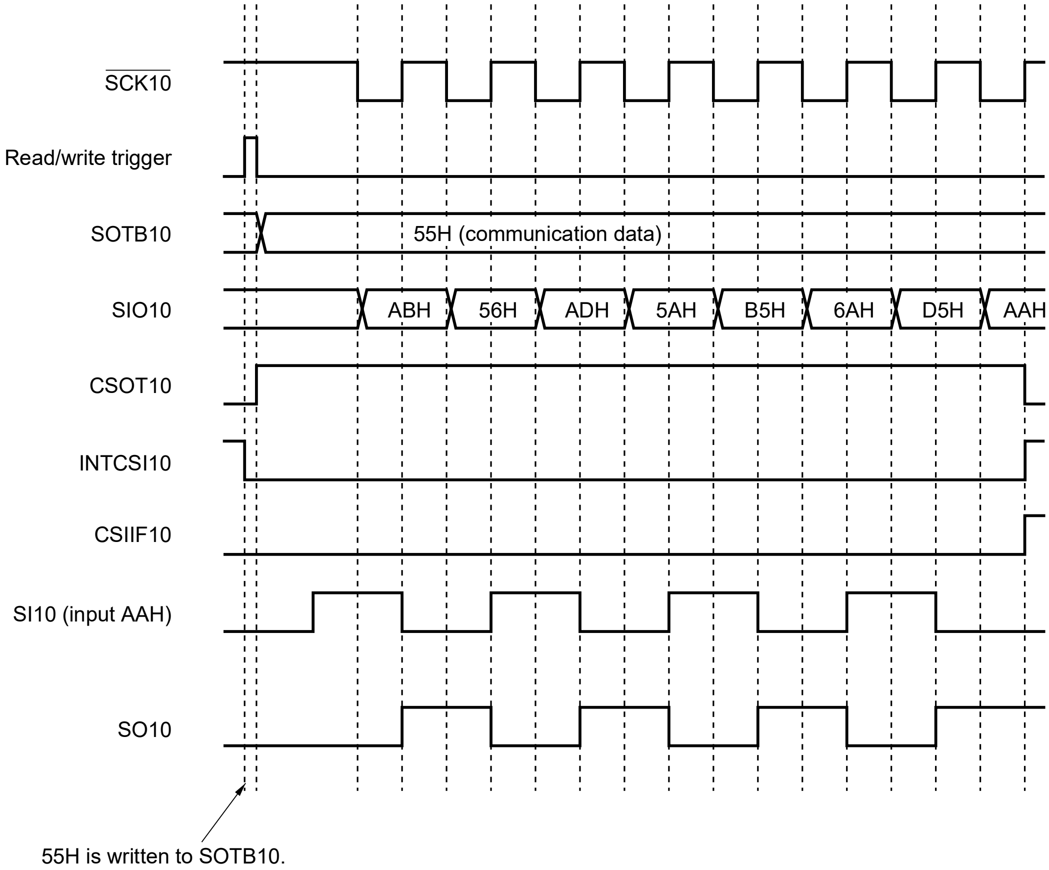
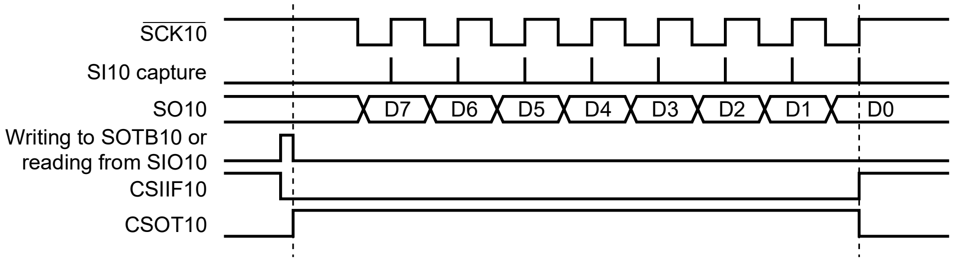
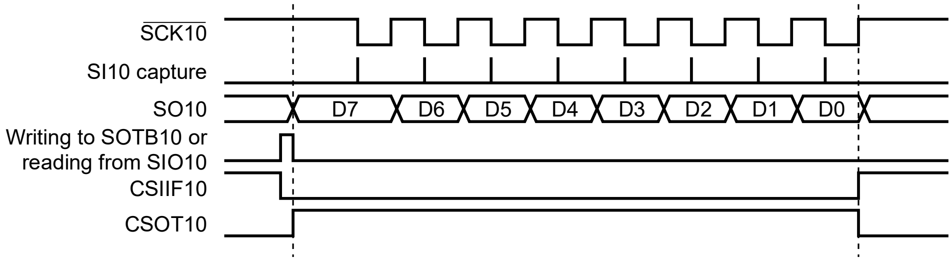


Figure 15-6. Timing of Clock/Data Phase

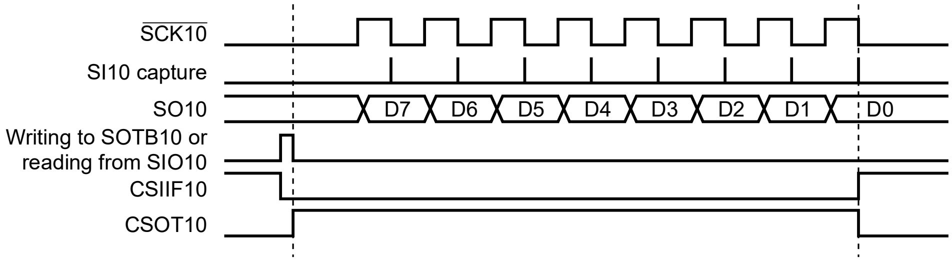
(a) Type 1; CKP10 = 0, DAP10 = 0



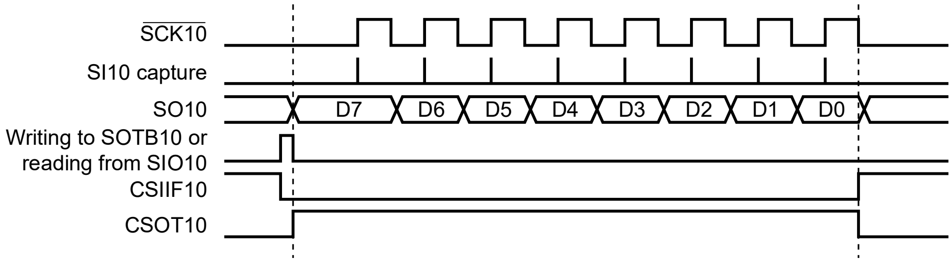
(b) Type 2; CKP10 = 0, DAP10 = 1



(c) Type 3; CKP10 = 1, DAP10 = 0



(d) Type 4; CKP10 = 1, DAP10 = 1

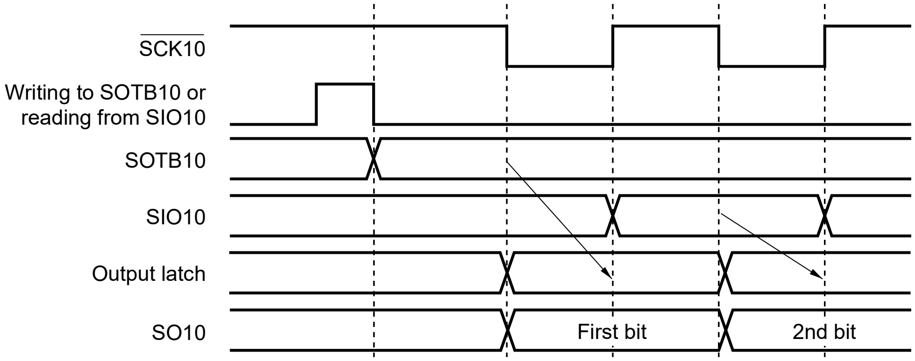


(3) Timing of output to SO10 pin (first bit)

When communication is started, the value of transmit buffer register 10 (SOTB10) is output from the SO10 pin. The output operation of the first bit at this time is described below.

Figure 15-7. Output Operation of First Bit

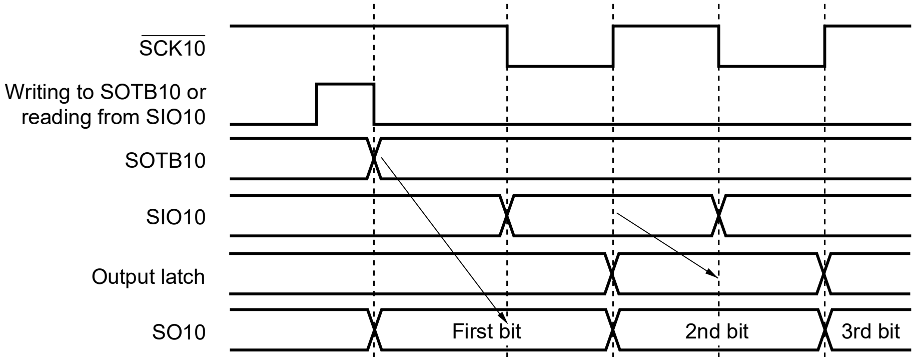
(1) When CKP10 = 0, DAP10 = 0 (or CKP10 = 1, DAP10 = 0)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
SOTB10
SIO10
Output latch
SO10
First bit
2nd bit

The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of $\overline{\text{SCK10}}$, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(2) When CKP10 = 0, DAP10 = 1 (or CKP10 = 1, DAP10 = 1)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
SOTB10
SIO10
Output latch
SO10
First bit
2nd bit
3rd bit

The first bit is directly latched by the SOTB10 register at the falling edge of the write signal of the SOTB10 register or the read signal of the SIO10 register, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

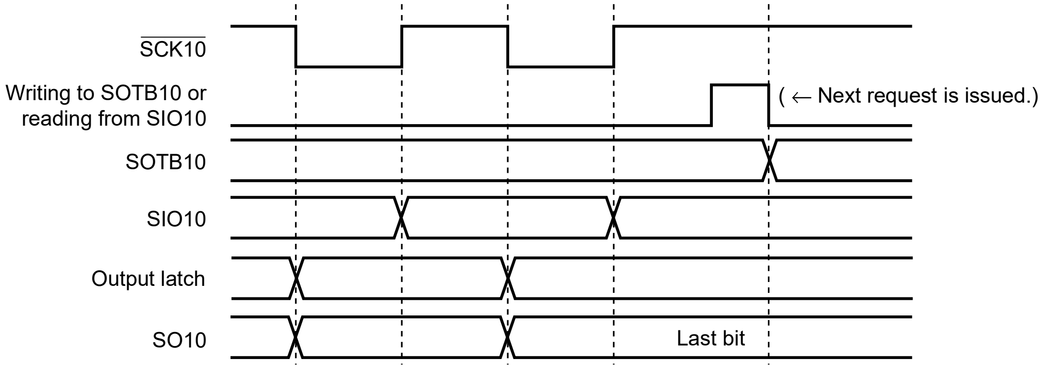
The second and subsequent bits are latched by the SIO10 register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(4) Output value of SO10 pin (last bit)

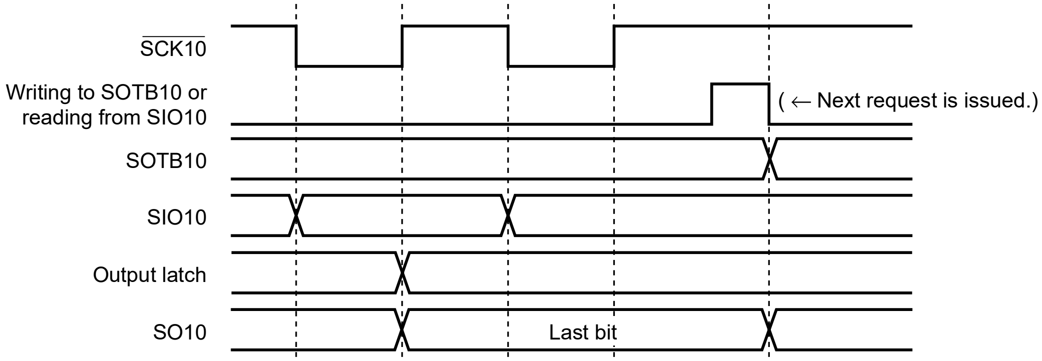
After communication has been completed, the SO10 pin holds the output value of the last bit.

Figure 15-8. Output Value of SO10 Pin (Last Bit)

(1) Type 1; when CKP10 = 0 and DAP10 = 0 (or CKP10 = 1, DAP10 = 0)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
(← Next request is issued.)
SOTB10
SIO10
Output latch
SO10
Last bit

(2) Type 2; when CKP10 = 0 and DAP10 = 1 (or CKP10 = 1, DAP10 = 1)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
(← Next request is issued.)
SOTB10
SIO10
Output latch
SO10
Last bit

(5) SO10 output

The status of the SO10 output is as follows if bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) is cleared to 0.

Table 15-3. SO10 Output Status

TRMD10	DAP10	DIR10	SO10 Output[Note 1]
TRMD10 = 0[Note 2]	–	–	Outputs low level[Note 2].
TRMD10 = 1	DAP10 = 0	–	Value of SO10 latch (low-level output)
	DAP10 = 1	DIR10 = 0	Value of bit 7 of SOTB10
		DIR10 = 1	Value of bit 0 of SOTB10

Notes 1. The actual output of the SO10/P12 pin is determined according to PM12 and P12, as well as the SO10 output.

2. Status after reset

Caution If a value is written to TRMD10, DAP10, and DIR10, the output value of SO10 changes.

CHAPTER 16 INTERRUPT FUNCTIONS

16.1 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 a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L).
Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupts with the same priority are generated simultaneously, each interrupt is serviced according to its predetermined priority (refer to **Table 16-1**).
A standby release signal is generated and STOP and HALT modes are released.
Eight external interrupt requests and 15 internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

16.2 Interrupt Sources and Configuration

A total of 24 interrupt sources exist for maskable and software interrupts (refer to **Table 16-1**).

Table 16-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTLVI	Low-voltage detection[Note 3]	Internal	0004H	(A)
	1	INTP0	Pin input edge detection	External	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 reception error generation	Internal	0012H	(A)
	8	INTSR6	End of UART6 reception		0014H	
	9	INTST6	End of UART6 transmission		0016H	
	10	INTCSI10/ INTST0	End of CSI10 communication/end of UART0 transmission		0018H	
	11	INTTMH1	Match between TMH1 and CMP01 (when compare register is specified)		001AH	
	12	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		001CH	
	13	INTTM50	Match between TM50 and CR50 (when compare register is specified)		001EH	
	14	INTTM000	Match between TM00 and CR000 (when compare register is specified), TI010 pin valid edge detection (when capture register is specified)		0020H	
	15	INTTM010	Match between TM00 and CR010 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		0022H	
	16	INTAD	End of A/D conversion		0024H	
	17	INTSR0	End of UART0 reception or reception error generation		0026H	
	18	INTWTI	Watch timer reference time interval signal		0028H	
	19	INTTM51	Match between TM51 and CR51 (when compare register is specified)		002AH	
	20	INTKR	Key interrupt detection	External	002CH	(C)
	21	INTWT	Watch timer overflow	Internal	002EH	(A)
	22	INTP6	Pin input edge detection	External	0030H	(B)

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 16-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Table 16-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Software	–	BRK	BRK instruction execution	–	003EH	(D)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on clear			
		LVI	Low-voltage detection[Note 3]			
		Clock monitor	High-speed system clock oscillation stop detection			
		WDT	WDT overflow			

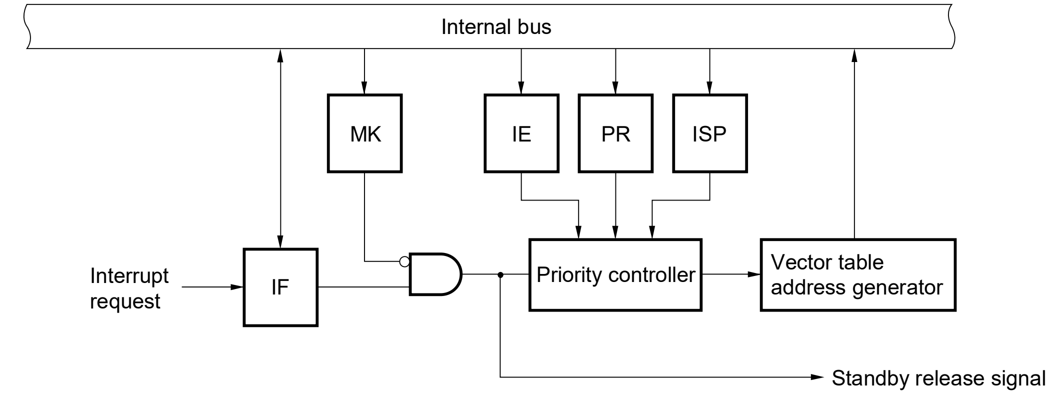
Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 28 is the lowest.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 16-1.

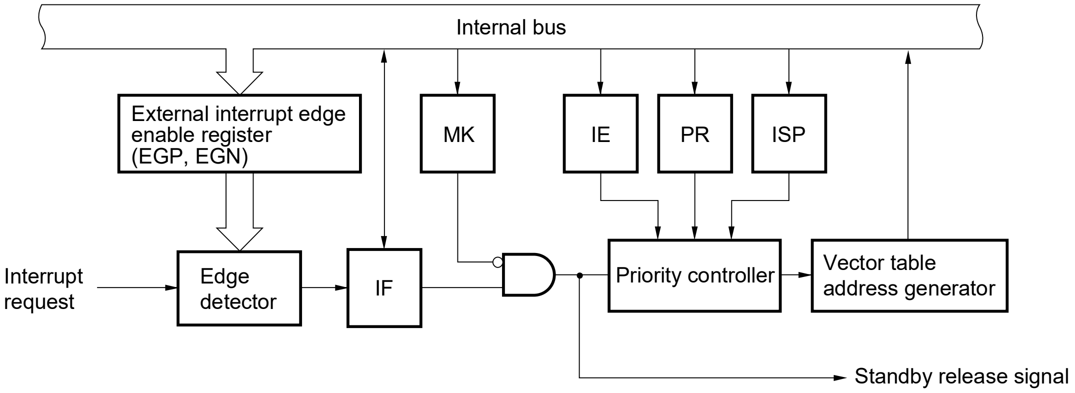
3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



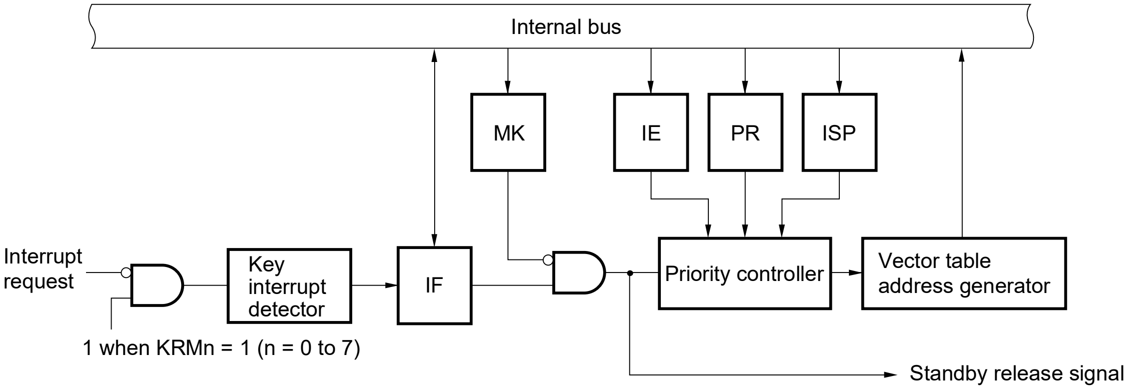
(B) External maskable interrupt (INTP0 to INTP6)



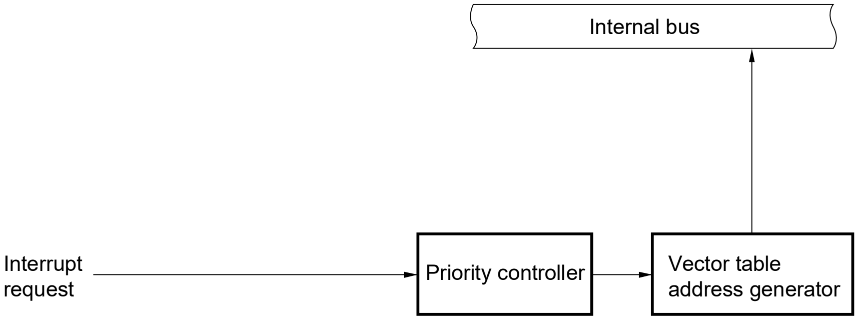
IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag

Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag
KRM:	Key return mode register

16.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L)
- Interrupt mask flag register (MK0L, MK0H, MK1L)
- Priority specification flag register (PR0L, PR0H, PR1L)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 16-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTLVI	LVIIF	IF0L	LVIMK	MK0L	LVIPR	PR0L
INTP0	PIF0		PMK0		PPR0	
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTSRE6	SREIF6		SREMK6		SREPR6	
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PR0H
INTST6	STIF6		STMK6		STPR6	
INTCSI10	DUALIF0[Note 1]		DUALMK0[Note 2]		DUALPR0[Note 2]	
INTST0						
INTTMH1	TMIFH1		TMMKH1		TMPRH1	
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM000	TMIF000		TMMK000		TMPR000	
INTTM010	TMIF010		TMMK010		TMPR010	
INTAD	ADIF	IF1L	ADMK	MK1L	ADPR	PR1L
INTSR0	SRIF0		SRMK0		SRPR0	
INTWTI	WTIIF		WTIMK		WTIPR	
INTTM51	TMIF51		TMMK51		TMPR51	
INTKR	KRIF		KRMK		KRPR	
INTWT	WTIF		WTMK		WTPR	
INTP6	PIF6		PMK6		PPR6	

Notes 1. If either of the two types of interrupt sources is generated, these flags are set (1).

2. Both types of interrupt sources are supported.

(1) Interrupt request flag registers (IF0L, IF0H, IF1L)

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 $\overline{\text{RESET}}$ input.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, and IF1L can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H are combined to form 16-bit register IF0, they can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L)

Address: FFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF0	STIF6	SRIF6

Address: FFE2H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	0[Note]	PIF6	WTIF	KRIF	TMIF51	WTIIF	SRIF0	ADIF

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note Be sure to clear bit 7 of IF1L to 0.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

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 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.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, and MK1L can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H are combined to form 16-bit register MK0, they can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 16-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L)

Address: FFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0	STMK6	SRMK6

Address: FFE6H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	1[Note]	PMK6	WTMK	KRMK	TMMK51	WTIMK	SRMK0	ADMK

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note Be sure to set bit 7 of MK1L to 1.

(3) Priority specification flag registers (PR0L, PR0H, PR1L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, and PR1L can be set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H are combined to form 16-bit register PR0, they can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 16-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L)

Address: FFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPR0	STPR6	SRPR6

Address: FFEAH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	1[Note]	PPR6	WTPR	KRPR	TMPR51	WTIPR	SRPR0	ADPR

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Note Be sure to set bit 7 of PR1L to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP6.

EGP and EGN can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 16-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 16-3 shows the ports corresponding to EGPn and EGNn.

Table 16-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 6

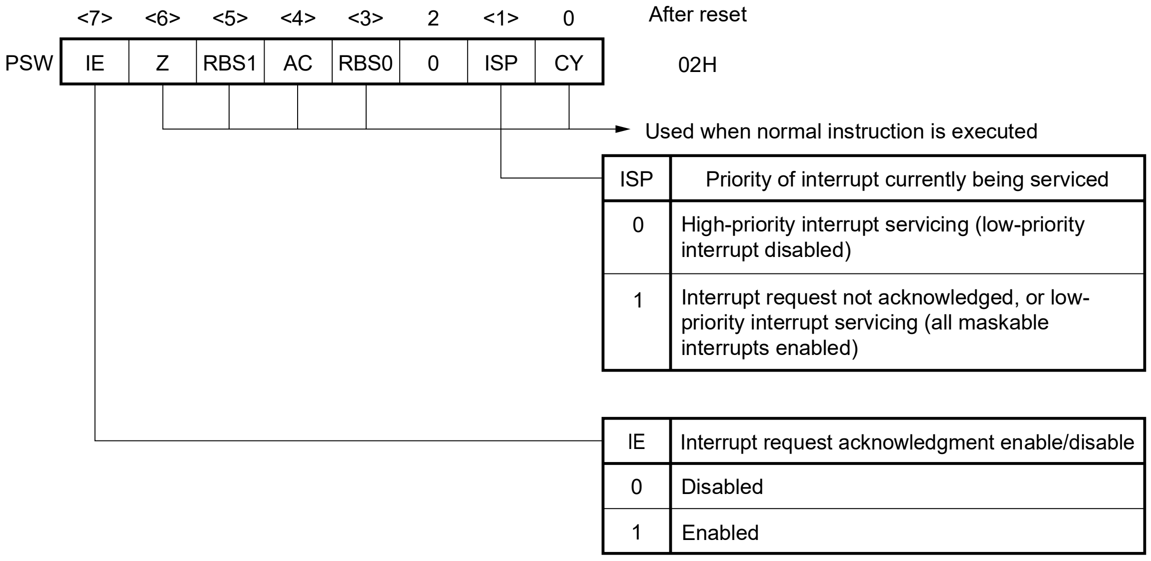
(5) Program status word (PSW)

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 ISP flag 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. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. 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.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 16-6. Format of Program Status Word



	<7>	<6>	<5>	<4>	<3>	2	<1>	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

16.4 Interrupt Servicing Operations

16.4.1 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 (when the ISP flag is reset to 0).

The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 16-4 below.

For the interrupt request acknowledgment timing, refer to **Figures 16-8** and **16-9**.

Table 16-4. Time from Generation of Maskable Interrupt Request Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

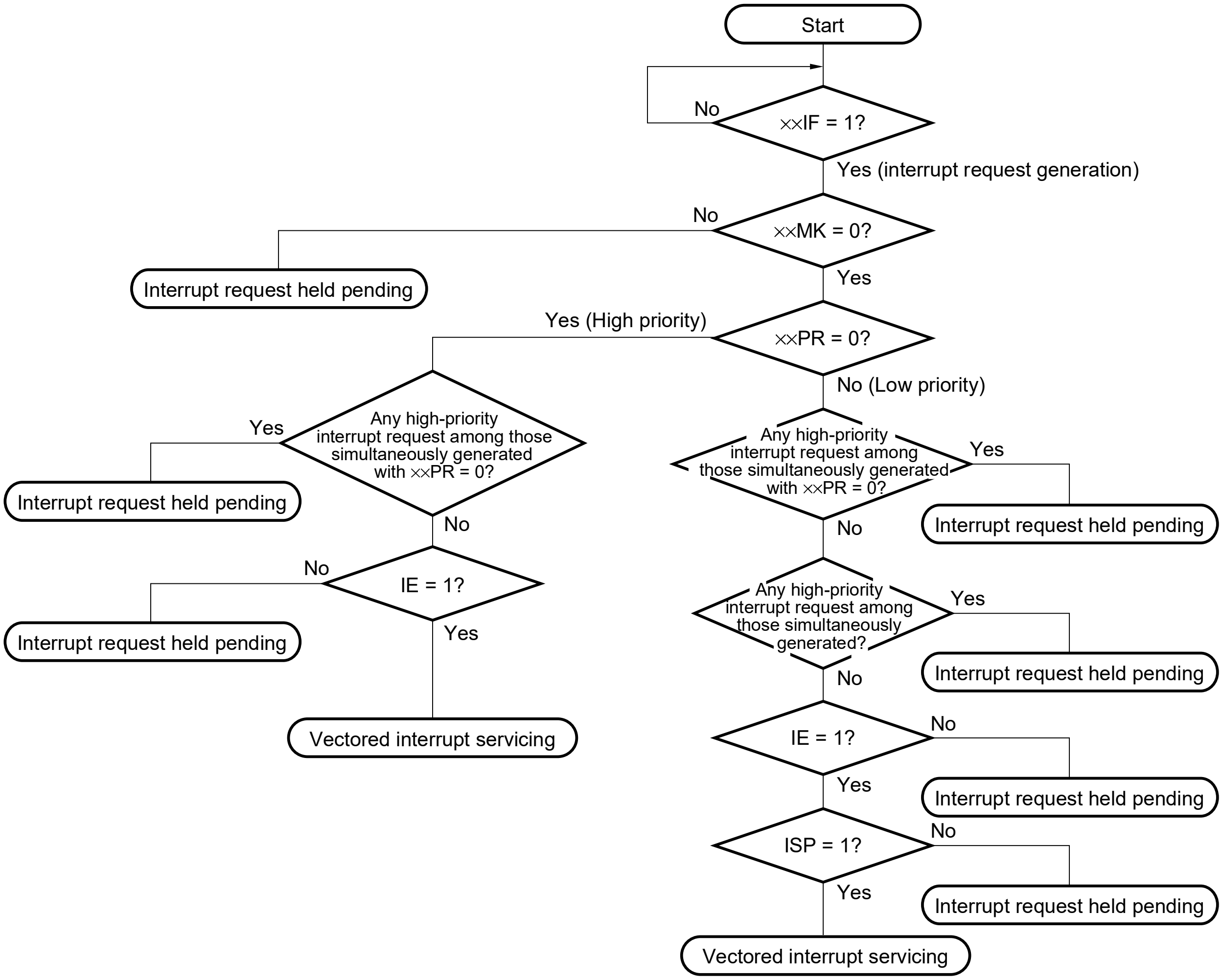
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-7 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 ISP flag. The vector table data determined for each interrupt request is loaded into the PC and branched.

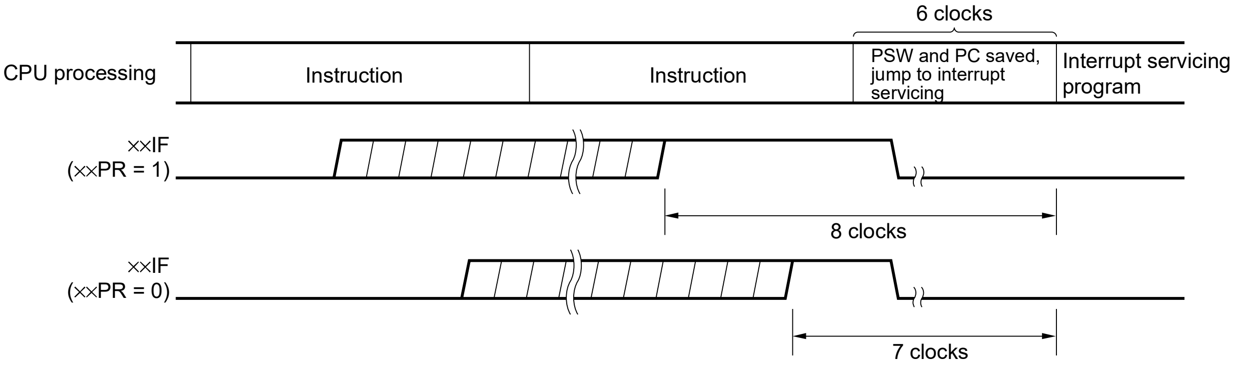
Restoring from an interrupt is possible by using the RETI instruction.

Figure 16-7. Interrupt Request Acknowledgment Processing Algorithm


Start
No
××IF = 1?
Yes (interrupt request generation)
No
××MK = 0?
Interrupt request held pending
Yes
Yes (High priority)
××PR = 0?
No (Low priority)
Yes
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Interrupt request held pending
No
No
IE = 1?
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Yes
Interrupt request held pending
ISP = 1?
No
Yes
Interrupt request held pending
Vectored interrupt servicing

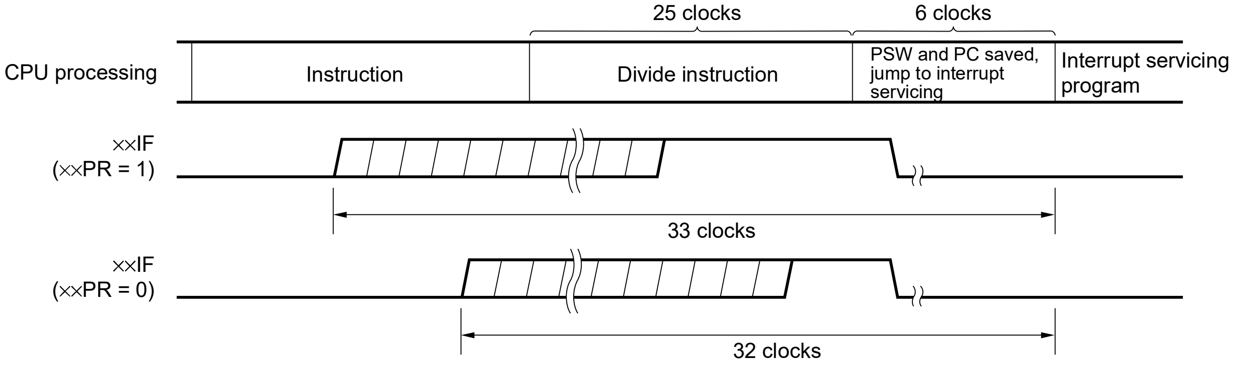
××IF:	Interrupt request flag
××MK:	Interrupt mask flag
××PR:	Priority specification flag
IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP:	Flag that indicates the priority level of the interrupt currently being serviced (0 = High-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 16-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 16-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

16.4.2 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 (003EH, 003FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Do not use the RETI instruction for restoring from the software interrupt.

16.4.3 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 equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority 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 16-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 16-10 shows multiple interrupt servicing examples.

Table 16-5. Relationship Between Interrupt Request Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request		Maskable Interrupt Request				Software Interrupt Request
		PR = 0		PR = 1		
Interrupt Being Serviced		IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
Software interrupt		○	×	○	×	○

Remarks **1.** ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

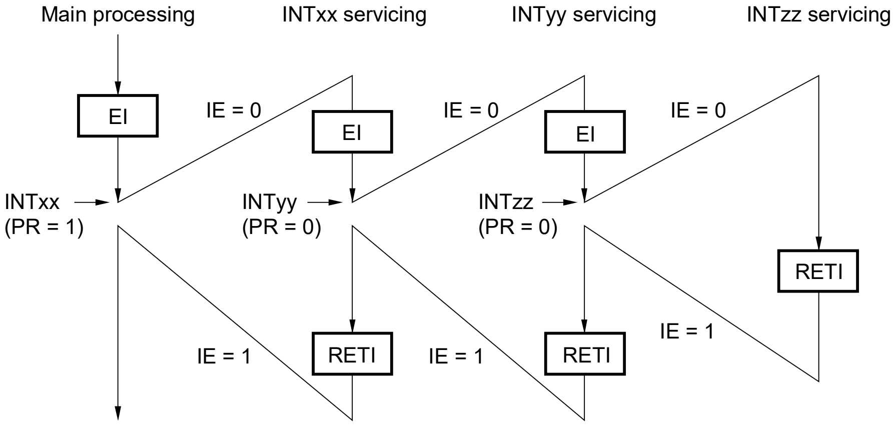
ISP = 0: An interrupt with higher priority is being serviced.
ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in PR0L, PR0H, and PR1L.

PR = 0: Higher priority level
PR = 1: Lower priority level

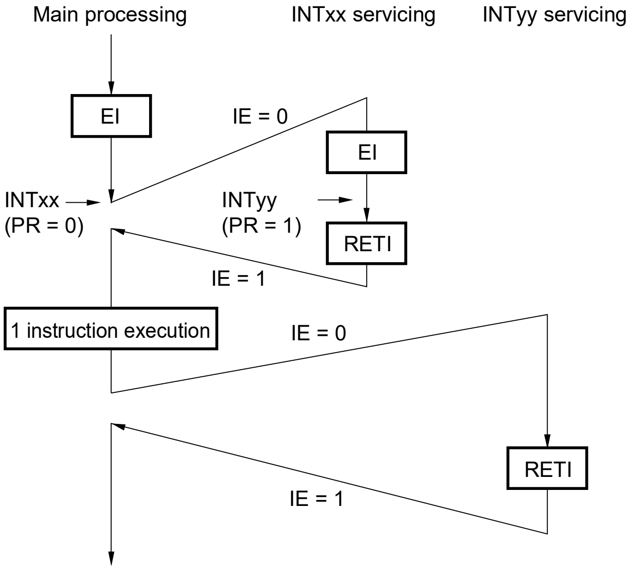
Figure 16-10. Examples of Multiple Interrupt Servicing (1/2)

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

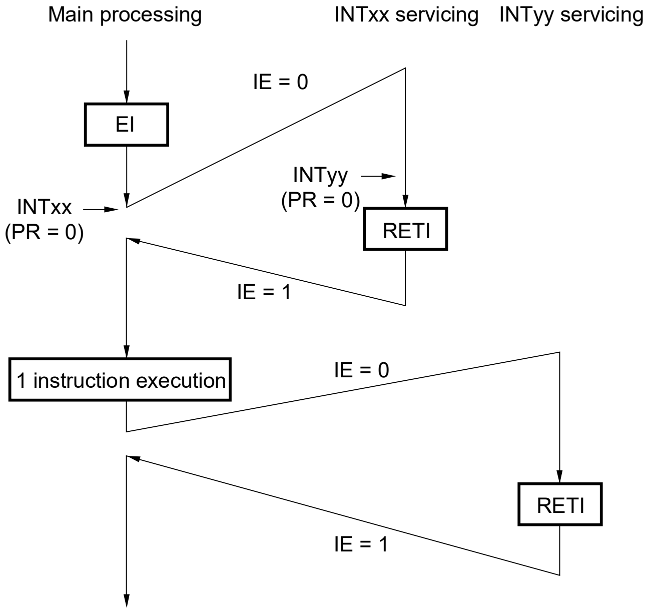


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0:	Higher priority level
PR = 1:	Lower priority level
IE = 0:	Interrupt request acknowledgment disabled

Figure 16-10. 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 = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

16.4.4 Interrupt request hold

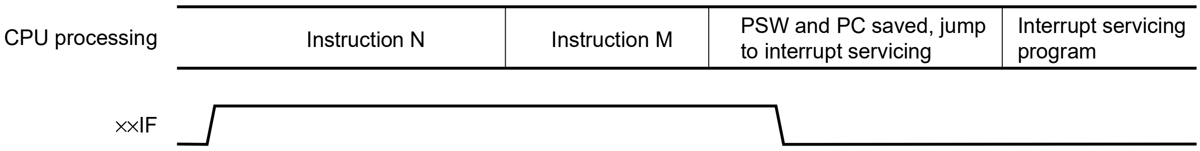
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, $addr16
- BF PSW. bit, $addr16
- BTCLR PSW. bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR0L, PR0H, and PR1L registers

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 16-11 shows the timing at which interrupt requests are held pending.

Figure 16-11. 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

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 17 KEY INTERRUPT FUNCTION

17.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

Table 17-1. Assignment of Key Interrupt Detection Pins

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5	Controls KR5 signal in 1-bit units.
KRM6	Controls KR6 signal in 1-bit units.
KRM7	Controls KR7 signal in 1-bit units.

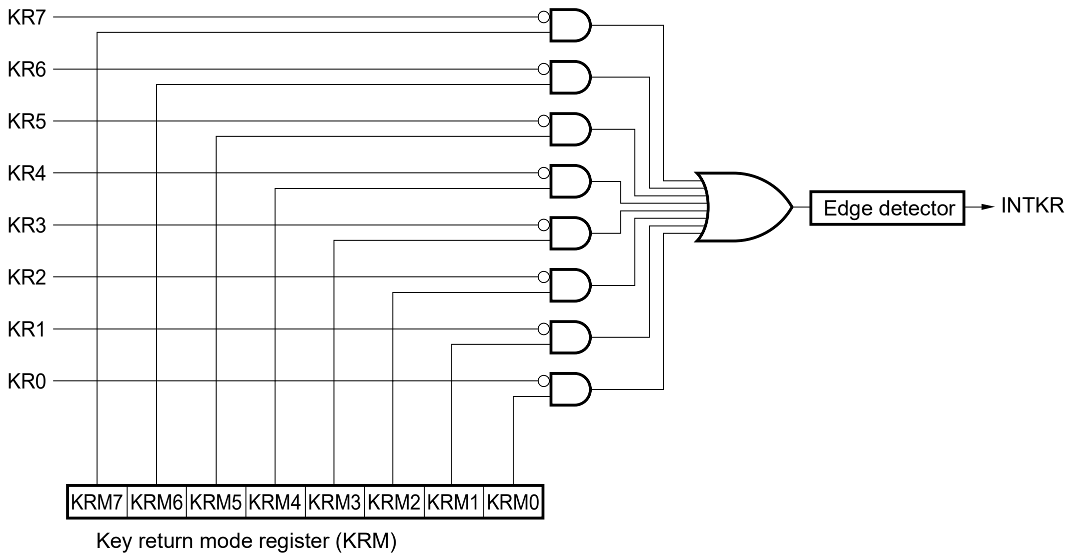
17.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 17-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 17-1. Block Diagram of Key Interrupt



17.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals, respectively.

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

$\overline{\text{RESET}}$ input clears KRM to 00H.

Figure 17-2. Format of Key Return Mode Register (KRM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM0 to KRM7 bits used is set to 1, set bits 0 to 7 (PU70 to PU77) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

3. The bits not used in the key interrupt mode can be used as normal ports.

CHAPTER 18 STANDBY FUNCTION

18.1 Standby Function and Configuration

18.1.1 Standby function

Table 18-1. Relationship Between Operation Clocks in Each Operation Status

Status	High-Speed System Clock Oscillator		Ring-OSC Oscillator			Subsystem Clock Oscillator	CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
Operation Mode	MSTOP = 0 MCC = 0	MSTOP = 1 MCC = 1	**Note 1**	**Note 2**				MCM0 = 0	MCM0 = 1
				RSTOP = 0	RSTOP = 1				
Reset	Stopped		Stopped			Oscillating	Ring-OSC	Stopped	
STOP			Oscillating	Oscillating	Stopped		**Note 3**	Stopped	
HALT	Oscillating	Stopped					**Note 4**	Ring-OSC	High-speed system clock

Notes 1. When “Cannot be stopped” is selected for Ring-OSC by the option byte.

2. When “Can be stopped by software” is selected for Ring-OSC by the option byte.

3. Operates using the CPU clock at STOP instruction execution.

4. Operates using the CPU clock at HALT instruction execution.

Caution The RSTOP setting is valid only when “Can be stopped by software” is set for Ring-OSC by the option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
MCC: Bit 7 of the processor clock control register (PCC)
RSTOP: Bit 0 of the Ring-OSC mode register (RCM)
MCM0: Bit 0 of the main clock mode register (MCM)

The standby function is designed to reduce the operating current of the system. The following two 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, Ring-OSC 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.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator stops, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be released by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

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. STOP mode can be used only when CPU is operating on the high-speed system clock or Ring-OSC clock. HALT mode can be used when CPU is operating on the high-speed system clock, Ring-OSC clock, or subsystem clock. However, when the STOP instruction is executed during Ring-OSC clock operation, the high-speed system clock oscillator stops, but Ring-OSC oscillator does not stop.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

4. If the Ring-OSC oscillator is operating before the STOP mode is set, oscillation of the Ring-OSC clock cannot be stopped in the STOP mode. However, when the Ring-OSC clock is used as the CPU clock, the CPU operation is stopped for 17/f_R (s) after STOP mode is released.

18.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, refer to **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

This is the status register of the high-speed system clock oscillation stabilization time counter. If the Ring-OSC clock is used as the CPU clock, the high-speed system clock oscillation stabilization time can be checked.

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

Reset release (reset by $\overline{\text{RESET}}$ input, POC, LVI, clock monitor, and WDT), the STOP instruction, MSTOP (bit 7 of MOC register) = 1, and MCC (bit 7 of PCC register) = 1 clear OSTC to 00H.

Figure 18-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status		
						f_{XP} = 10 MHz	f_{XP} = 16 MHz
1	0	0	0	0	$2^{11}/f_{XP}$ min.	204.8 μs min.	128 μs min.
1	1	0	0	0	$2^{13}/f_{XP}$ min.	819.2 μs min.	512 μs min.
1	1	1	0	0	$2^{14}/f_{XP}$ min.	1.64 ms min.	1.02 ms min.
1	1	1	1	0	$2^{15}/f_{XP}$ min.	3.27 ms min.	2.04 ms min.
1	1	1	1	1	$2^{16}/f_{XP}$ min.	6.55 ms min.	4.09 ms min.

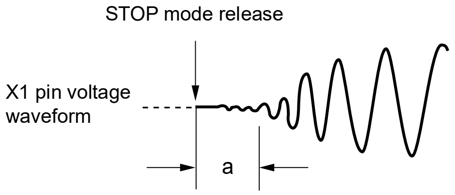
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the Ring-OSC clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The high-speed system clock oscillation stabilization time counter counts only during the oscillation stabilization time set by OSTS. Therefore, note that only the statuses during the oscillation stabilization time set by OSTS are set to OSTC after STOP mode has been released.

3. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts ("a" below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: High-speed system clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the high-speed system clock oscillation stabilization wait time when STOP mode is released. The wait time set by OSTS is valid only after STOP mode is released when the high-speed system clock is selected as the CPU clock. After STOP mode is released when the Ring-OSC clock is selected as the CPU clock, check the oscillation stabilization time using OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 05H.

Figure 18-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

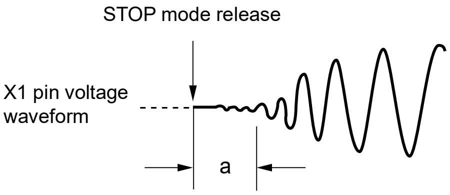
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_{XP} = 10 MHz	f_{XP} = 16 MHz
0	0	1	$2^{11}/f_{XP}$	204.8 μs	128 μs
0	1	0	$2^{13}/f_{XP}$	819.2 μs	512 μs
0	1	1	$2^{14}/f_{XP}$	1.64 ms	1.02 ms
1	0	0	$2^{15}/f_{XP}$	3.27 ms	2.04 ms
1	0	1	$2^{16}/f_{XP}$	6.55 ms	4.09 ms
Other than above			Setting prohibited		

Cautions 1. If the STOP mode is entered and then released while the Ring-OSC clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The high-speed system clock oscillation stabilization time counter counts only during the oscillation stabilization time set by OSTS. Therefore, note that only the statuses during the oscillation stabilization time set by OSTS are set to OSTC after STOP mode has been released.

2. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts (“a” below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: High-speed system clock oscillation frequency

18.2 Standby Function Operation

18.2.1 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, Ring-OSC clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 18-2. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on High-Speed System Clock				When HALT Instruction Is Executed While CPU Is Operating on Ring-OSC Clock			
		When Ring-OSC Oscillation Continues		When Ring-OSC Oscillation Stopped[Note 1]		When High-Speed System Clock Oscillation Continues		When High-Speed System Clock Oscillation Stopped	
Item		When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used
System clock		Clock supply to the CPU is stopped.							
CPU		Operation stopped							
Port (latch)		Status before HALT mode was set is retained							
16-bit timer/event counter 00		Operable				Operation not guaranteed			
8-bit timer/event counter 50		Operable				Operation not guaranteed when count clock other than TI50 is selected			
8-bit timer/event counter 51		Operable				Operation not guaranteed when count clock other than TI51 is selected			
8-bit timer H0		Operable				Operation not guaranteed when count clock other than TM50 output is selected during 8-bit timer/event counter 50 operation			
8-bit timer H1		Operable				Operation not guaranteed when count clock other than $f_R/2^7$ is selected			
Watch timer		Operable	Operable[Note 2]	Operable	Operable[Note 2]	Operable[Note 3]	Operation not guaranteed	Operable[Note 3]	Operation not guaranteed
Watchdog timer	Ring-OSC cannot be stopped[Note 4]	Operable		–		Operable			
	Ring-OSC can be stopped[Note 4]	Operation stopped							
A/D converter		Operable				Operation not guaranteed			
Serial interface	UART0	Operable				Operation not guaranteed when serial clock other than TM50 output is selected during TM50 operation			
	UART6	Operable							
	CSI10	Operable				Operation not guaranteed when serial clock other than external $\overline{\text{SCK10}}$ is selected			
Clock monitor		Operable		Operation stopped		Operable		Operation stopped	
Power-on-clear function		Operable							
Low-voltage detection function		Operable							
External interrupt		Operable							

Notes 1. When "Stopped by software" is selected for Ring-OSC by the option byte and Ring-OSC is stopped by software (for the option byte, refer to **CHAPTER 23 OPTION BYTE**).

2. Operable when the high-speed system clock is selected.

3. Operation not guaranteed when other than subsystem clock is selected.

4. "Ring-OSC cannot be stopped" or "Ring-OSC can be stopped by software" can be selected by the option byte.

Table 18-2. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock			
		When High-Speed System Clock Oscillation Continues		When High-Speed System Clock Oscillation Stopped	
Item		When Ring-OSC Oscillation Continues	When Ring-OSC Oscillation Stopped[Note 1]	When Ring-OSC Oscillation Continues	When Ring-OSC Oscillation Stopped[Note 1]
System clock		Clock supply to the CPU is stopped.			
CPU		Operation stopped			
Port (latch)		Status before HALT mode was set is retained			
16-bit timer/event counter 00		Operable		Operation stopped	
8-bit timer/event counter 50		Operable		Operable only when TI50 is selected as the count clock	
8-bit timer/event counter 51		Operable		Operable only when TI51 is selected as the count clock	
8-bit timer H0		Operable		Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation	
8-bit timer H1		Operable	Operable only when the high-speed system clock is selected as the count clock	Operable only when $f_R/2^7$ is selected as the count clock	Operation stopped
Watch timer		Operable		Operation guaranteed only when subsystem clock is selected	
Watchdog timer	Ring-OSC cannot be stopped[Note 2]	Operable	–	Operable	–
	Ring-OSC can be stopped[Note 2]	Operation stopped			
A/D converter		Operable		Not operable	
Serial interface	UART0	Operable		Operable only when TM50 output is selected as the serial clock during TM50 operation	
	UART6	Operable			
	CSI10	Operable		Operable only when external $\overline{SCK10}$ is selected as the serial clock	
Clock monitor		Operable	Operation stopped		
Power-on-clear function		Operable			
Low-voltage detection function		Operable			
External interrupt		Operable			

Notes 1. When "Stopped by software" is selected for Ring-OSC by the option byte and Ring-OSC is stopped by software (for the option byte, refer to **CHAPTER 23 OPTION BYTE**).

2. "Ring-OSC cannot be stopped" or "Ring-OSC can be stopped by software" can be selected by the option byte.

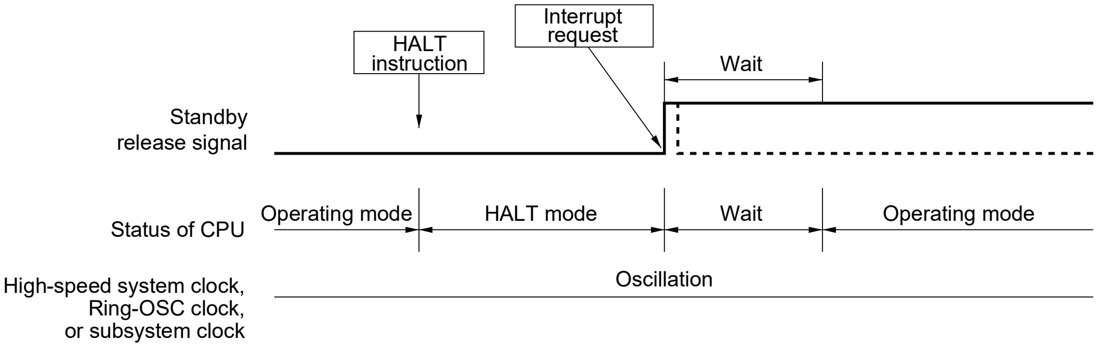
(2) HALT mode release

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 18-3. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

2. The wait time is as follows:

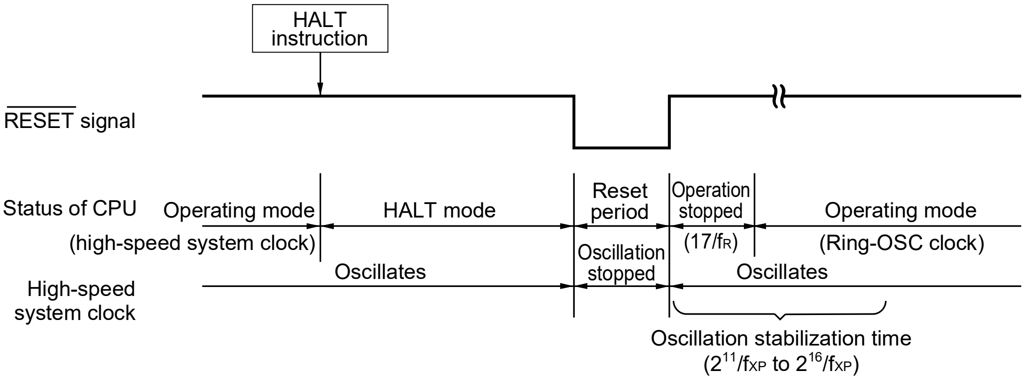
- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

(b) Release by $\overline{\text{RESET}}$ input

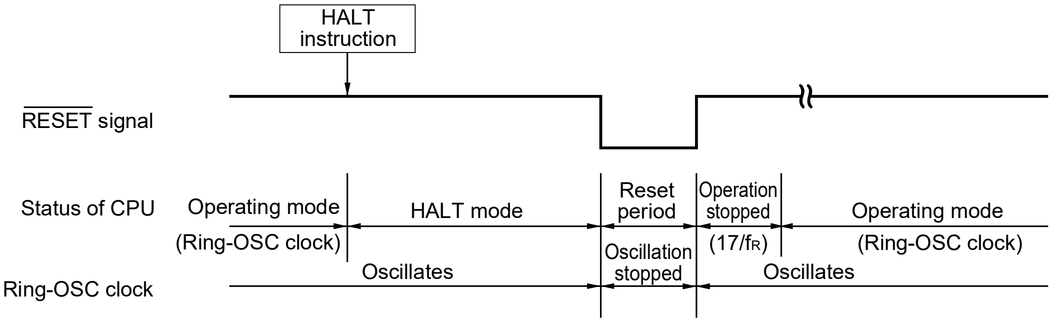
When the $\overline{\text{RESET}}$ signal is input, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 18-4. HALT Mode Release by $\overline{\text{RESET}}$ Input (1/2)

(1) When high-speed system clock is used as CPU clock


HALT instruction
$\overline{\text{RESET}}$ signal
Status of CPU
Operating mode (high-speed system clock)
HALT mode
Reset period
Operation stopped (17/fR)
Operating mode (Ring-OSC clock)
High-speed system clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (2^{11}/fXP to 2^{16}/fXP)

(2) When Ring-OSC clock is used as CPU clock

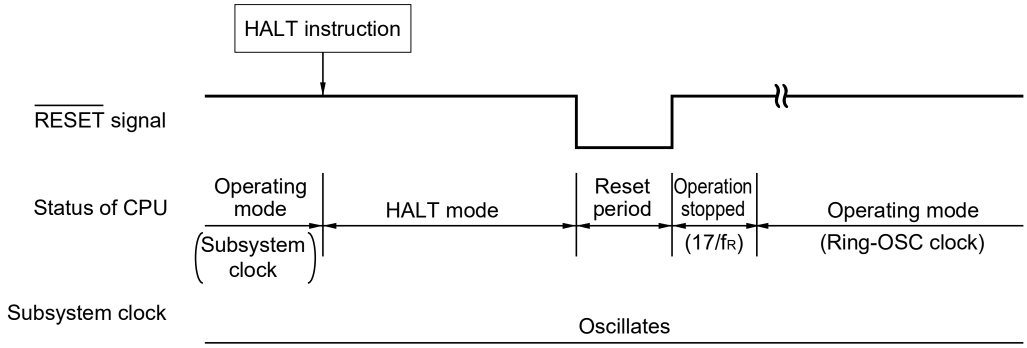

HALT instruction
$\overline{\text{RESET}}$ signal
Status of CPU
Operating mode (Ring-OSC clock)
HALT mode
Reset period
Operation stopped (17/fR)
Operating mode (Ring-OSC clock)
Ring-OSC clock
Oscillates
Oscillation stopped
Oscillates

Remarks 1. fXP: High-speed system clock oscillation frequency

2. fR: Ring-OSC clock oscillation frequency

Figure 18-4. HALT Mode Release by $\overline{\text{RESET}}$ Input (2/2)

(3) When subsystem clock is used as CPU clock



Remark f_R: Ring-OSC clock oscillation frequency

Table 18-3. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: don't care

18.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set when the CPU clock before the setting was the high-speed system clock or Ring-OSC clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 18-4. Operating Statuses in STOP Mode

STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on High-Speed System Clock				When STOP Instruction Is Executed While CPU Is Operating on Ring-OSC Clock	
		When Ring-OSC Oscillation Continues		When Ring-OSC Oscillation Stopped[Note 1]			
Item		When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used	When Subsystem Clock Used	When Subsystem Clock Not Used
System clock		Only high-speed system clock oscillator oscillation is stopped. Clock supply to the CPU is stopped.					
CPU		Operation stopped					
Port (latch)		Status before STOP mode was set is retained					
16-bit timer/event counter 00		Operation stopped					
8-bit timer/event counter 50		Operable only when TI50 is selected as the count clock					
8-bit timer/event counter 51		Operable only when TI51 is selected as the count clock					
8-bit timer H0		Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation					
8-bit timer H1		Operable[Note 2]		Operation stopped		Operable[Note 2]	
Watch timer		Operable[Note 3]	Operation stopped	Operable[Note 3]	Operation stopped	Operable[Note 3]	Operation stopped
Watchdog timer	Ring-OSC cannot be stopped[Note 4]	Operable		–		Operable	
	Ring-OSC can be stopped[Note 4]	Operation stopped					
A/D converter		Operation stopped					
Serial interface	UART0	Operable only when TM50 output is selected as the serial clock during TM50 operation					
	UART6						
	CSI10	Operable only when external $\overline{\text{SCK10}}$ is selected as the serial clock					
Clock monitor		Operation stopped					
Power-on-clear function		Operable					
Low-voltage detection function		Operable					
External interrupt		Operable					

Notes 1. When “Stopped by software” is selected for Ring-OSC by the option byte and Ring-OSC is stopped by software (for the option byte, refer to **CHAPTER 23 OPTION BYTE**).

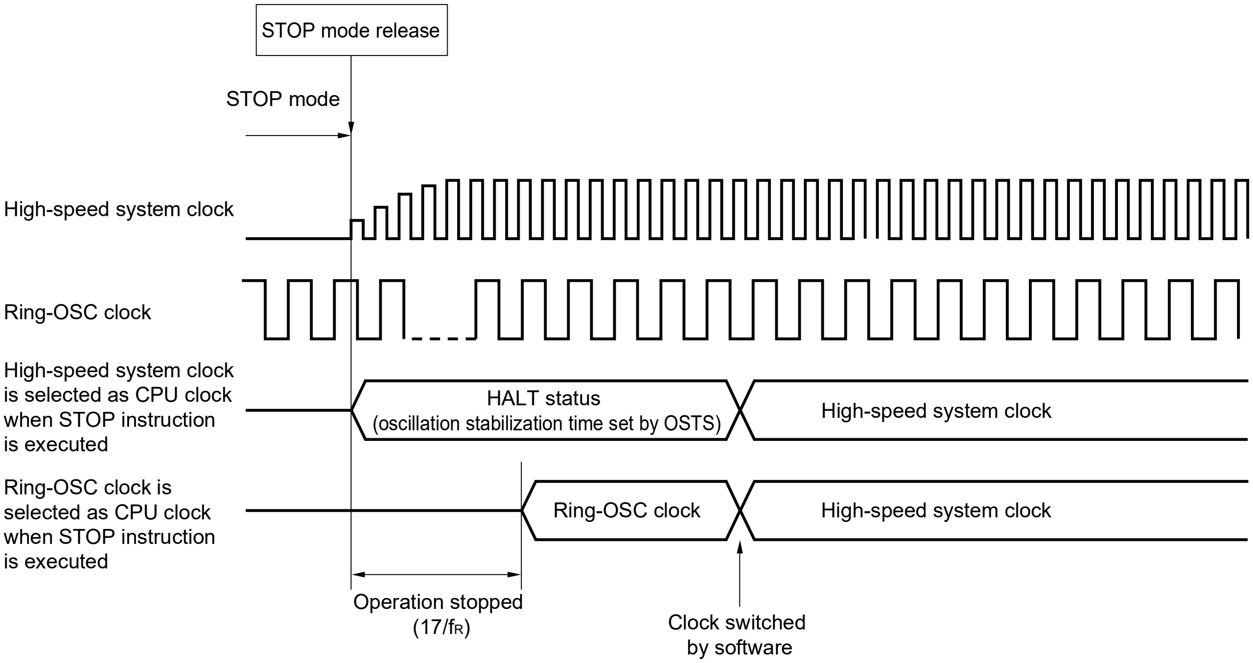
2. Operable only when $f_R/2^7$ is selected as the count clock.

3. Operable when the subsystem clock is selected.

4. “Ring-OSC cannot be stopped” or “Ring-OSC can be stopped by software” can be selected by the option byte.

(2) STOP mode release

Figure 18-5. Operation Timing When STOP Mode Is Released


STOP mode release
STOP mode
High-speed system clock
Ring-OSC clock
High-speed system clock is selected as CPU clock when STOP instruction is executed
HALT status
(oscillation stabilization time set by OSTS)
High-speed system clock
Ring-OSC clock is selected as CPU clock when STOP instruction is executed
Ring-OSC clock
High-speed system clock
Operation stopped
(17/fR)
Clock switched
by software

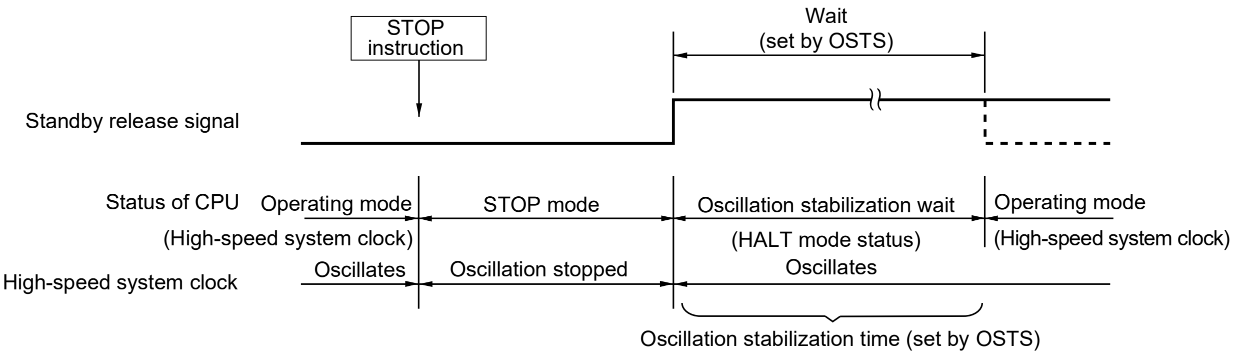
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

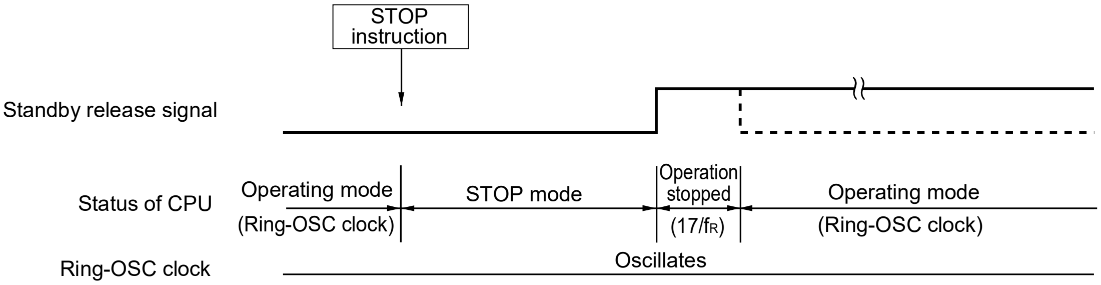
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 18-6. STOP Mode Release by Interrupt Request Generation

(1) When high-speed system clock is used as CPU clock


STOP instruction
Wait (set by OSTS)
Standby release signal
Status of CPU
Operating mode (High-speed system clock)
STOP mode
Oscillation stabilization wait (HALT mode status)
Operating mode (High-speed system clock)
High-speed system clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (set by OSTS)

(2) When Ring-OSC clock is used as CPU clock


STOP instruction
Standby release signal
Status of CPU
Operating mode (Ring-OSC clock)
STOP mode
Operation stopped (17/fR)
Operating mode (Ring-OSC clock)
Ring-OSC clock
Oscillates

Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

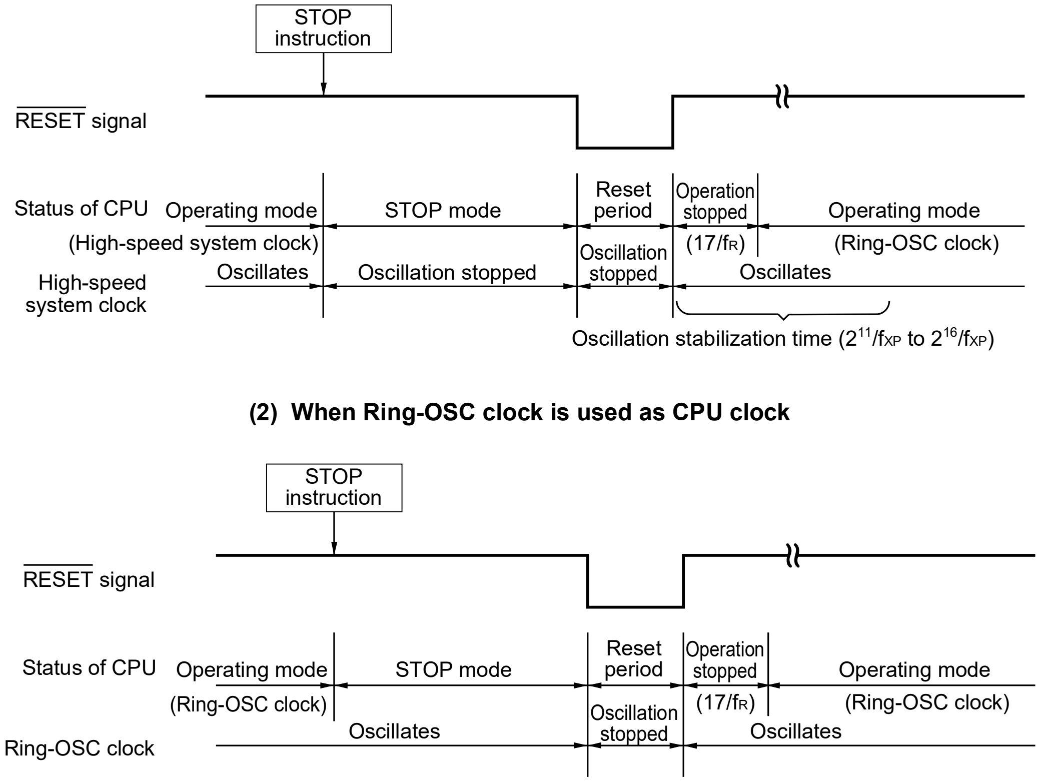
2. f_R: Ring-OSC clock oscillation frequency

(b) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

Figure 18-7. STOP Mode Release by $\overline{\text{RESET}}$ Input

(1) When high-speed system clock is used as CPU clock



(2) When Ring-OSC clock is used as CPU clock

Remarks 1. fXP: High-speed system clock oscillation frequency

2. fR: Ring-OSC clock oscillation frequency

Table 18-5. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: don't care

CHAPTER 19 RESET FUNCTION

The following five 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 clock monitor high-speed system clock oscillation stop detection
(4) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(5) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

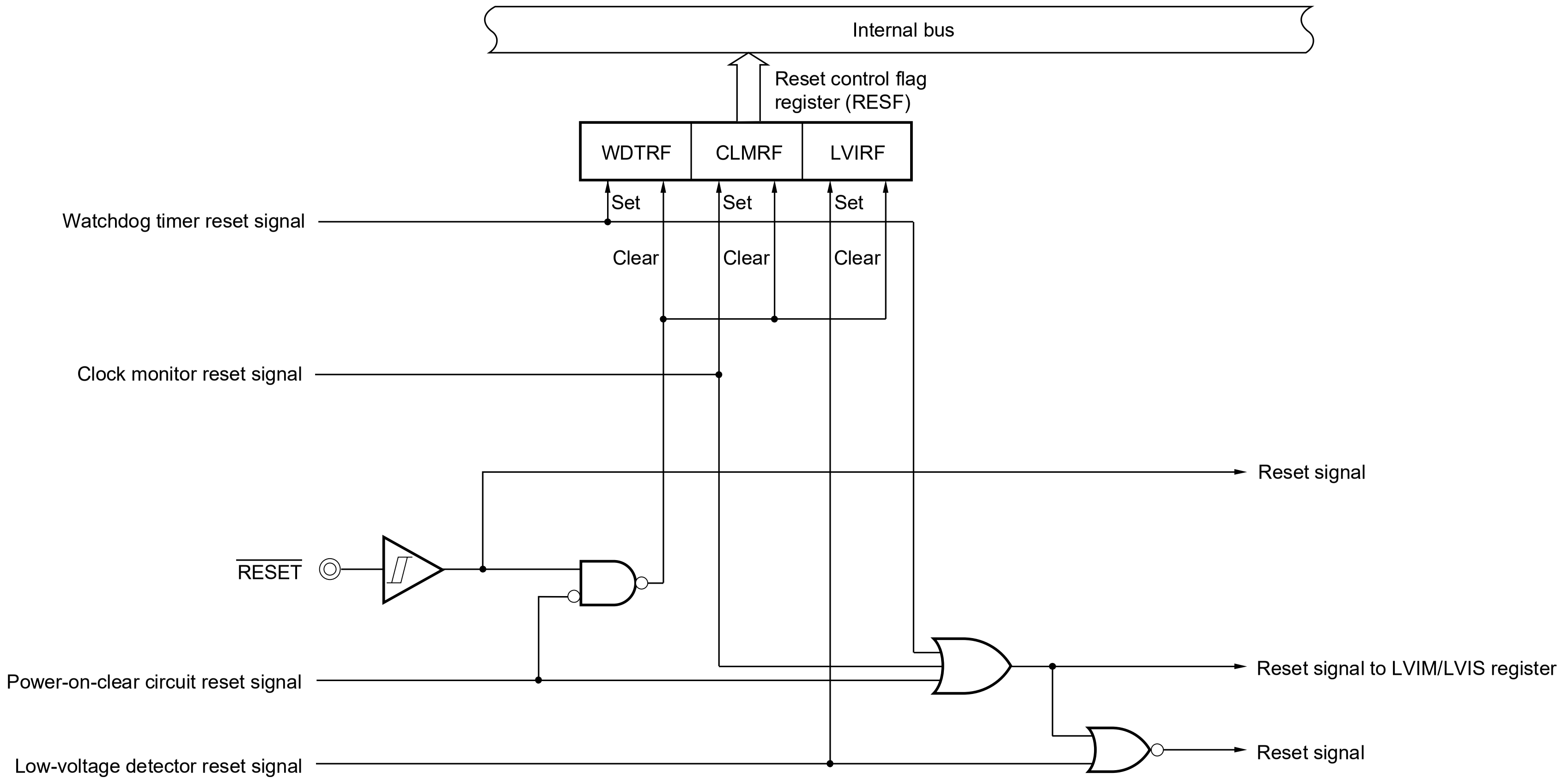
External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is input.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, high-speed system clock oscillation stop is detected by the clock monitor, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Table 19-1. Each pin is high impedance during reset input or during the oscillation stabilization time just after reset release, except for P130, which is low-level output.

When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s). A reset generated by the watchdog timer and clock monitor sources is automatically released after the reset, and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s) (refer to **Figures 19-2** to **19-4**). Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} > V_{POC}$ or $V_{DD} > V_{LVI}$ after the reset, and program execution starts using the Ring-OSC clock after the CPU clock operation has stopped for $17/f_R$ (s) (refer to **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, the high-speed system clock and Ring-OSC clock stop oscillating.
3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance, except for P130, which is set to low-level output.

Figure 19-1. Block Diagram of Reset Function

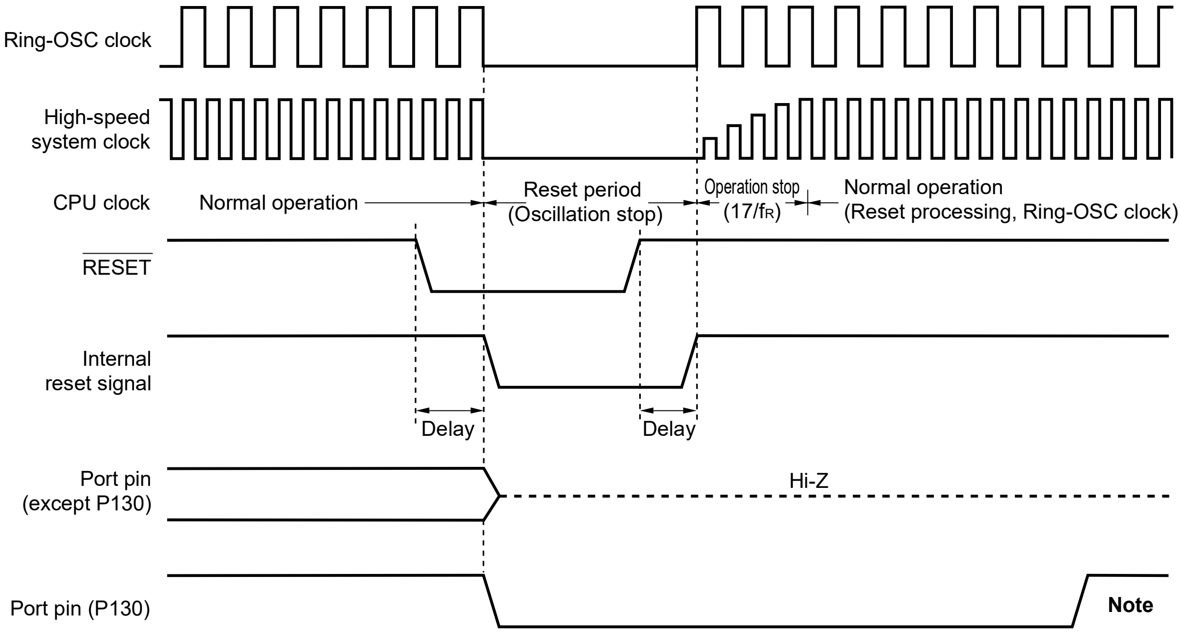

Internal bus
Reset control flag register (RESF)
WDTRF
CLMRF
LVIRF
Set
Clear
Set
Clear
Set
Clear
Watchdog timer reset signal
Clock monitor reset signal
RESET
Power-on-clear circuit reset signal
Low-voltage detector reset signal
Reset signal
Reset signal to LVIM/LVIS register
Reset signal

Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level selection register

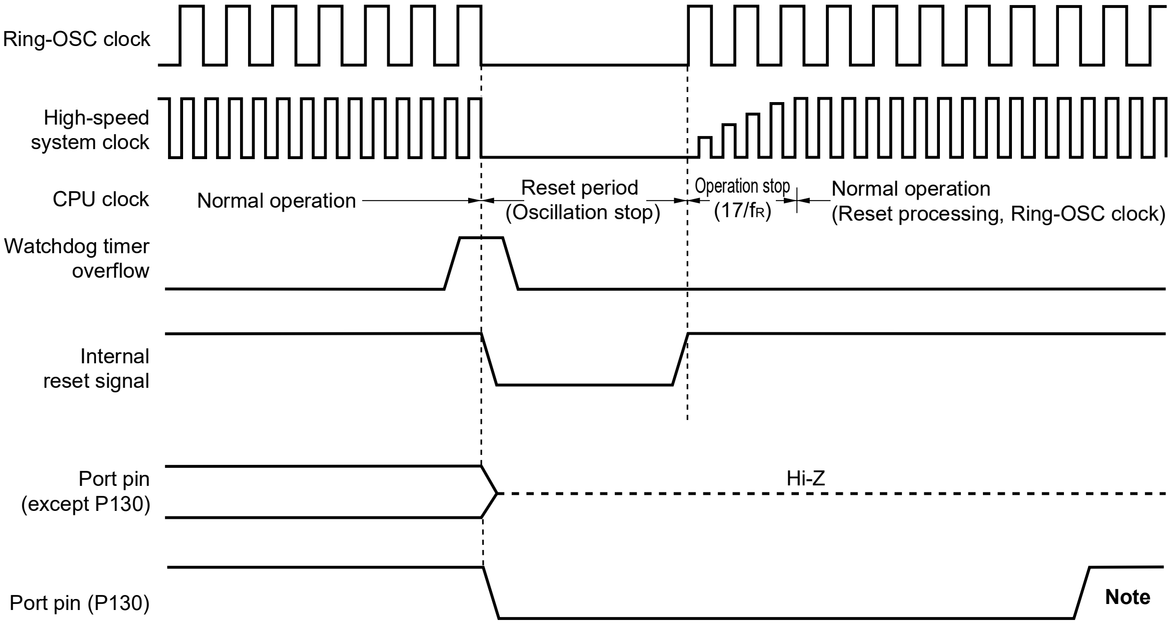
Figure 19-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the reset signal to the CPU.

Figure 19-3. Timing of Reset Due to Watchdog Timer Overflow

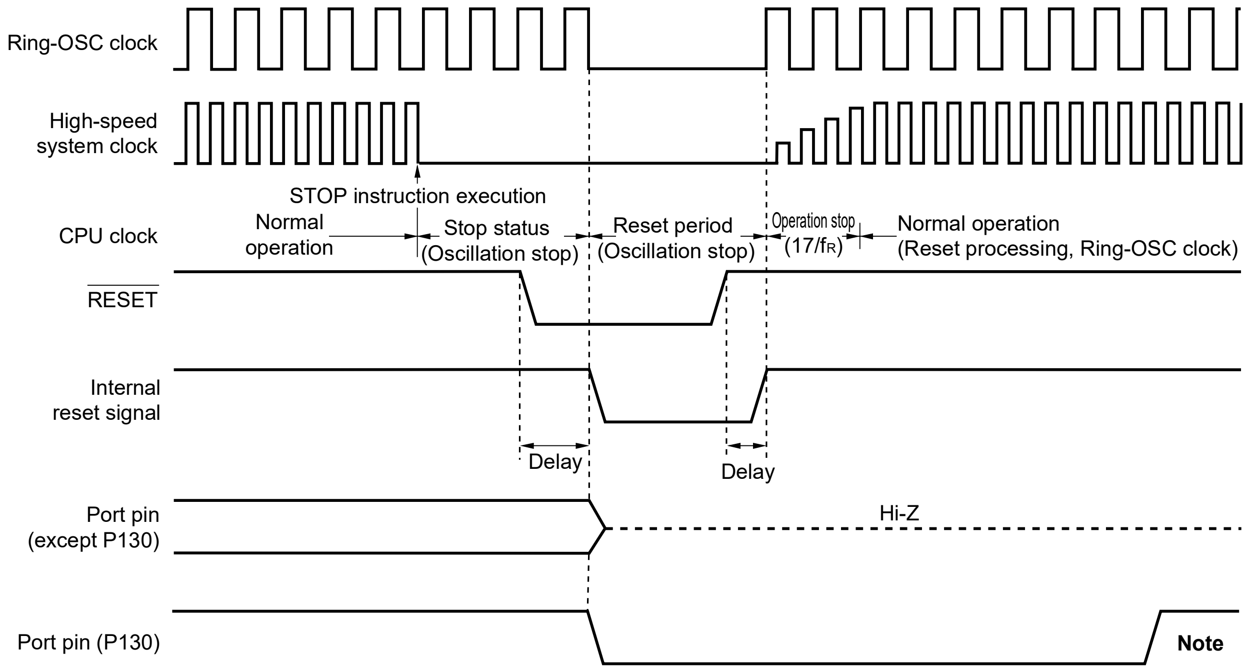


Note Set P130 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level immediately after reset is effected, the output signal of P130 can be dummy-output as the reset signal to the CPU.

Figure 19-4. Timing of Reset in STOP Mode by RESET Input



Note Set P130 to high-level output by software.

Remarks 1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level immediately after reset is effected, the output signal of P130 can be dummy-output as the reset signal to the CPU.

2. For the reset timing of the power-on-clear circuit and low-voltage detector, refer to **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**.

Table 19-1. Hardware Statuses After Reset Acknowledgment (1/2)

Hardware		Status After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P3, P6, P7, P12 to P14) (output latches)		00H (undefined only for P2)
Port mode registers (PM0, PM1, PM3, PM6, PM7, PM12, PM14)		FFH
Pull-up resistor option registers (PU0, PU1, PU3, PU7, PU12, PU14)		00H
Input switch control register (ISC)		00H
Internal memory size switching register (IMS)		CFH
Processor clock control register (PCC)		00H
Ring-OSC mode register (RCM)		00H
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		00H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
System wait control register (VSWC)		00H[Note 3]
16-bit timer/event counter 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 000, 010 (CR000, CR010)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timers H0, H1	Compare registers 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	Mode registers (TMHMD0, TMHMD1)	00H
	Carrier control register 1 (TMCYC1)[Note 4]	00H
Watch timer	Operation mode register (WTM)	00H
Clock output controller	Clock output selection register (CKS)	00H
Watchdog timer	Mode register (WDTM)	67H
	Enable register (WDTE)	9AH

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

3. Do not access VSWC in the current IE environment (IE-78K0-NS, IE-78K0-NS-A, and IE-78K0K1-ET).

4. 8-bit timer H1 only.

Table 19-1. Hardware Statuses After Reset Acknowledgment (2/2)

Hardware		Status After Reset Acknowledgment
A/D converter	Conversion result register (ADCR)	Undefined
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	Power-fail comparison mode register (PFM)	00H
	Power-fail comparison threshold register (PFT)	00H
Serial interface UART0	Receive buffer register 0 (RXB0)	FFH
	Transmit shift register 0 (TXS0)	FFH
	Asynchronous serial interface operation mode register 0 (ASIM0)	01H
	Baud rate generator control register 0 (BRGC0)	1FH
Serial interface UART6	Receive buffer register 6 (RXB6)	FFH
	Transmit buffer register 6 (TXB6)	FFH
	Asynchronous serial interface operation mode register 6 (ASIM6)	01H
	Asynchronous serial interface reception error status register 6 (ASIS6)	00H
	Asynchronous serial interface transmission status register 6 (ASIF6)	00H
	Clock selection register 6 (CKSR6)	00H
	Baud rate generator control register 6 (BRGC6)	FFH
	Asynchronous serial interface control register 6 (ASICL6)	16H
Serial interface CSI10	Transmit buffer register 10 (SOTB10)	Undefined
	Serial I/O shift register 10 (SIO10)	00H
	Serial operation mode register 10 (CSIM10)	00H
	Serial clock selection register 10 (CSIC10)	00H
Key interrupt	Key return mode register (KRM)	00H
Clock monitor	Mode register (CLM)	00H
Reset function	Reset control flag register (RESF)	00H[Note 1]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 1]
	Low-voltage detection level selection register (LVIS)	00H[Note 1]
Interrupt	Request flag registers 0L, 0H, 1L (IF0L, IF0H, IF1L)	00H
	Mask flag registers 0L, 0H, 1L (MK0L, MK0H, MK1L)	FFH
	Priority specification flag registers 0L, 0H, 1L (PR0L, PR0H, PR1L)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
Flash memory	Flash protect command register (PFCMD)	Undefined
	Flash status register (PFS)	00H
	Flash programming mode control register (FLPMC)	0XH[Note 2]

Notes 1. These values vary depending on the reset source.

Reset Source / Register	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
RESF	Refer to **Table 19-2**.				
LVIM	Cleared (00H)	Cleared (00H)	Cleared (00H)	Cleared (00H)	Held
LVIS					

2. Differs depending on the reset source.

- User mode: 08H
- On-board mode: 0CH

19.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0/KD1+. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset input by power-on-clear (POC) circuit, and reading RESF clear RESF to 00H.

Figure 19-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	CLMRF	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

CLMRF	Internal reset request by clock monitor (CLM)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 19-2.

Table 19-2. RESF Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ input	Reset by POC	Reset by WDT	Reset by CLM	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held	Held
CLMRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

CHAPTER 20 CLOCK MONITOR

20.1 Functions of Clock Monitor

The clock monitor samples the high-speed system clock using the on-chip Ring-OSC, and generates an internal reset signal when the high-speed system clock is stopped.

When a reset signal is generated by the clock monitor, bit 1 (CLMRF) of the reset control flag register (RESF) is set to 1. For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

The clock monitor automatically stops under the following conditions.

- Reset is released and during the oscillation stabilization time
- In STOP mode and during the oscillation stabilization time
- When the high-speed system clock is stopped by software (MSTOP = 1 or MCC = 1) and during the oscillation stabilization time
- When the Ring-OSC clock is stopped

Remark MSTOP: Bit 7 of main OSC control register (MOC)
MCC: Bit 7 of processor clock control register (PCC)

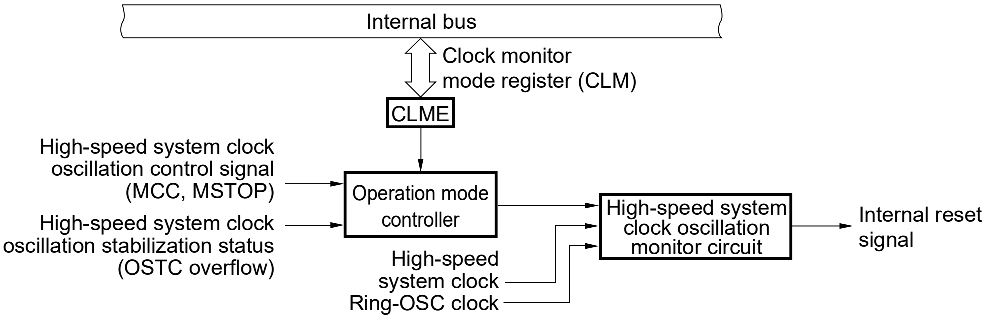
20.2 Configuration of Clock Monitor

The clock monitor includes the following hardware.

Table 20-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 20-1. Block Diagram of Clock Monitor


Internal bus
Clock monitor mode register (CLM)
CLME
High-speed system clock oscillation control signal (MCC, MSTOP)
High-speed system clock oscillation stabilization status (OSTC overflow)
Operation mode controller
High-speed system clock oscillation monitor circuit
High-speed system clock
Ring-OSC clock
Internal reset signal

Remark MCC: Bit 7 of processor clock control register (PCC)
MSTOP: Bit 7 of main OSC control register (MOC)
OSTC: Oscillation stabilization time counter status register (OSTC)

20.3 Register Controlling Clock Monitor

The clock monitor is controlled by the clock monitor mode register (CLM).

(1) Clock monitor mode register (CLM)

This register sets the operation mode of the clock monitor.

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

$\overline{\text{RESET}}$ input clears CLM to 00H.

Figure 20-2. Format of Clock Monitor Mode Register (CLM)

Address: FFA9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME

CLME	Enables/disables clock monitor operation
0	Disables clock monitor operation
1	Enables clock monitor operation

Cautions 1. Once bit 0 (CLME) is set to 1, it cannot be cleared to 0 except by $\overline{\text{RESET}}$ input or the internal reset signal.

2. If the reset signal is generated by the clock monitor, CLME is cleared to 0 and bit 1 (CLMRF) of the reset control flag register (RESF) is set to 1.

20.4 Operation of Clock Monitor

This section explains the functions of the clock monitor. The monitor start and stop conditions are as follows.

<Monitor start condition>

When bit 0 (CLME) of the clock monitor mode register (CLM) is set to operation enabled (1).

<Monitor stop condition>

- Reset is released and during the oscillation stabilization time
- In STOP mode and during the oscillation stabilization time
- When the high-speed system clock is stopped by software (MSTOP = 1 or MCC = 1) and during the oscillation stabilization time
- When the Ring-OSC clock is stopped

Remark MSTOP: Bit 7 of main OSC control register (MOC)
MCC: Bit 7 of processor clock control register (PCC)

Table 20-2. Operation Status of Clock Monitor (When CLME = 1)

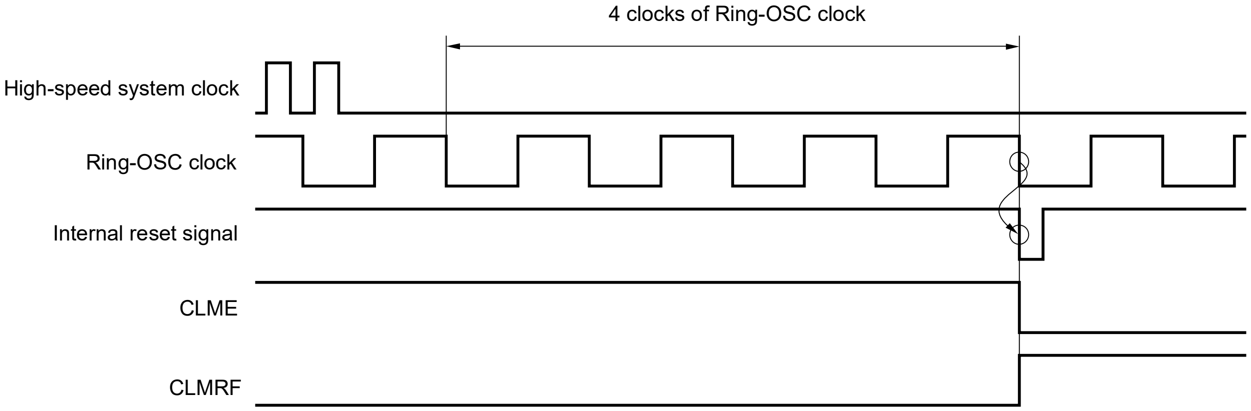
CPU Operation Clock	Operation Mode	High-Speed System Clock Status	Ring-OSC Clock Status	Clock Monitor Status
High-speed system clock	STOP mode	Stopped	Oscillating	Stopped
			Stopped[Note]	
	$\overline{\text{RESET}}$ input		Oscillating	
			Stopped[Note]	
	Normal operation mode HALT mode	Oscillating	Oscillating	Operating
			Stopped[Note]	Stopped
Ring-OSC clock	STOP mode	Stopped	Oscillating	Stopped
	$\overline{\text{RESET}}$ input			
	Normal operation mode HALT mode	Oscillating		Operating
		Stopped		Stopped

Note The Ring-OSC clock is stopped only when the “Ring-OSC can be stopped by software” is selected by the option byte. If “Ring-OSC cannot be stopped” is selected, the Ring-OSC clock cannot be stopped.

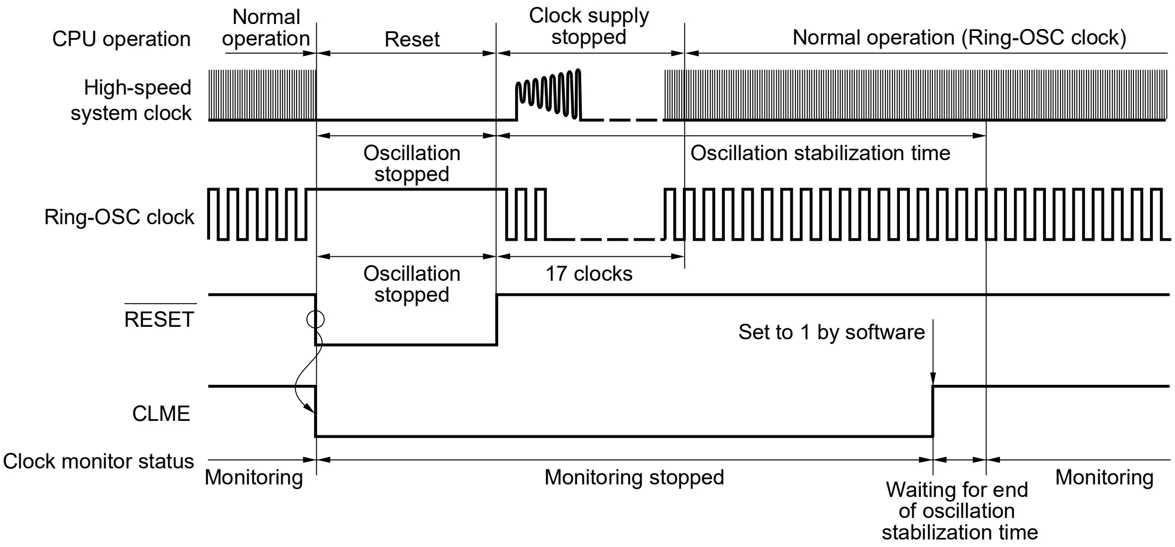
The clock monitor timing is as shown in Figure 20-3.

Figure 20-3. Timing of Clock Monitor (1/4)

(1) When internal reset is executed by oscillation stop of high-speed system clock



(2) Clock monitor status after $\overline{\text{RESET}}$ input (CLME = 1 is set after $\overline{\text{RESET}}$ input and during high-speed system clock oscillation stabilization time)

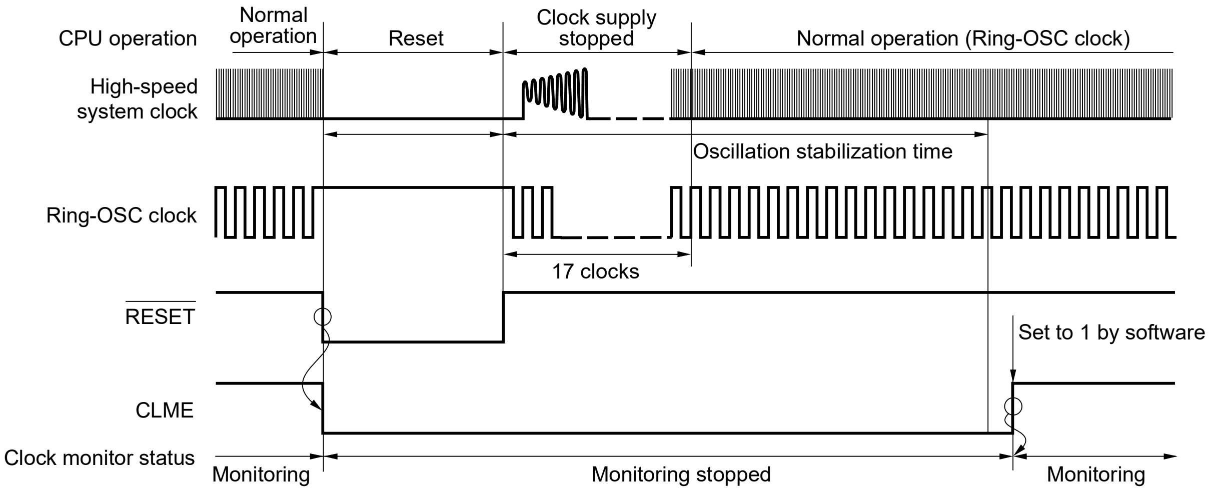


$\overline{\text{RESET}}$ input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. Even if CLME is set to 1 by software during the oscillation stabilization time (reset value of OSTS register is 05H ($2^{16}/f_{XP}$)) of the high-speed system clock, monitoring is not performed until the oscillation stabilization time of the high-speed system clock ends. Monitoring is automatically started at the end of the oscillation stabilization time.

Caution Waiting for the oscillation stabilization time is not required when the external RC oscillation clock is selected as the high-speed system clock by the option byte. Therefore, the CPU clock can be switched without reading the OSTC value. However, the clock monitor starts operation after the oscillation stabilization time (OSTS register reset value = 05H ($2^{16}/f_{XP}$)) has elapsed.

Figure 20-3. Timing of Clock Monitor (2/4)

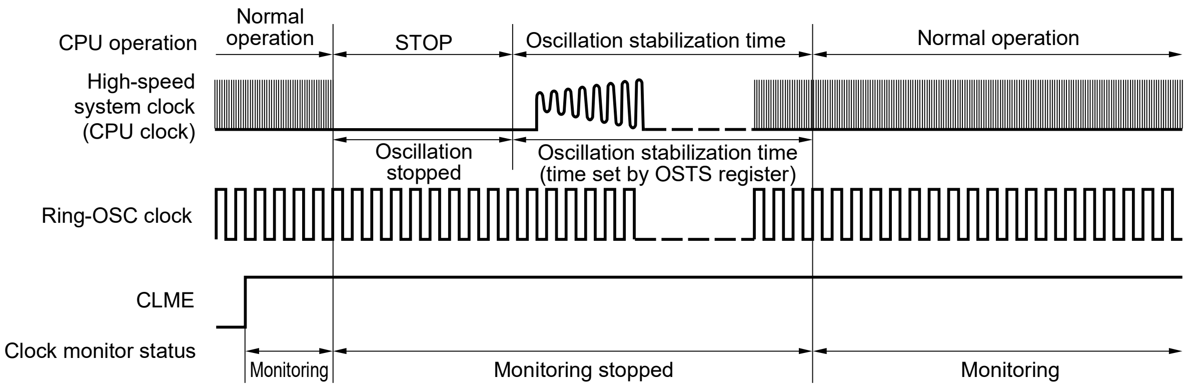
(3) Clock monitor status after $\overline{\text{RESET}}$ input
(CLME = 1 is set after $\overline{\text{RESET}}$ input and at the end of high-speed system clock oscillation stabilization time)



$\overline{\text{RESET}}$ input clears bit 0 (CLME) of the clock monitor mode register (CLM) to 0 and stops the clock monitor operation. When CLME is set to 1 by software at the end of the oscillation stabilization time (reset value of OSTS register is 05H (2^{16}/f_{XP})) of the high-speed system clock, monitoring is started.

Caution Waiting for the oscillation stabilization time is not required when the external RC oscillation clock is selected as the high-speed system clock by the option byte. Therefore, the CPU clock can be switched without reading the OSTC value. However, the clock monitor starts operation after the oscillation stabilization time (OSTS register reset value = 05H (2^{16}/f_{XP})) has elapsed.

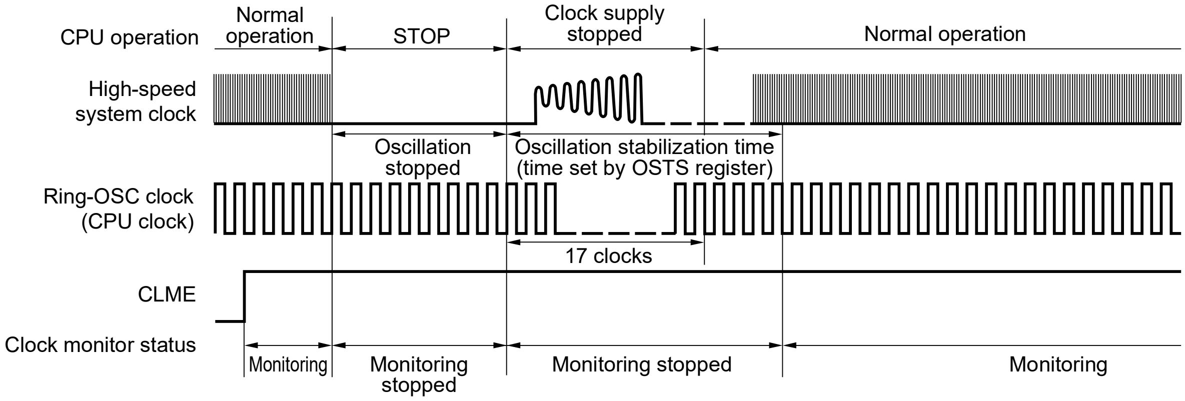
(4) Clock monitor status after STOP mode is released
(CLME = 1 is set when CPU clock operates on high-speed system clock and before entering STOP mode)



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the high-speed system clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

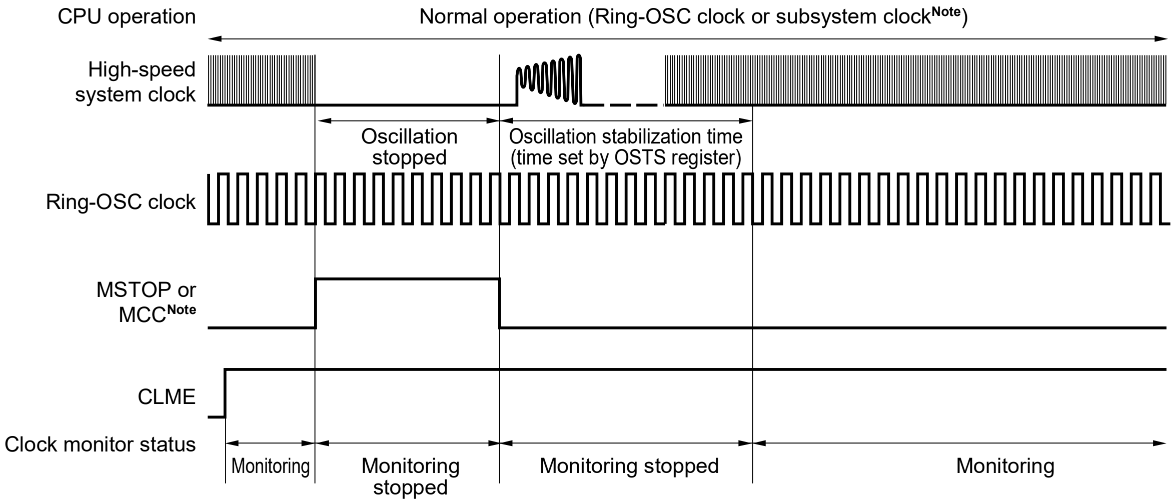
Figure 20-3. Timing of Clock Monitor (3/4)

(5) Clock monitor status after STOP mode is released (CLME = 1 is set when CPU clock operates on Ring-OSC clock and before entering STOP mode)



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before entering STOP mode, monitoring automatically starts at the end of the high-speed system clock oscillation stabilization time. Monitoring is stopped in STOP mode and during the oscillation stabilization time.

(6) Clock monitor status after high-speed system clock oscillation is stopped by software



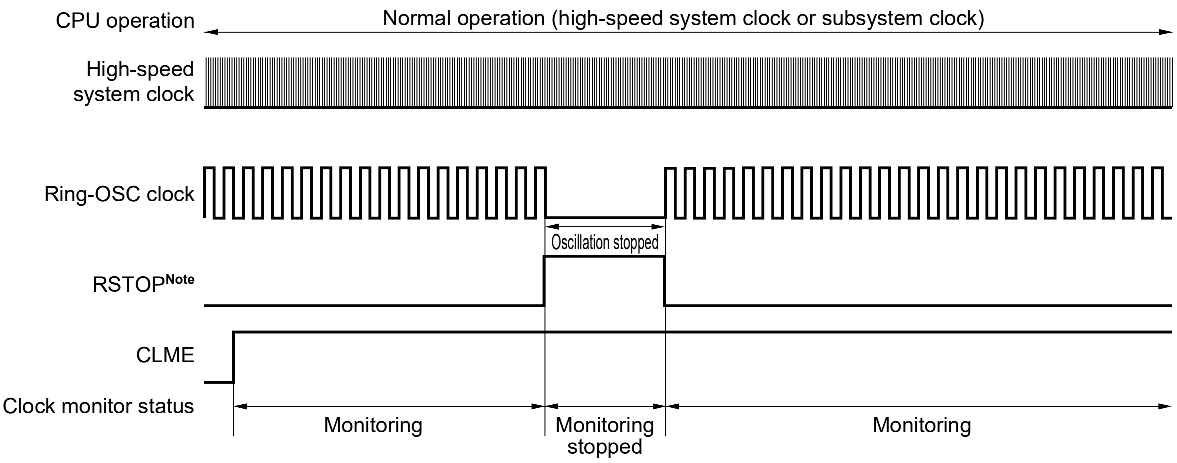
When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before or while oscillation of the high-speed system clock is stopped, monitoring automatically starts at the end of the high-speed system clock oscillation stabilization time. Monitoring is stopped when oscillation of the high-speed system clock is stopped and during the oscillation stabilization time.

Note The register that controls oscillation of the high-speed system clock differs depending on the type of the clock supplied to the CPU.

- When CPU operates on Ring-OSC clock: Controlled by bit 7 (MSTOP) of the main OSC control register (MOC)
- When CPU operates on subsystem clock: Controlled by bit 7 (MCC) of the processor clock control register (PCC)

Figure 20-3. Timing of Clock Monitor (4/4)

(7) Clock monitor status after Ring-OSC clock oscillation is stopped by software



When bit 0 (CLME) of the clock monitor mode register (CLM) is set to 1 before or while oscillation of the Ring-OSC clock is stopped, monitoring automatically starts after the Ring-OSC clock is stopped. Monitoring is stopped when oscillation of the Ring-OSC clock is stopped.

Note If it is specified by the option byte that Ring-OSC cannot be stopped, the setting of bit 0 (RSTOP) of the Ring-OSC mode register (RCM) is invalid. To set RSTOP, be sure to confirm that bit 1 (MCS) of the main clock mode register (MCM) is 1.

CHAPTER 21 POWER-ON-CLEAR CIRCUIT

21.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 2.1 V ±0.1 V[Note]), and generates internal reset signal when $V_{DD} < V_{POC}$.

Note This value may change after evaluation.

Cautions 1. If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

2. The supply voltage is V_{DD} = 2.0 to 5.5 V when the Ring-OSC clock or subsystem clock is used, but be sure to use the product in a voltage range of 2.2 to 5.5 V because the detection voltage (V_{POC}) of the POC circuit is 2.1 V ±0.1 V.

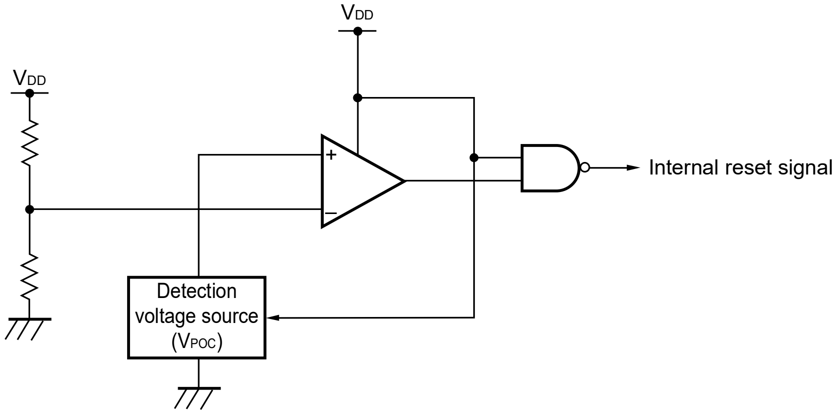
Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detection (LVI) circuit, or clock monitor. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI, or the clock monitor.

For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

21.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 21-1.

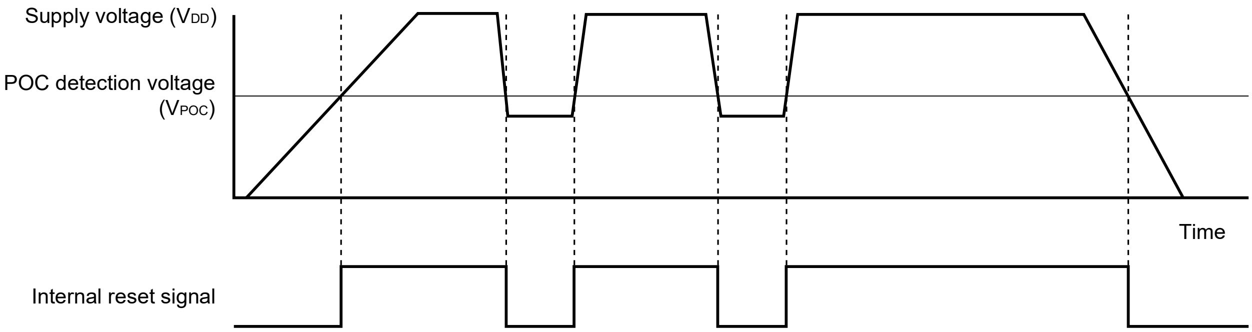
Figure 21-1. Block Diagram of Power-on-Clear Circuit



21.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage (V_{DD}) and detection voltage (V_{POC}) are compared, and when $V_{DD} < V_{POC}$, an internal reset signal is generated.

Figure 21-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



21.4 Cautions for Power-on-Clear Circuit

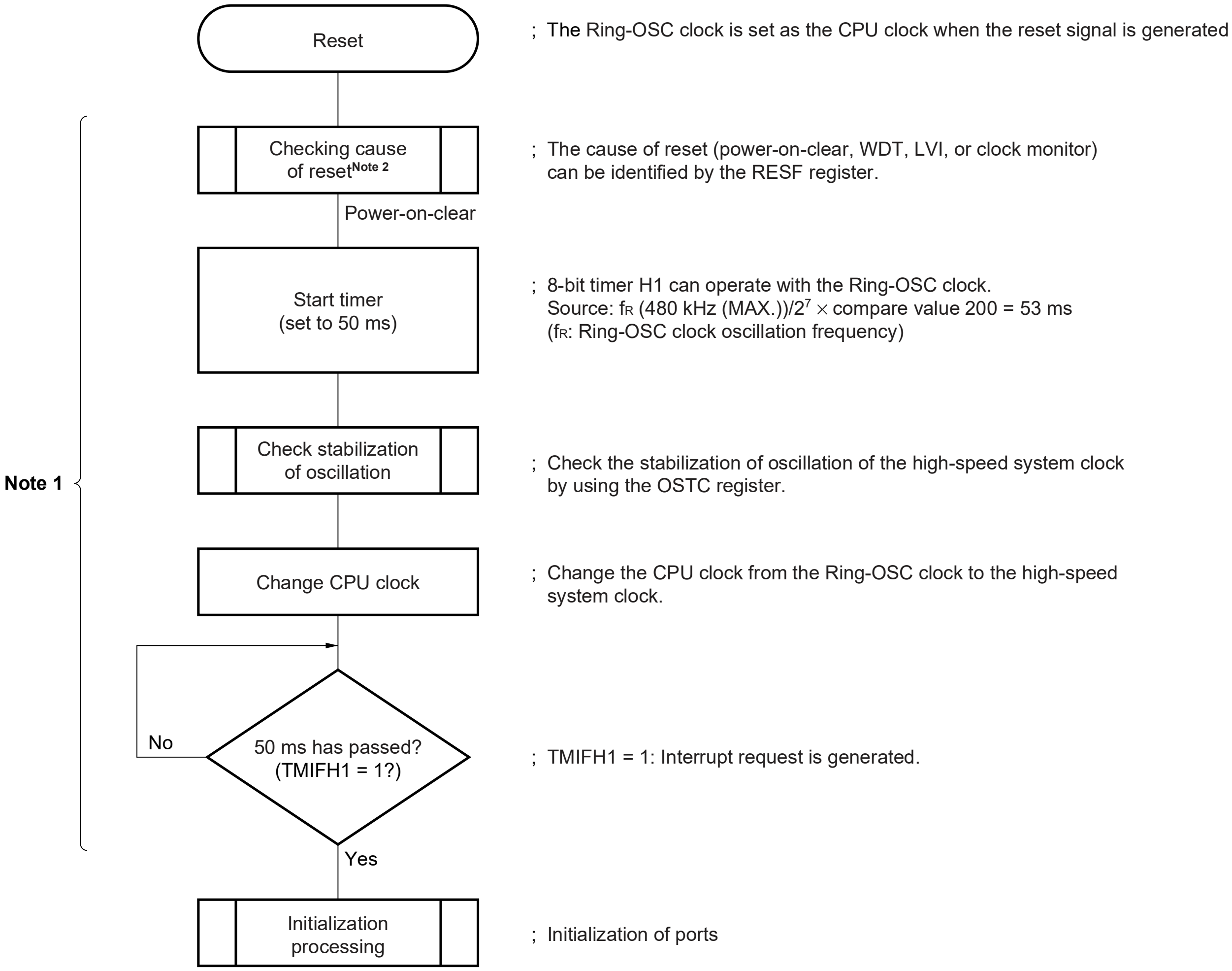
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 21-3. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

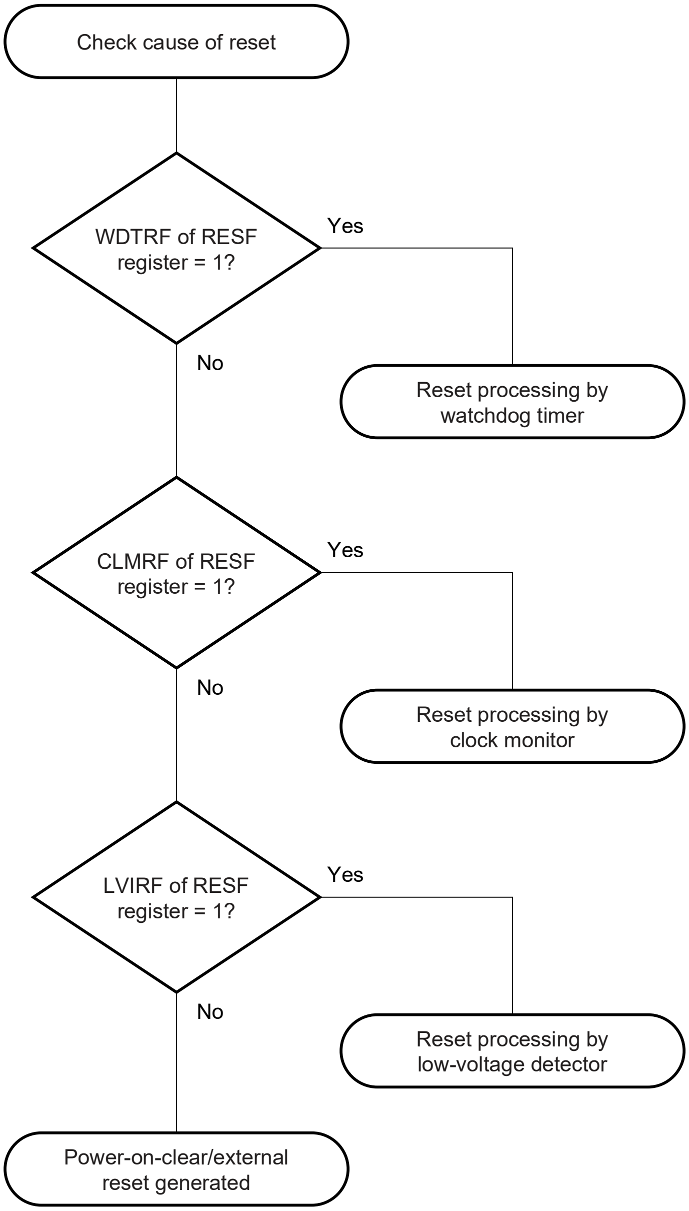

Reset
; The Ring-OSC clock is set as the CPU clock when the reset signal is generated
Checking cause of reset[Note 2]
; The cause of reset (power-on-clear, WDT, LVI, or clock monitor) can be identified by the RESF register.
Power-on-clear
Start timer (set to 50 ms)
; 8-bit timer H1 can operate with the Ring-OSC clock.
Source: fR (480 kHz (MAX.))/2^7 × compare value 200 = 53 ms
(fR: Ring-OSC clock oscillation frequency)
Note 1
Check stabilization of oscillation
; Check the stabilization of oscillation of the high-speed system clock by using the OSTC register.
Change CPU clock
; Change the CPU clock from the Ring-OSC clock to the high-speed system clock.
No
50 ms has passed? (TMIFH1 = 1?)
; TMIFH1 = 1: Interrupt request is generated.
Yes
Initialization processing
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 21-3. Example of Software Processing After Release of Reset (2/2)

- Checking cause of reset


Check cause of reset
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
CLMRF of RESF register = 1?
Yes
No
Reset processing by clock monitor
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

CHAPTER 22 LOW-VOLTAGE DETECTOR

22.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

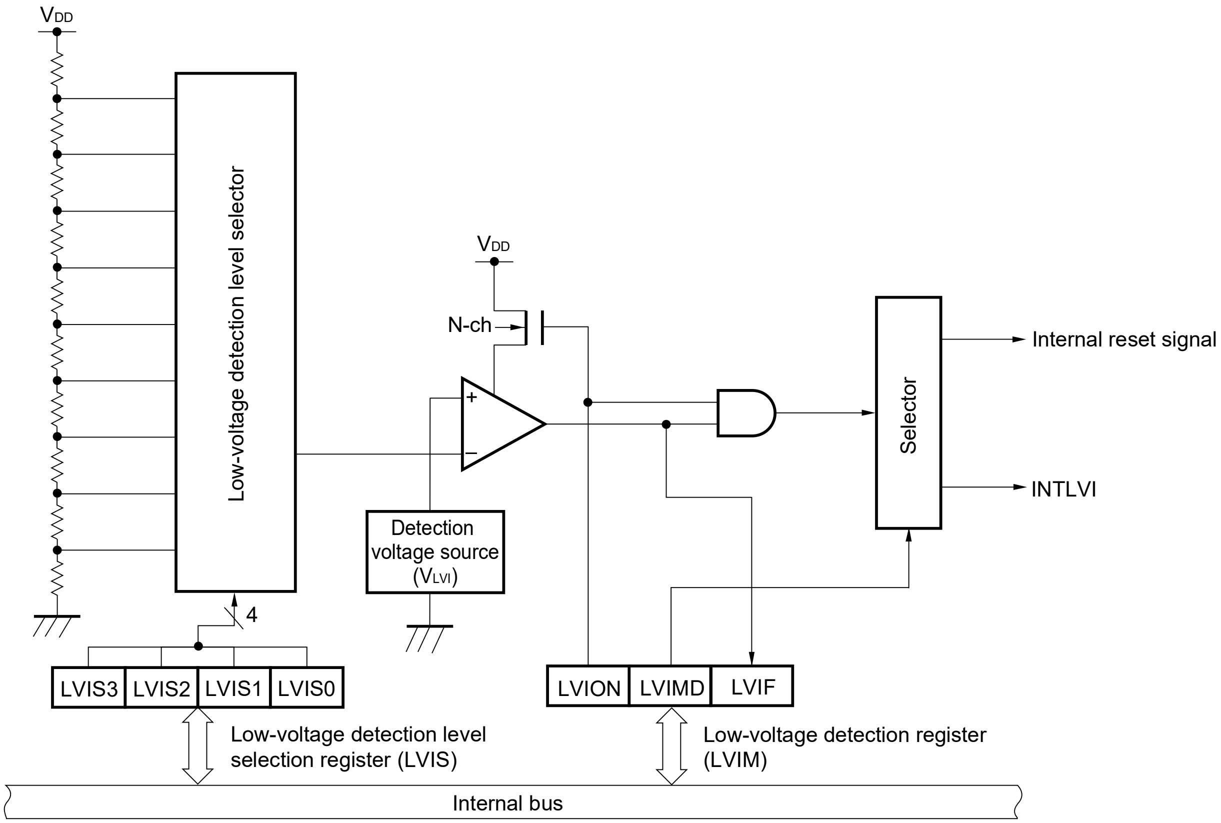
- Compares supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- Detection levels (nine levels) of supply voltage can be changed by software.
- Interrupt or reset function can be selected by software.
- Operable in STOP mode.

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

22.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown below.

Figure 22-1. Block Diagram of Low-Voltage Detector


VDD
Low-voltage detection level selector
VDD
N-ch
+
−
Detection voltage source (VLVI)
Selector
Internal reset signal
INTLVI
4
LVIS3
LVIS2
LVIS1
LVIS0
LVION
LVIMD
LVIF
Low-voltage detection level selection register (LVIS)
Low-voltage detection register (LVIM)
Internal bus

22.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level selection register (LVIS)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

$\overline{\text{RESET}}$ input clears LVIM to 00H.

Figure 22-2. Format of Low-Voltage Detection Register (LVIM)

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

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0[Note 2]	0	0	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIMD[Note 3]	Low-voltage detection operation mode selection
0	Generates interrupt signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})
1	Generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})

LVIF[Note 5]	Low-voltage detection flag
0	Supply voltage (V_{DD}) > detection voltage (V_{LVI}), or when operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVI})

Notes 1. Bit 0 is read-only.

2. Bit 4 may be 0 or 1. This bit corresponds to the LVIE bit in the 78K0/KD1.

3. LVION and LVIMD are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to instigate a wait of at least 0.2 ms from when LVION is set to 1 until the voltage is confirmed at LVIF.

5. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Caution To stop LVI, follow either of the procedures below.

- **When using 8-bit manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

(2) Low-voltage detection level selection register (LVIS)

This register selects the low-voltage detection level.

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

$\overline{\text{RESET}}$ input clears LVIS to 00H.

Figure 22-3. Format of Low-Voltage Detection Level Selection Register (LVIS)

Address: FFBFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level[Note]
0	0	0	0	V_{LVI0} (4.3 V ±0.2 V)
0	0	0	1	V_{LVI1} (4.1 V ±0.2 V)
0	0	1	0	V_{LVI2} (3.9 V ±0.2 V)
0	0	1	1	V_{LVI3} (3.7 V ±0.2 V)
0	1	0	0	V_{LVI4} (3.5 V ±0.2 V)
0	1	0	1	V_{LVI5} (3.3 V ±0.15 V)
0	1	1	0	V_{LVI6} (3.1 V ±0.15 V)
0	1	1	1	V_{LVI7} (2.85 V ±0.15 V)
1	0	0	0	V_{LVI8} (2.6 V ±0.1 V)
1	0	0	1	V_{LVI9} (2.35 V ±0.1 V)
Other than above				Setting prohibited

Note Do not set V_{LVI7}, V_{LVI8}, or V_{LVI9} when using the 78K0/KD1+ to evaluate the program of a mask ROM version of the 78K0/KD1.

Caution Be sure to clear bits 4 to 7 to 0.

22.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

- Used as reset
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal reset signal when $V_{DD} < V_{LVI}$.
- Used as interrupt
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an interrupt signal (INTLVI) when $V_{DD} < V_{LVI}$.

The operation is set as follows.

(1) When used as reset

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to instigate a wait of at least 0.2 ms.

<5> Confirm that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” at bit 0 (LVIF) of LVIM.

<6> Set bit 1 (LVIMD) of LVIM to 1 (generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})).

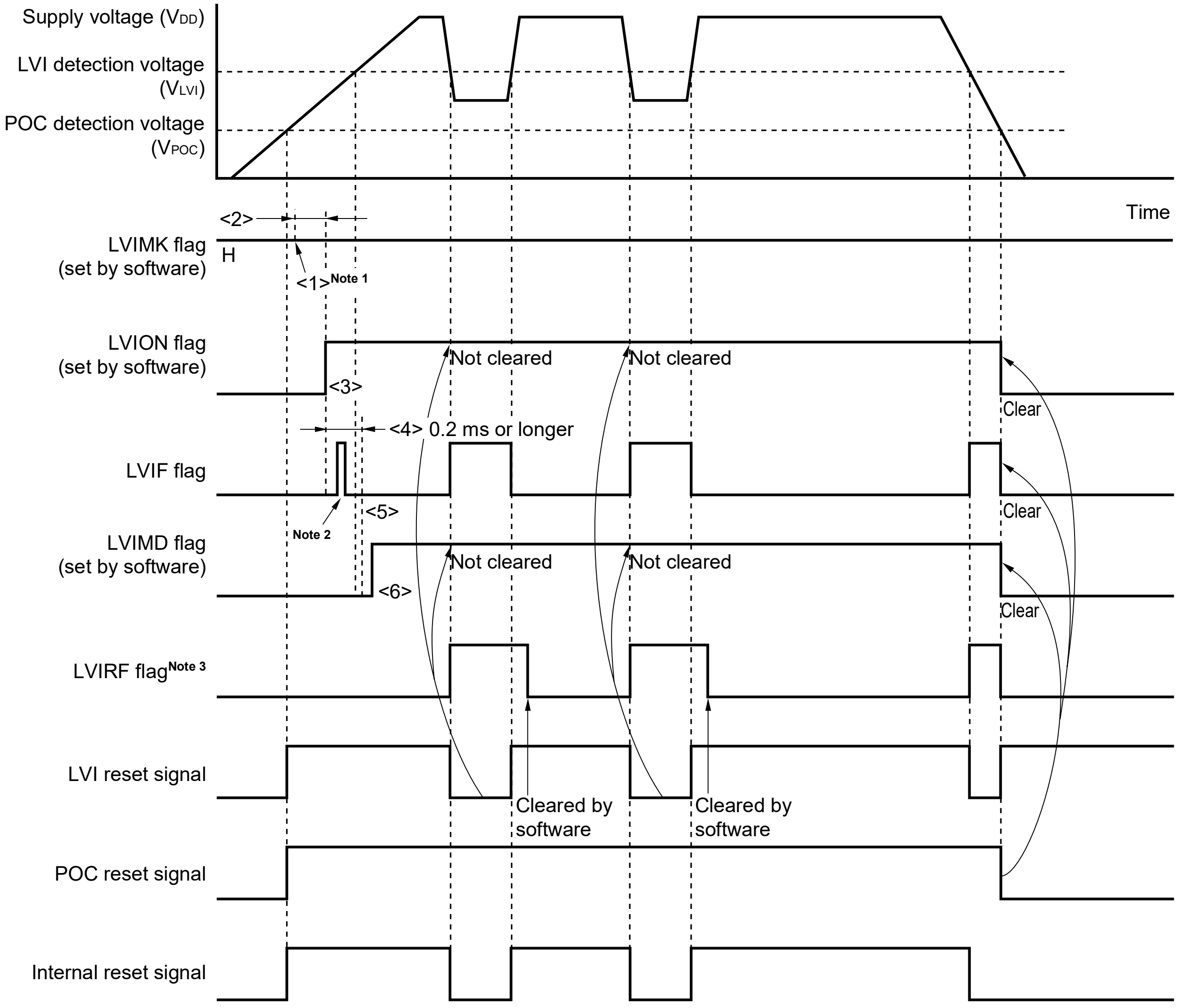
Figure 22-4 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

2. If supply voltage (V_{DD}) > detection voltage (V_{LVI}) when LVIM is set to 1, an internal reset signal is not generated.

- When stopping operation
  Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 first, and then clear LVION to 0.

Figure 22-4. Timing of Low-Voltage Detector Internal Reset Signal Generation



Notes 1. The LVIMK flag is set to “1” by $\overline{\text{RESET}}$ input.

2. The LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, refer to **CHAPTER 19 RESET FUNCTION**.

Remark <1> to <6> in Figure 22-4 above correspond to <1> to <6> in the description of “when starting operation” in **22.4 (1) When used as reset**.

(2) When used as interrupt

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to instigate a wait of at least 0.2 ms.

<5> Confirm that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” at bit 0 (LVIF) of LVIM.

<6> Clear the interrupt request flag of LVI (LVIIF) to 0.

<7> Release the interrupt mask flag of LVI (LVIMK).

<8> Execute the EI instruction (when vectored interrupts are used).

Figure 22-5 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

- When stopping operation

  Either of the following procedures must be executed.

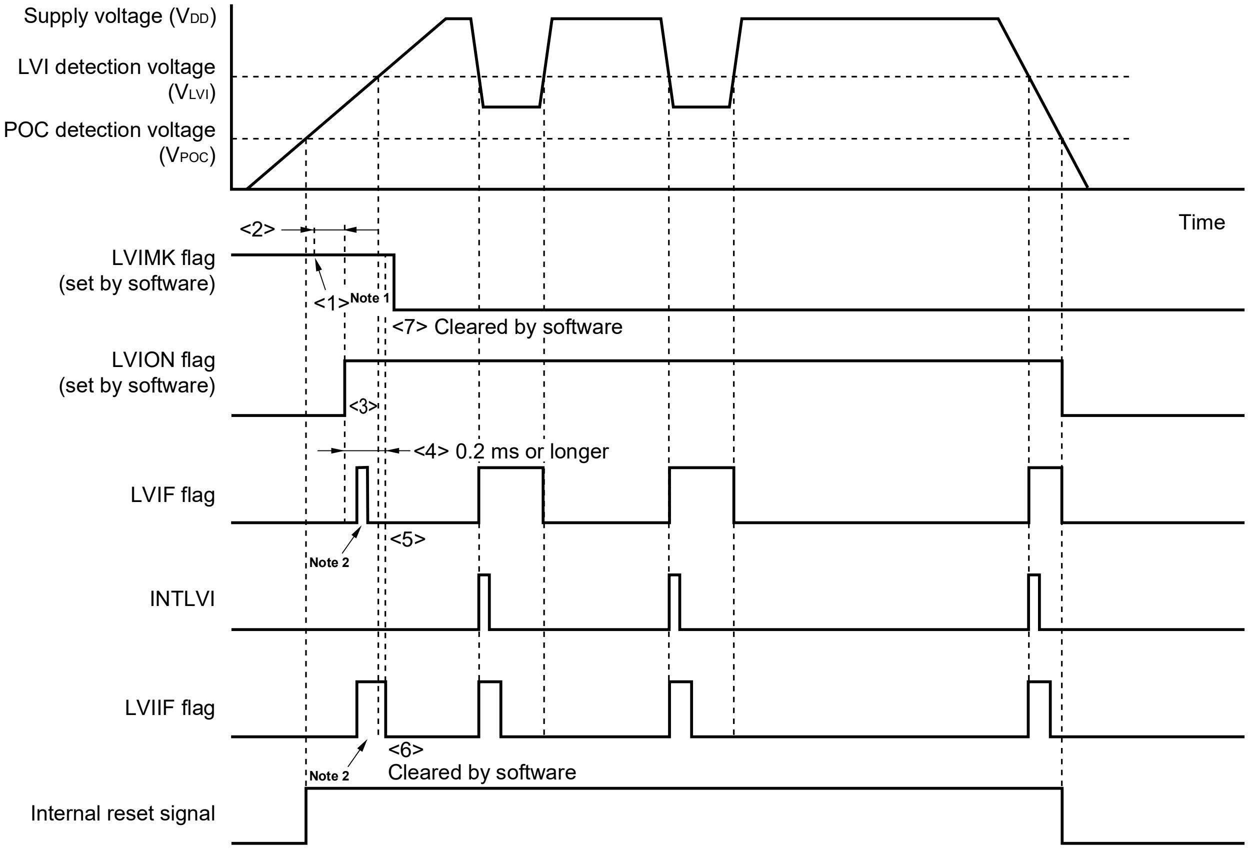
  - When using 8-bit memory manipulation instruction:

    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:

    Clear LVION to 0.

Figure 22-5. Timing of Low-Voltage Detector Interrupt Signal Generation


Supply voltage (V_{DD})
LVI detection voltage (V_{LVI})
POC detection voltage (V_{POC})
Time
<2>
LVIMK flag (set by software)
<1>Note 1
<7> Cleared by software
LVION flag (set by software)
<3>
<4> 0.2 ms or longer
LVIF flag
Note 2
<5>
INTLVI
LVIIF flag
<6>
Cleared by software
Note 2
Internal reset signal

Notes 1. The LVIMK flag is set to “1” by $\overline{\text{RESET}}$ input.

2. The LVIF and LVIIF flags may be set (1).

Remark <1> to <7> in Figure 22-5 above correspond to <1> to <7> in the description of “when starting operation” in **22.4 (2) When used as interrupt**.

22.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

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 action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take action (2) below.

In this system, take the following actions.

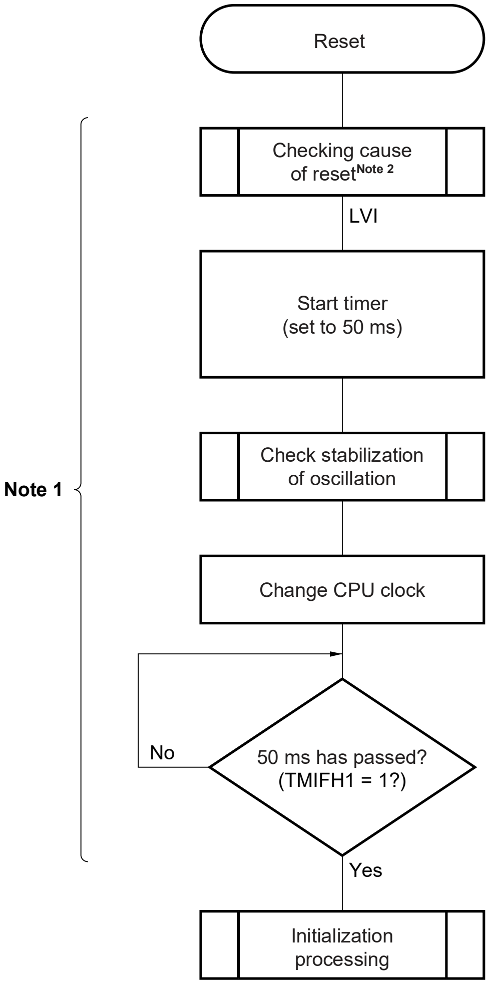
<Action>

(1) When used as reset

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 22-6. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage


Reset
Checking cause of reset[Note 2]
LVI
Start timer (set to 50 ms)
Check stabilization of oscillation
Change CPU clock
No
50 ms has passed? (TMIFH1 = 1?)
Yes
Initialization processing
Note 1

; The Ring-OSC clock is set as the CPU clock when the reset signal is generated

; The cause of reset (power-on-clear, WDT, LVI, or clock monitor) can be identified by the RESF register.

; 8-bit timer H1 can operate with the Ring-OSC clock.
Source: f_R (480 kHz (MAX.))/2^7 × compare value 200 = 53 ms
(f_R: Ring-OSC clock oscillation frequency)

; Check the stabilization of oscillation of the high-speed system clock by using the OSTC register.

; Change the CPU clock from the Ring-OSC clock to the high-speed system clock.

; TMIFH1 = 1: Interrupt request is generated.

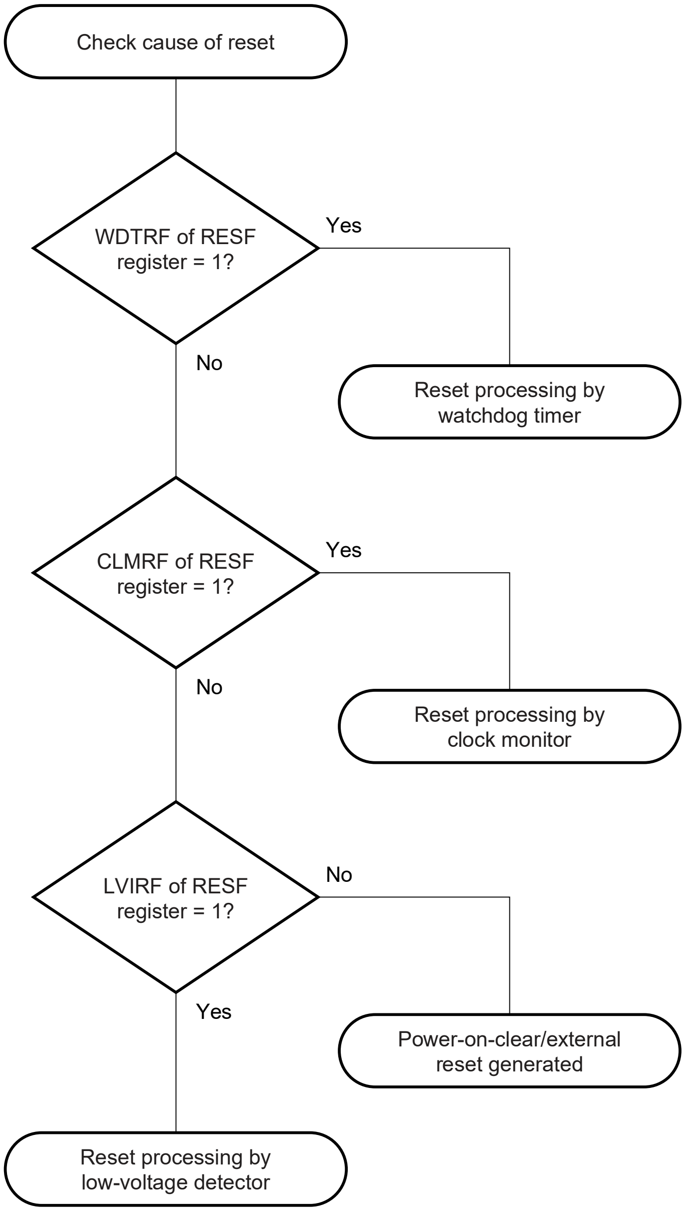
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 22-6. Example of Software Processing After Release of Reset (2/2)

- Checking cause of reset


Check cause of reset
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
CLMRF of RESF register = 1?
Yes
No
Reset processing by clock monitor
LVIRF of RESF register = 1?
No
Yes
Power-on-clear/external reset generated
Reset processing by low-voltage detector

(2) When used as interrupt

Check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 0 (LVIIF) of interrupt request flag register 0L (IF0L) to 0 and enable interrupts (EI).

In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” using the LVIF flag, and then enable interrupts (EI).

CHAPTER 23 OPTION BYTE

The 78K0/KD1+ has an area called an option byte at address 0080H of the flash memory. When using the product, be sure to set the following functions by using the option byte.

○ **Ring-OSC oscillation**

- Cannot be stopped.
- Can be stopped by software.

Figure 23-1. Allocation of Option Byte

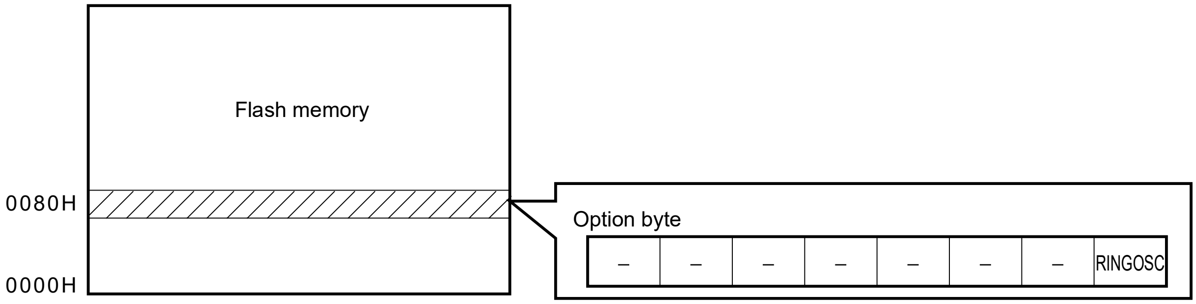


Figure 23-2. Format of Option Byte

Address: 0080H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	RINGOSC

RINGOSC	Ring-OSC oscillation
0	Can be stopped by software
1	Cannot be stopped

Cautions 1. To use the boot swap function, be sure to store the option data in the boot cluster 1 (for the boot swap function, refer to 24.8 Boot Swap Function).

2. Be sure to clear bits 1 to 7 to 0.

Remark An example of software coding for setting the option bytes is shown below.

```
OPT       CSEG      AT 0080H
OPTION:   DB        01H       ; Set to option byte (Ring-OSC cannot be stopped)
```

CHAPTER 24 FLASH MEMORY

The μPD78F0122H, 78F0123H, and 78F0124H/HD replace the internal mask ROM of the μPD780122, 780123, and 780124 of the 78K0/KD1 respectively with flash memory to which a program can be written, erased, and overwritten while mounted on the board. Table 24-1 lists the differences between the 78K0/KD1+ and the 78K0/KD1.

Table 24-1. Differences Between 78K0/KD1+ and 78K0/KD1

Item	78K0/KD1+	78K0/KD1	
	μPD78F0122H, 78F0123H, 78F0124H, 78F0124HD	μPD78F0124	μPD780121, 780122, 780123, 780124
Internal ROM configuration	Flash memory (single power supply)	Flash memory (two power supplies)	Mask ROM
Internal ROM capacity	μPD78F0122H: 16 KB[Note 1] μPD78F0123H: 24 KB[Note 1] μPD78F0124H: 32 KB[Note 1] μPD78F0124HD: 32 KB[Note 1]	μPD78F0124: 32 KB[Note 1]	μPD780121: 8 KB μPD780122: 16 KB μPD780123: 24 KB μPD780124: 32 KB
Internal high-speed RAM capacity	μPD78F0122H: 512 bytes[Note 1] μPD78F0123H: 1024 bytes[Note 1] μPD78F0124H: 1024 bytes[Note 1] μPD78F0124HD: 1024 bytes[Note 1]	μPD78F0124: 1024 bytes[Note 1]	μPD780121: 512 bytes μPD780122: 512 bytes μPD780123: 1024 bytes μPD780124: 1024 bytes
Pin 3	FLMD0 pin	V_{PP} pin	IC pin
Pin 19	P17/TI50/TO50/FLMD1 pin	P17/TI50/TO50 pin	
Power-on-clear (POC) function	Detection voltage is fixed (V_{POC} = 2.1 V ±0.1 V[Note 2])	Enabling use of POC and detection voltage selectable by product	Enabling use of POC and detection voltage selectable by mask option
Regulator	None	Available[Note 3]	
Self-programming function	Available	None	–
On-chip debug function	Available only in μPD78F0124HD	None	–
Electrical specifications	Refer to the electrical specifications chapter in the user's manual of each product.		

Notes 1. The same capacity as the mask ROM versions can be specified by means of the internal memory size switching register (IMS).

2. This value may change after evaluation.

3. The regulator cannot be used in (A1) grade products and (A2) grade products.

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When pre-producing an application set with the flash memory version and then mass-producing it with the mask ROM version, be sure to conduct sufficient evaluations for the commercial samples (not engineering samples) of the mask ROM versions.

24.1 Internal Memory Size Switching Register

The internal memory capacity can be selected using the internal memory size switching register (IMS).

IMS can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IMS to CFH.

Caution Be sure to set each product to the values shown in Table 24-2 at initialization. Also, when using the 78K0/KD1+ to evaluate the program of a mask ROM version of the 78K0/KD1, be sure to set the values shown in Table 24-2.

Figure 24-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
0	1	0	512 bytes
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
0	0	1	0	8 KB
0	1	0	0	16 KB
0	1	1	0	24 KB
1	0	0	0	32 KB
Other than above				Setting prohibited

The IMS settings required to obtain the same memory map as mask ROM versions of the 78K0/KD1 are shown in Table 24-2.

Table 24-2. Internal Memory Size Switching Register Settings

Flash Memory Version (78K0/KD1+)	Target Mask ROM Version (78K0/KD1)	IMS Setting
−	μPD780121	42H
μPD78F0122H	μPD780122	44H
μPD78F0123H	μPD780123	C6H
μPD78F0124H, 78F0124HD	μPD780124	C8H

24.2 Writing with Flash Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the 78K0/KD1+ has been mounted on the target system. The connectors that connect the dedicated flash 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 (FA series) before the 78K0/KD1+ is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 24-3. Wiring Between 78K0/KD1+ and Dedicated Flash Programmer

Pin Configuration of Dedicated Flash Programmer			With CSI10		With CSI10 + HS		With UART6	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SO10/P12	30	SO10/P12	30	TxD6/P13	29
SO/TxD	Output	Transmit signal	SI10/RxD0/P11	31	SI10/RxD0/P11	31	RxD6/P14	28
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/TxD0/P10	32	$\overline{\text{SCK10}}$/TxD0/P10	32	Not needed	Not needed
CLK	Output	Clock to 78K0/KD1+	X1	7	X1	7	X1	7
			X2[Note]	8	X2[Note]	8	X2[Note]	8
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9
FLMD0	Output	Mode signal	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	Output	Mode signal	FLMD1/TI50/TO50/P17	19	FLMD1/TI50/TO50/P17	19	FLMD1/TI50/TO50/P17	19
H/S	Input	Handshake signal	Not needed	Not needed	HS/P15/TOH0	21	Not needed	Not needed
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}	4	V_{DD}	4	V_{DD}	4
			EV_{DD}	27	EV_{DD}	27	EV_{DD}	27
			AV_{REF}	1	AV_{REF}	1	AV_{REF}	1
GND	–	Ground	V_{SS}	6	V_{SS}	6	V_{SS}	6
			EV_{SS}	26	EV_{SS}	26	EV_{SS}	26
			AV_{SS}	2	AV_{SS}	2	AV_{SS}	2

Note When using the clock out of the flash programmer, connect CLK of the programmer to X1, and connect its inverse signal to X2.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 24-2. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

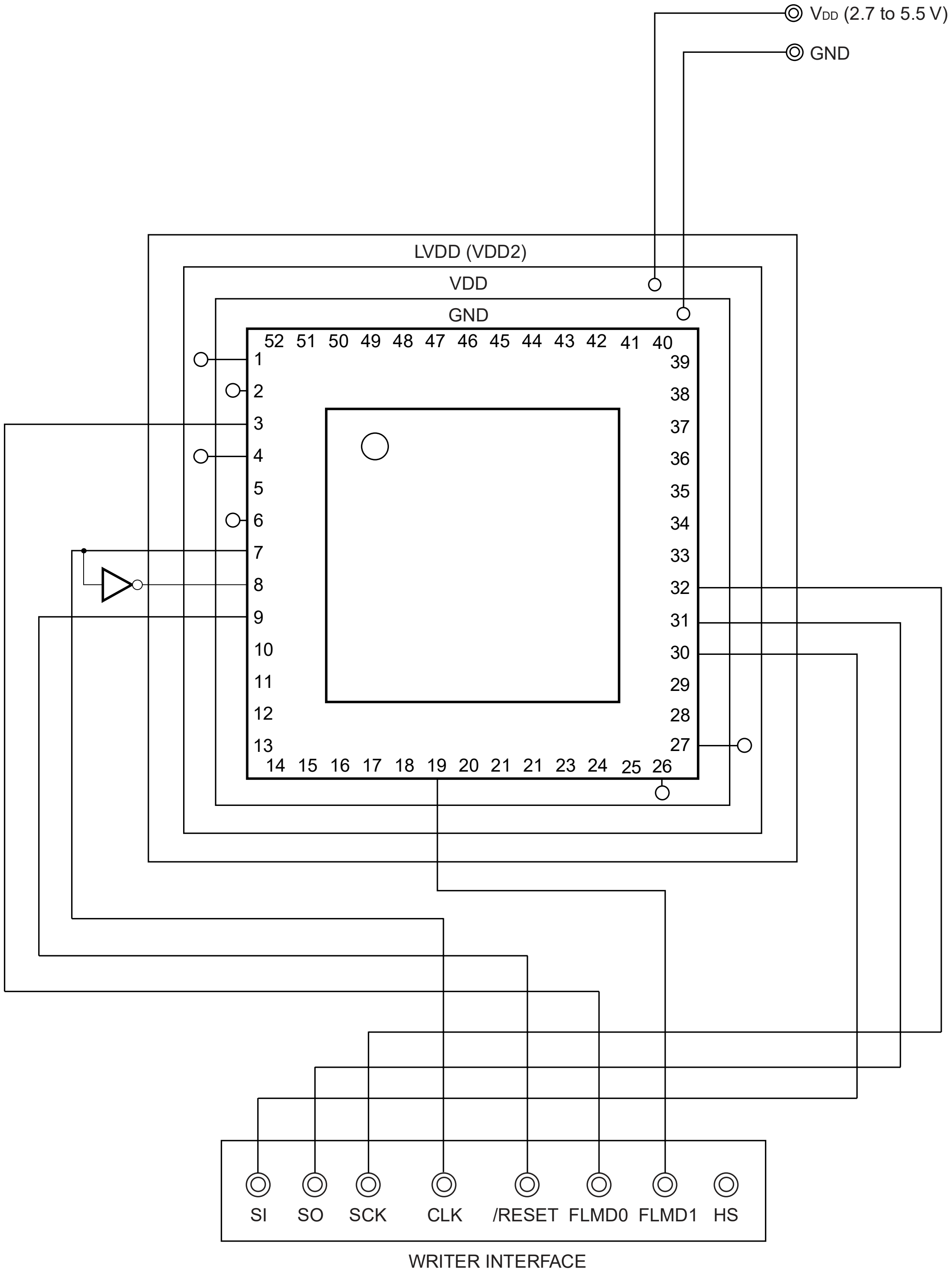


Figure 24-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10 + HS) Mode

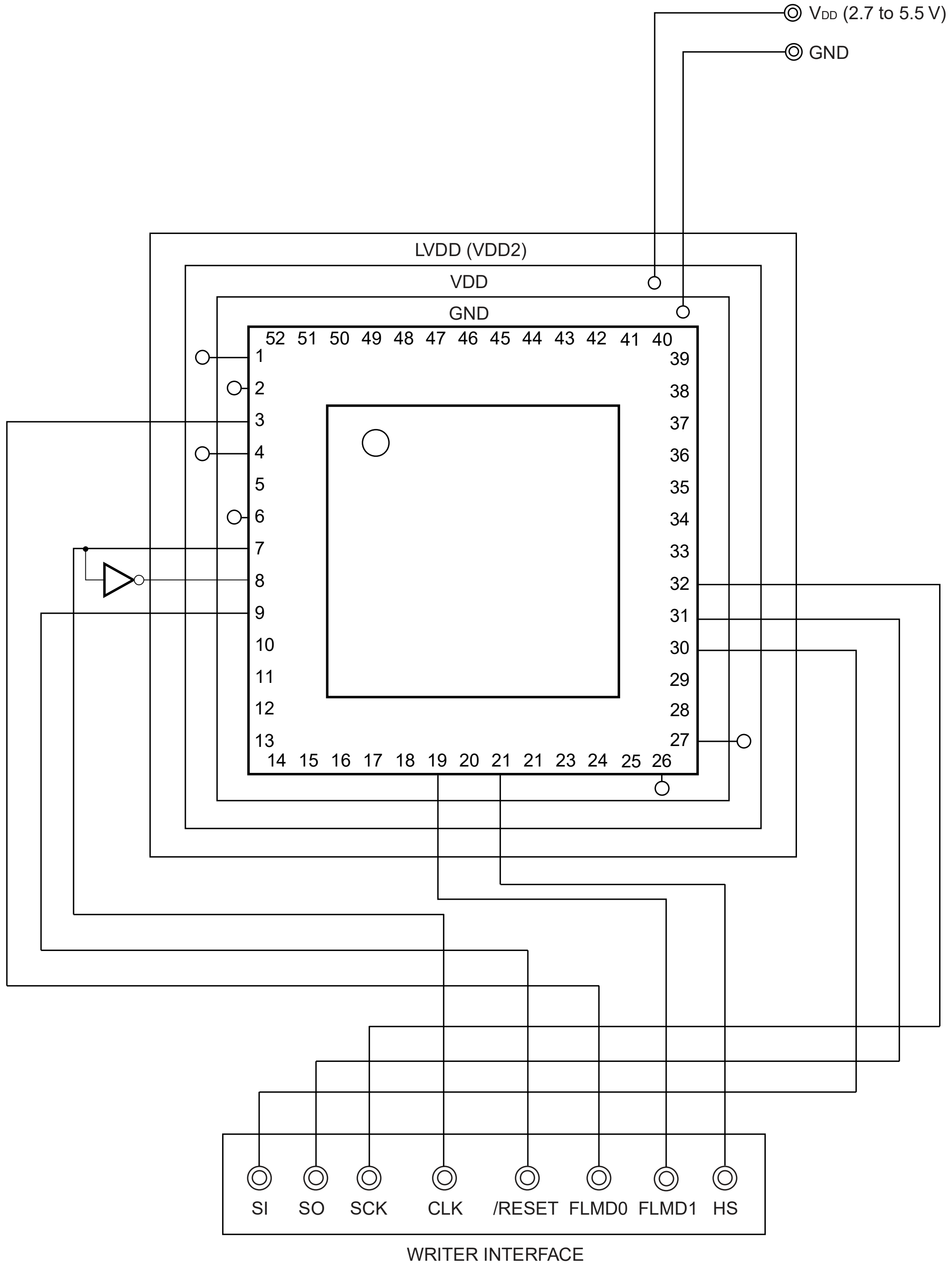
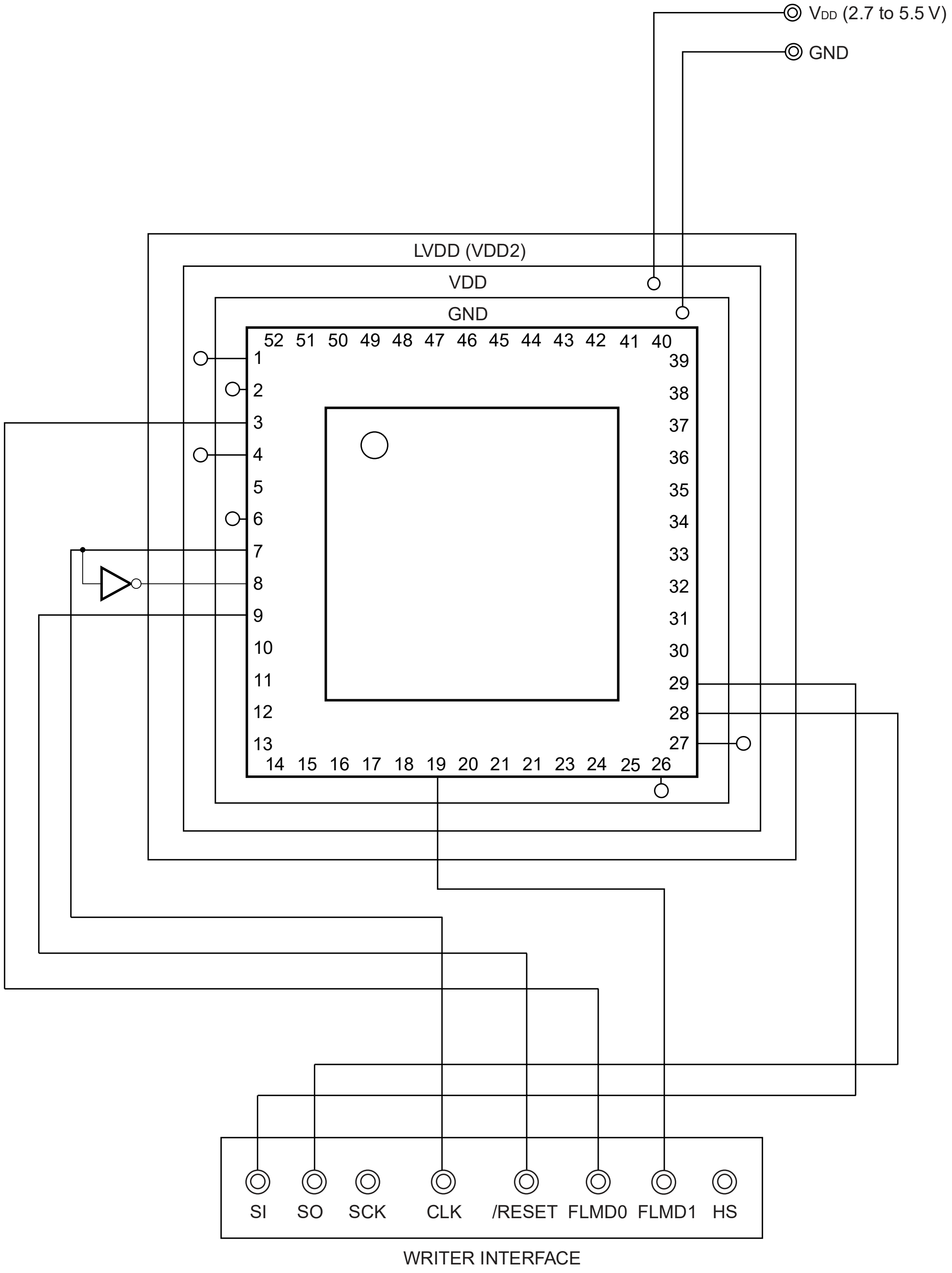


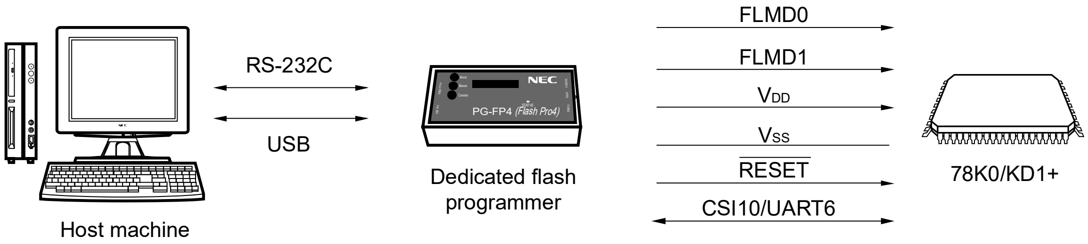
Figure 24-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART6) Mode



24.3 Programming Environment

The environment required for writing a program to the flash memory of the 78K0/KD1+ is illustrated below.

Figure 24-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash programmer is necessary.

To interface between the dedicated flash programmer and the 78K0/KD1+, CSI10 or UART6 is used for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

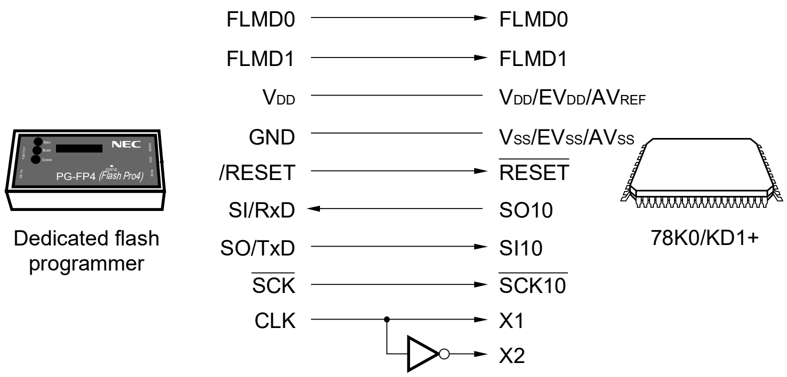
24.4 Communication Mode

Communication between the dedicated flash programmer and the 78K0/KD1+ is established by serial communication via CSI10 or UART6 of the 78K0/KD1+.

(1) CSI10

Transfer rate: 200 kHz to 2 MHz

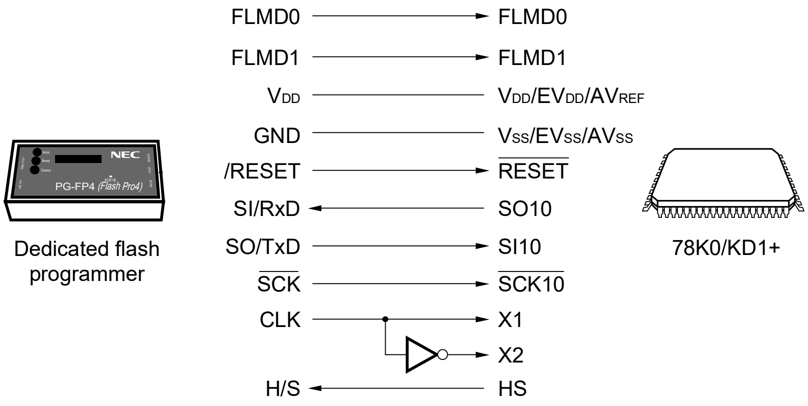
Figure 24-6. Communication with Dedicated Flash Programmer (CSI10)



(2) CSI communication mode supporting handshake

Transfer rate: 200 kHz to 2 MHz

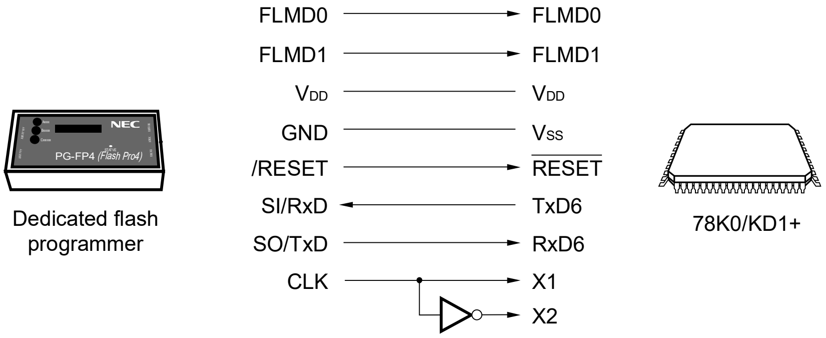
Figure 24-7. Communication with Dedicated Flash Programmer (CSI10 + HS)



(3) UART6

Transfer rate: 4800 to 76800 bps

Figure 24-8. Communication with Dedicated Flash Programmer (UART6)



If Flashpro IV is used as the dedicated flash programmer, Flashpro IV generates the following signal for the 78K0/KD1+. For details, refer to the Flashpro IV Manual.

Table 24-4. Pin Connection

Flashpro IV			78K0/KD1+	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI00	UART6
FLMD0	Output	Mode signal	FLMD0	◎	◎
FLMD1	Output	Mode signal	FLMD1	○	○
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}, EV_{DD}, AV_{REF}	◎	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎	◎
CLK	Output	Clock output to 78K0/KD1+	X1, X2[Note]	○	○
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD6	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD6	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×
H/S	Input	Handshake signal	HS	△	×

Note When using the clock out of the flash programmer, connect CLK of the programmer to X1, and connect its inverse signal to X2.

Remark ◎: Be sure to connect the pin.
○: The pin does not have to be connected if the signal is generated on the target board.
×: The pin does not have to be connected.
△: In handshake mode

24.5 Handling of Pins on Board

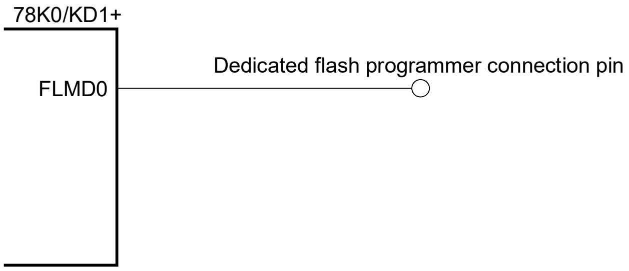
To write the flash memory on-board, connectors that connect the dedicated flash programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

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.

24.5.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. An FLMD0 pin connection example is shown below.

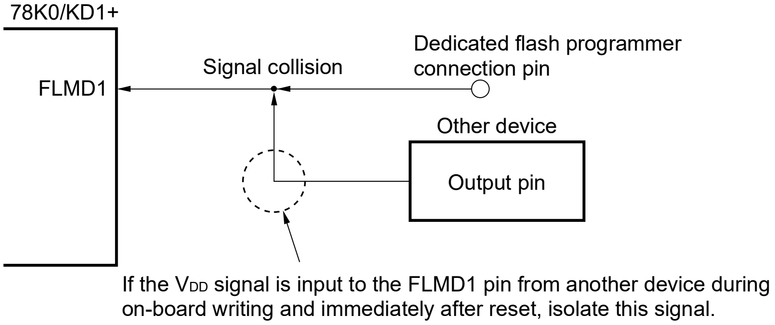
Figure 24-9. FLMD0 Pin Connection Example



24.5.2 FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so the FLMD1 pin must be the same voltage as V_{SS}. An FLMD1 pin connection example is shown below.

Figure 24-10. FLMD1 Pin Connection Example


78K0/KD1+
FLMD1
Signal collision
Dedicated flash programmer connection pin
Other device
Output pin
If the VDD signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

24.5.3 Serial interface pins

The pins used by each serial interface are listed below.

Table 24-5. Pins Used by Each Serial Interface

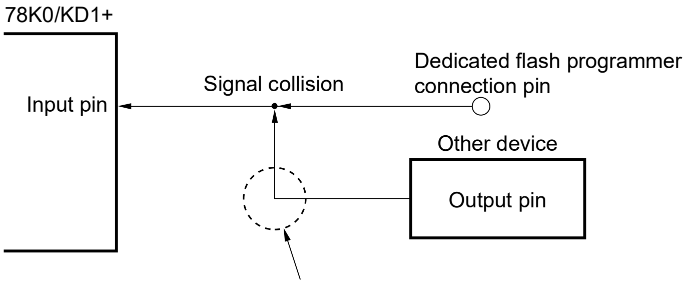
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{\text{SCK10}}$
CSI10 + HS	SO10, SI10, $\overline{\text{SCK10}}$, HS/P15
UART6	TxD6, RxD6

To connect the dedicated flash programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

Figure 24-11. Signal Collision (Input Pin of Serial Interface)

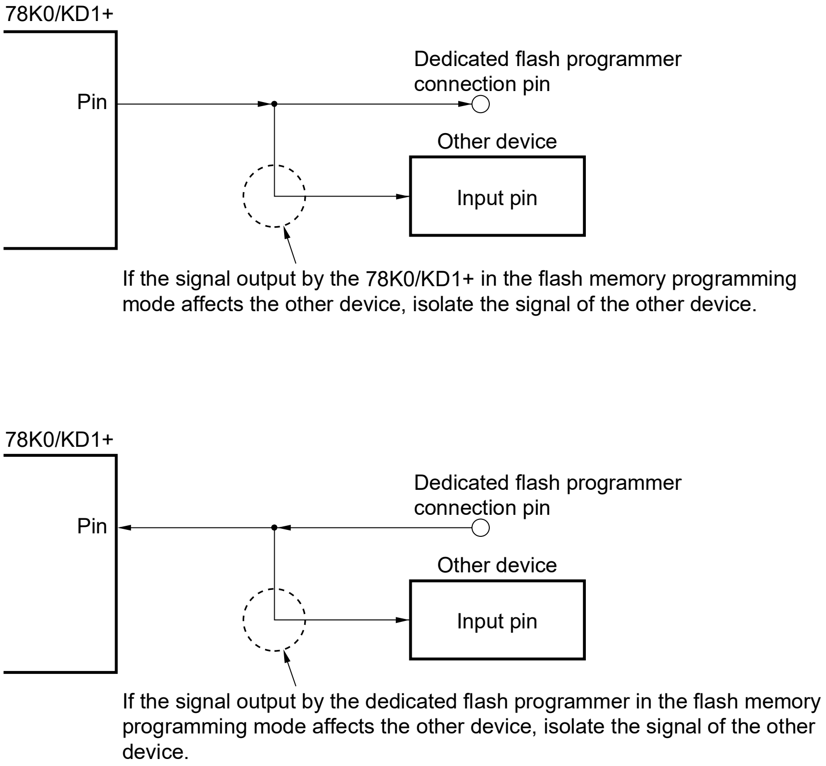


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash programmer. Therefore, isolate the signal of the other device.

(2) Malfunction of other device

If the dedicated flash programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 24-12. Malfunction of Other Device


78K0/KD1+
Pin
Dedicated flash programmer connection pin
Other device
Input pin
78K0/KD1+
Pin
Dedicated flash programmer connection pin
Other device
Input pin

If the signal output by the 78K0/KD1+ in the flash memory programming mode affects the other device, isolate the signal of the other device.

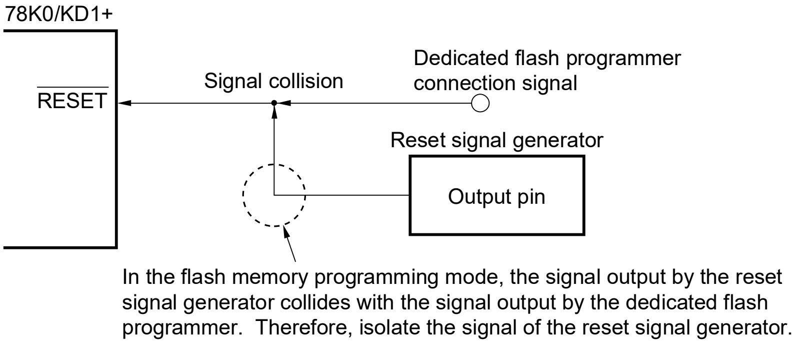
If the signal output by the dedicated flash programmer in the flash memory programming mode affects the other device, isolate the signal of the other device.

24.5.4 $\overline{\text{RESET}}$ pin

If the reset signal of the dedicated flash programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash programmer.

Figure 24-13. Signal Collision ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, the signal output by the reset signal generator collides with the signal output by the dedicated flash programmer. Therefore, isolate the signal of the reset signal generator.

24.5.5 Port pins

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.

24.5.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the programmer, however, connect the clock out of the programmer to X1, and its inverse signal to X2.

24.5.7 Power supply

To use the supply voltage output of the flash programmer, connect the V_{DD} pin to V_{DD} of the flash programmer, and the V_{SS} pin to V_{SS} of the flash programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

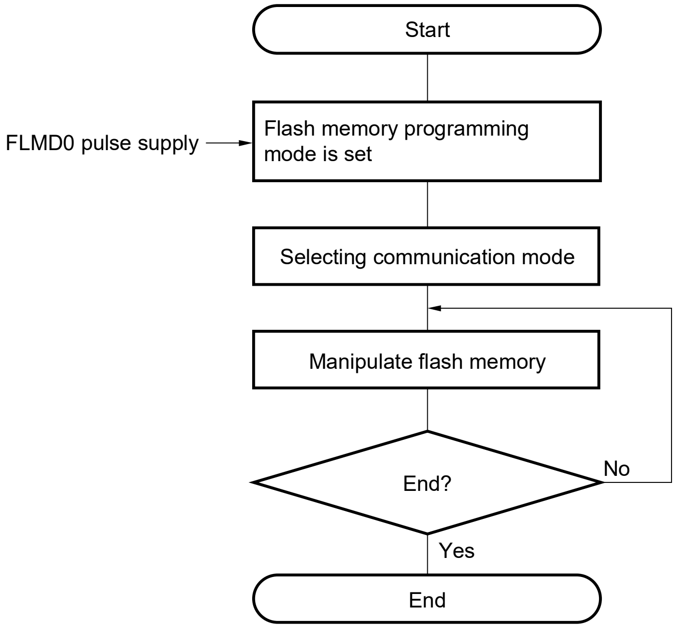
Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

24.6 Programming Method

24.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 24-14. Flash Memory Manipulation Procedure


Start
FLMD0 pulse supply
Flash memory programming mode is set
Selecting communication mode
Manipulate flash memory
End?
No
Yes
End

24.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash programmer, set the 78K0/KD1+ in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 24-15. Flash Memory Programming Mode

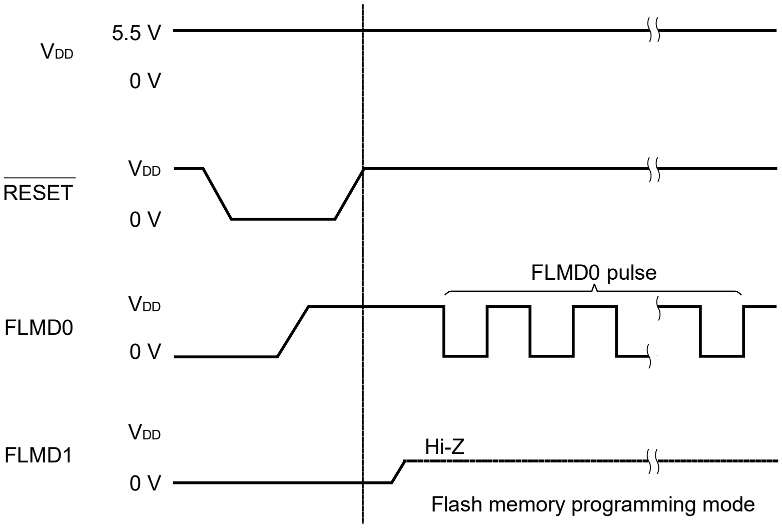


Table 24-6. Relationship Between FLMD0, FLMD1 Pins and Operation Mode After Reset Release

FLMD0	FLMD1	Operation Mode
0	Any	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

24.6.3 Selecting communication mode

In the 78K0/KD1+, a communication mode is selected by inputting pulses (up to 11 pulses) to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 24-7. Communication Modes

Communication Mode	Standard Setting[Note 1]					Pins Used	Number of FLMD0 Pulses
	Port	Speed	On Target	Frequency	Multiply Rate		
UART (UART6)	UART-ch0	4800 to 76800 bps[Notes 2, 3]	Arbitrary	2 to 16 MHz	1.0	TxD6, RxD6	0
3-wire serial I/O (CSI10)	SIO-ch0	200 kHz to 2 MHz[Note 2]				SO10, SI10, $\overline{\text{SCK10}}$	8
3-wire serial I/O with handshake (CSI10 + HS)	SIO-H/S	200 kHz to 2 MHz[Note 2]				SO10, SI10, $\overline{\text{SCK10}}$, HS/P15	11

Notes 1. Selection items for Standard settings on Flashpro IV.

2. The possible setting range differs depending on the voltage. For details, refer to the chapters of electrical specifications.

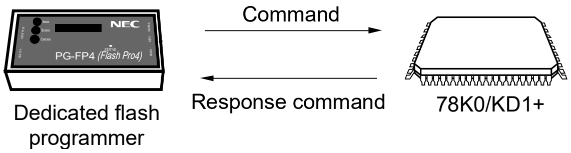
3. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

Caution When UART6 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after the FLMD0 pulse has been received.

24.6.4 Communication commands

The 78K0/KD1+ communicates with the dedicated flash programmer by using commands. The signals sent from the flash programmer to the 78K0/KD1+ are called commands, and the commands sent from the 78K0/KD1+ to the dedicated flash programmer are called response commands.

Figure 24-16. Communication Commands



The flash memory control commands of the 78K0/KD1+ are listed in the table below. All these commands are issued from the programmer and the 78K0/KD1+ performs processing corresponding to the respective commands.

Table 24-8. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Batch verify command	Compares the contents of the entire memory with the input data.
Erase	Batch erase command	Erases the contents of the entire memory.
Blank check	Batch blank check command	Checks the erasure status of the entire memory.
Data write	High-speed write command	Writes data by specifying the write address and number of bytes to be written, and executes a verify check.
	Successive write command	Writes data from the address following that of the high-speed write command executed immediately before, and executes a verify check.
System setting, control	Status read command	Obtains the operation status
	Oscillation frequency setting command	Sets the oscillation frequency
	Erase time setting command	Sets the erase time for batch erase
	Write time setting command	Sets the write time for writing data
	Baud rate setting command	Sets the baud rate when UART is used
	Silicon signature command	Reads the silicon signature information
	Reset command	Escapes from each status

The 78K0/KD1+ returns a response command for the command issued by the dedicated flash programmer. The response commands sent from the 78K0/KD1+ are listed below.

Table 24-9. Response Commands

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

24.7 Flash Memory Programming by Self-Writing

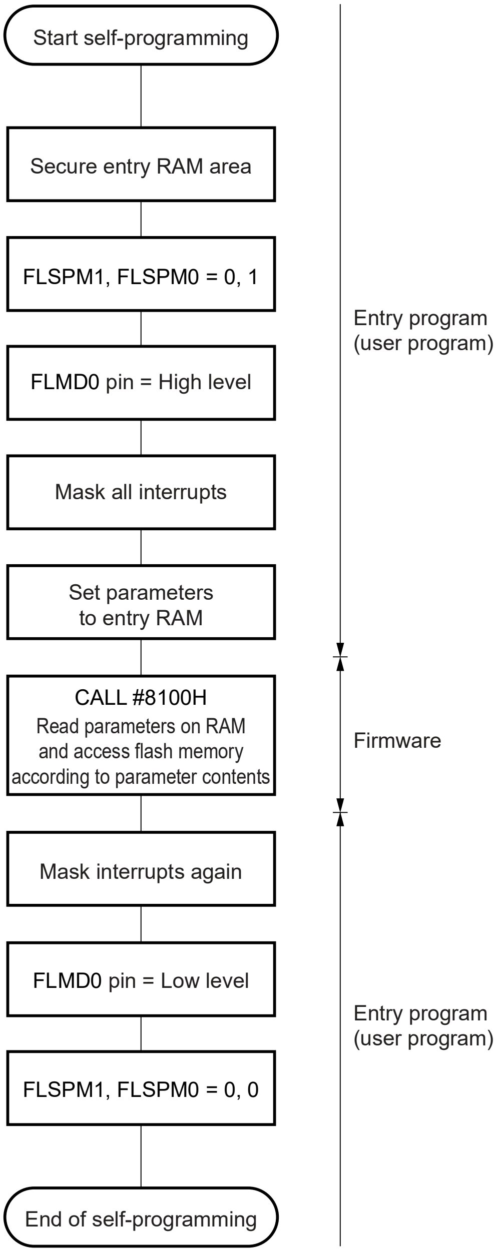
The 78K0/KD1+ supports a self-programming function that can be used to rewrite the flash memory via a user program, so that the program can be upgraded in the field.

The programming mode is selected by bits 0 and 1 (FLSPM0 and FLSPM1) of the flash programming mode control register (FLPMC).

The procedure of self-programming is illustrated below.

Remark For details of the self programming function, refer to a separate document to be published (document name: 78K0/Kx1+ Application Note, release schedule: Pending).

Figure 24-17. Self-Programming Procedure



24.7.1 Registers used for self-programming function

The following three registers are used for the self-programming function.

- Flash programming mode control register (FLPMC)
- Flash protect command register (PFCMD)
- Flash status register (PFS)

(1) Flash programming mode control register (FLPMC)

This register is used to enable or disable writing or erasing of the flash memory and to set the operation mode during self-programming.

The FLPMC can be written only in a specific sequence (refer to **24.7.1 (2) Flash protect command register (PFCMD)**) so that the application system does not stop inadvertently due to malfunction caused by noise or program hang-up.

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

$\overline{\text{RESET}}$ input sets FLPMC to 0XH[Note].

Note Differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

Figure 24-18. Format of Flash Programming Mode Control Register (FLPMC)

Address: FFC4H After reset: 0×H[Note 1] R/W[Note 2]

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	0	FWEDIS	FWEPR	FLSPM1	FLSPM0

FWEDIS	Control of flash memory writing/erasing
0	Writing/erasing enabled[Note 3]
1	Writing/erasing disabled

FWEPR	Status of FLMD0 pin
0	Low level
1	High level[Note 3]

FLSPM1[Note 4]	FLSPM0[Note 4]	Selection of operation mode during self-programming
0	0	Normal mode Instructions of flash memory can be fetched from all addresses.
0	1	Self-programming mode A1 Firmware can be called (CALL #8100H).
1	1	Self-programming mode A2 Instructions are fetched from firmware ROM. This mode is set in firmware and cannot be set by the user.
1	0	Setting prohibited

Notes 1. Differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

2. Bit 2 (FWEPR) is read-only.

3. For actual writing/erasing, the FLMD0 pin must be high (FWEPR = 1), as well as FWEDIS = 0.

FWEDIS	FWEPR	Enable or disable of flash memory writing/erasing
0	1	Writing/erasing enabled
Other than above		Writing/erasing disabled

4. The user ROM (flash memory) or firmware ROM can be selected by FLSPM1 and FLSPM0, and the operation mode set on the application system by the mode pin or the self-programming mode can be selected.

Cautions 1. Be sure to keep FWEDIS at 0 until writing or erasing of the flash memory is completed.

2. Make sure that FWEDIS = 1 in the normal mode.

3. Manipulate FLSPM1 and FLSPM0 after execution branches to the internal RAM. The address of the flash memory is specified by an address signal from the CPU when FLSPM1 = 0 or the set value of the firmware written when FLSPM1 = 1. In the on-board mode, the specifications of FLSPM1 and FLSPM0 are ignored.

(2) Flash protect command register (PFCMD)

If the application system stops inadvertently due to malfunction caused by noise or program hang-up, an operation to write the flash programming mode control register (FLPMC) may have a serious effect on the system. PFCMD is used to protect FLPMC from being written, so that the application system does not stop inadvertently.

Writing FLPMC is enabled only when a write operation is performed in the following specific sequence.

<1> Write a specific value to PFCMD (PFCMD = A5H)
<2> Write the value to be set to FLPMC (writing in this step is invalid)
<3> Write the inverted value of the value to be set to FLPMC (writing in this step is invalid)
<4> Write the value to be set to FLPMC (writing in this step is valid)

This rewrites the value of the register, so that the register cannot be written illegally.

Occurrence of an illegal store operation can be checked by bit 0 (FPRERR) of the flash status register (PFS).

A5H must be written to PFCMD each time the value of FLPMC is changed.

PFCMD can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes PFCMD undefined.

Figure 24-19. Format of Flash Protect Command Register (PFCMD)

Address: FFC0H After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) Flash status register (PFS)

If data is not written to the flash programming mode control register (FLPMC), which is protected, in the correct sequence (writing the flash protect command register (PFCMD)), FLPMC is not written and a protection error occurs. If this happens, bit 0 of PFS (FPRERR) is set to 1.

This bit is a cumulative flag. After checking FPRERR, clear it by writing 0 to it.

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

$\overline{\text{RESET}}$ input clears PFS to 00H.

Figure 24-20. Format of Flash Status Register (PFS)

Address: FFC2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

The operating conditions of the FPRERR flag are as follows.

<Setting conditions>

- If PFCMD is written when the store instruction operation recently performed on a peripheral register is not to write a specific value (A5H) to PFCMD
- If the first store instruction operation after <1> is on a peripheral register other than FLPMC
- If the first store instruction operation after <2> is on a peripheral register other than FLPMC
- If a value other than the inverted value of the value to be set to FLPMC is written by the first store instruction after <2>
- If the first store instruction operation after <3> is on a peripheral register other than FLPMC
- If a value other than the value to be set to FLPMC (value written in <2>) is written by the first store instruction after <3>

Remark The numbers in angle brackets above correspond to the those in **(2) Flash protect command register (PFCMD)**.

<Reset conditions>

- If 0 is written to the FPRERR flag
- If $\overline{\text{RESET}}$ is input

<Example of description in specific sequence>

To write 05H to FLPMC

```
MOV    PFCMD, #0A5H     ; Writes A5H to PFCMD.
MOV    FLPMC, #05H      ; Writes 05H to FLPMC.
MOV    FLPMC, #0FAH     ; Writes 0FAH (inverted value of 05H) to FLPMC.
MOV    FLPMC, #05H      ; Writes 05H to FLPMC.
```

24.8 Boot Swap Function

The 78K0/KD1+ has a boot swap function.

Even if a momentary power failure occurs for some reason while the boot area is being rewritten by self-programming and the program in the boot area is lost, the boot swap function can execute the program correctly after re-application of power, reset, and start.

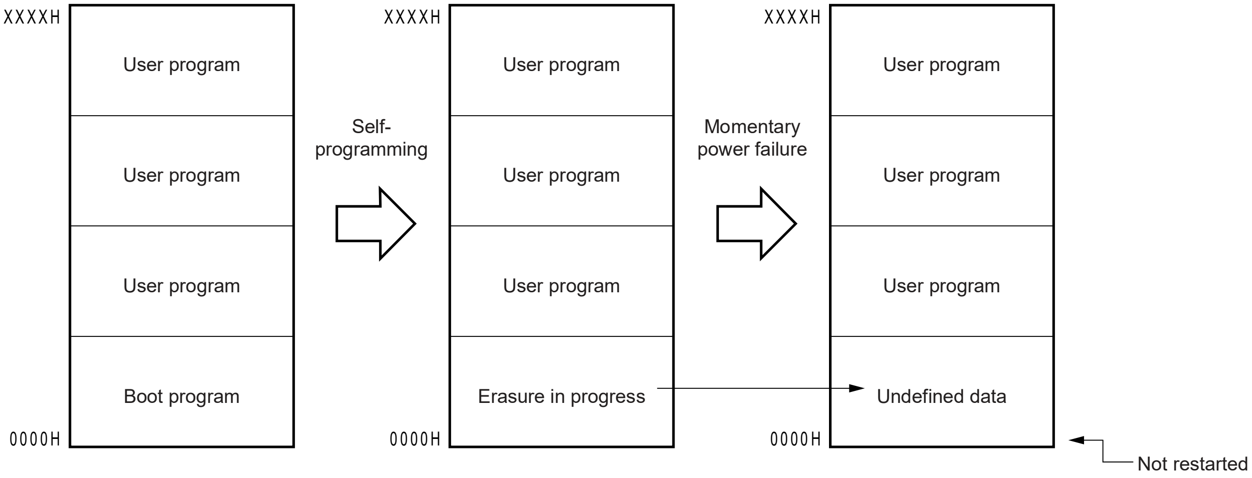
24.8.1 Outline of boot swap function

Before erasing the boot program area by self-programming, write a new boot program to the block to be swapped, and also set the boot flag[Note]. Even if a momentary power failure occurs, the address is swapped when the system is reset and started next time. Consequently, the above area to be swapped is used as a boot area, and the program is executed correctly. Figure 24-21 shows an image of the boot swap function.

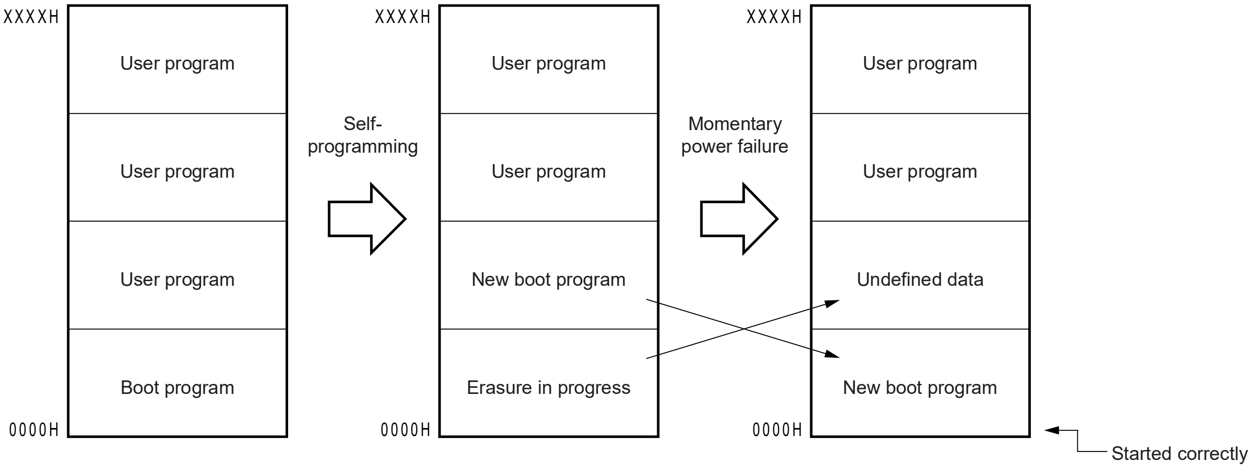
Note The boot flag is controlled by the flash memory control firmware of the 78K0/KD1+.

Figure 24-21. Image of Boot Swap Function

(1) If boot swap is not supported



(2) If boot swap is supported



24.8.2 Memory map and boot area

Figure 24-22 shows the memory map and boot area. The boot program area of the 78K0/KD1+ is in 4 KB units. When boot swap is executed, boot cluster 0 and boot cluster 1 in the figure are exchanged.

Figure 24-22. Memory Map and Boot Area (1/4)

(1) μPD78F0122H

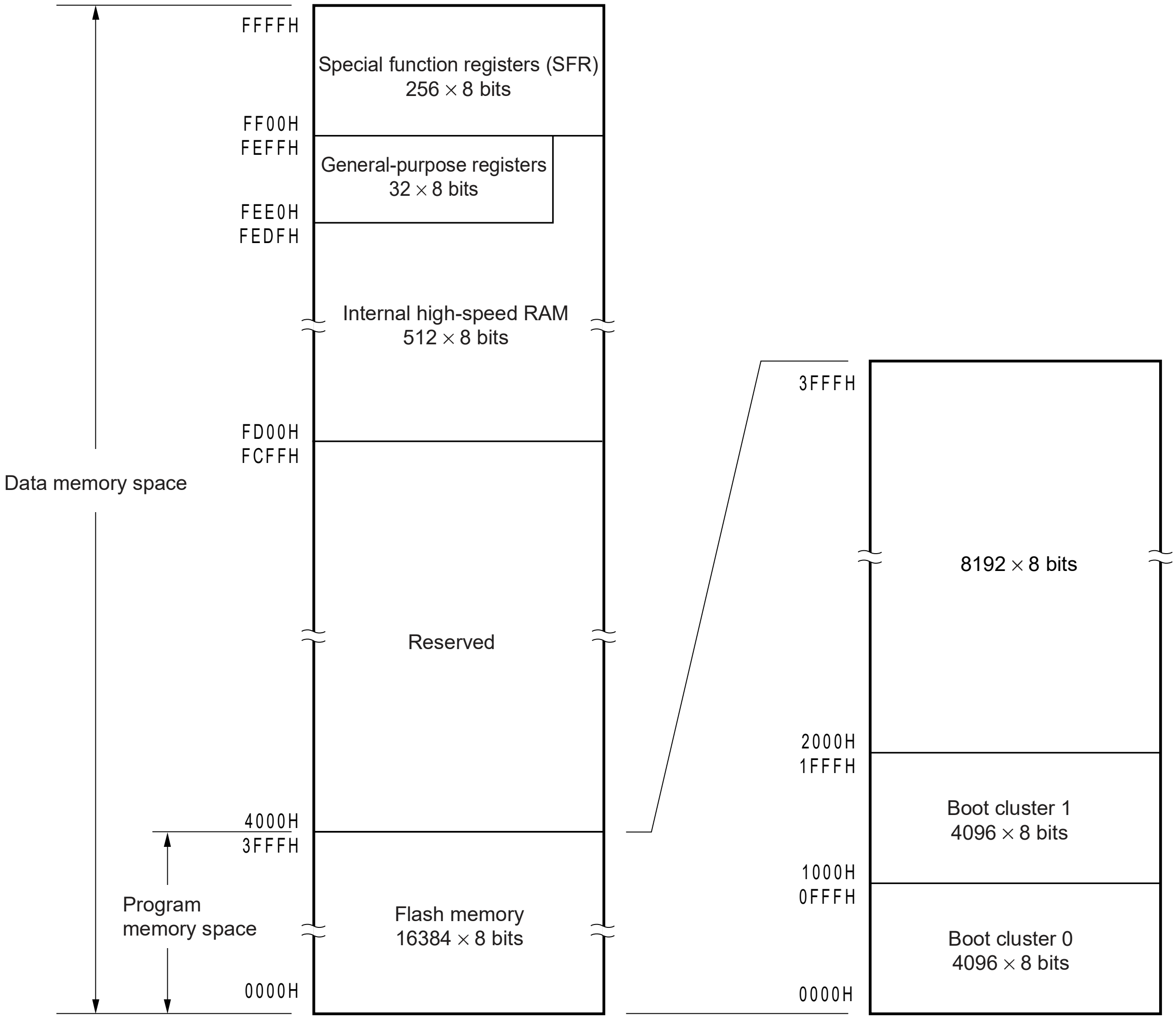

FFFFH
Special function registers (SFR) 256 × 8 bits
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Internal high-speed RAM 512 × 8 bits
FD00H
FCFFH
Data memory space
Reserved
4000H
3FFFH
Program memory space
Flash memory 16384 × 8 bits
0000H
3FFFH
8192 × 8 bits
2000H
1FFFH
Boot cluster 1 4096 × 8 bits
1000H
0FFFH
Boot cluster 0 4096 × 8 bits
0000H

Figure 24-22. Memory Map and Boot Area (2/4)

(2) μPD78F0123H

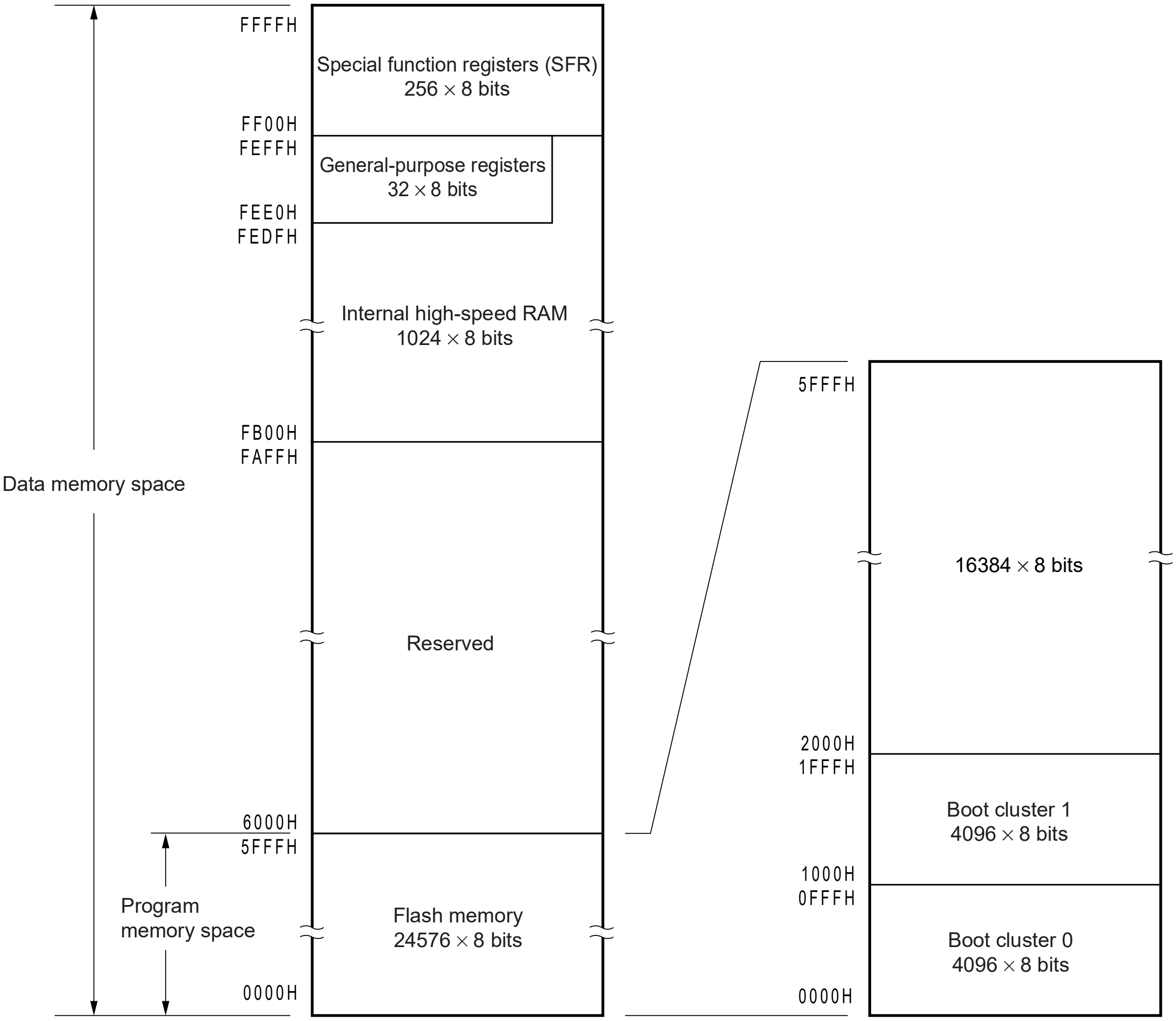


Figure 24-22. Memory Map and Boot Area (3/4)

(3) μPD78F0124H

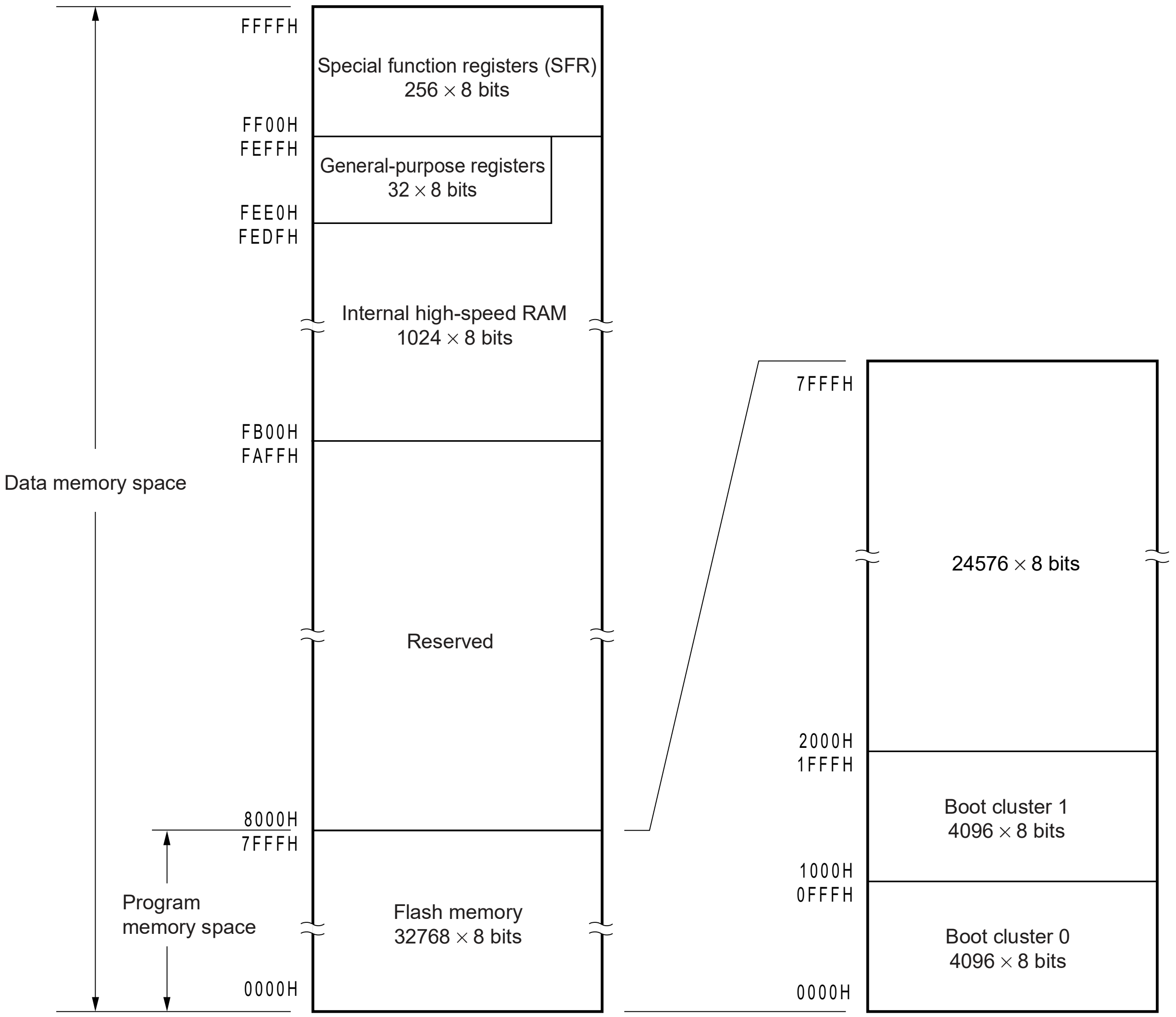
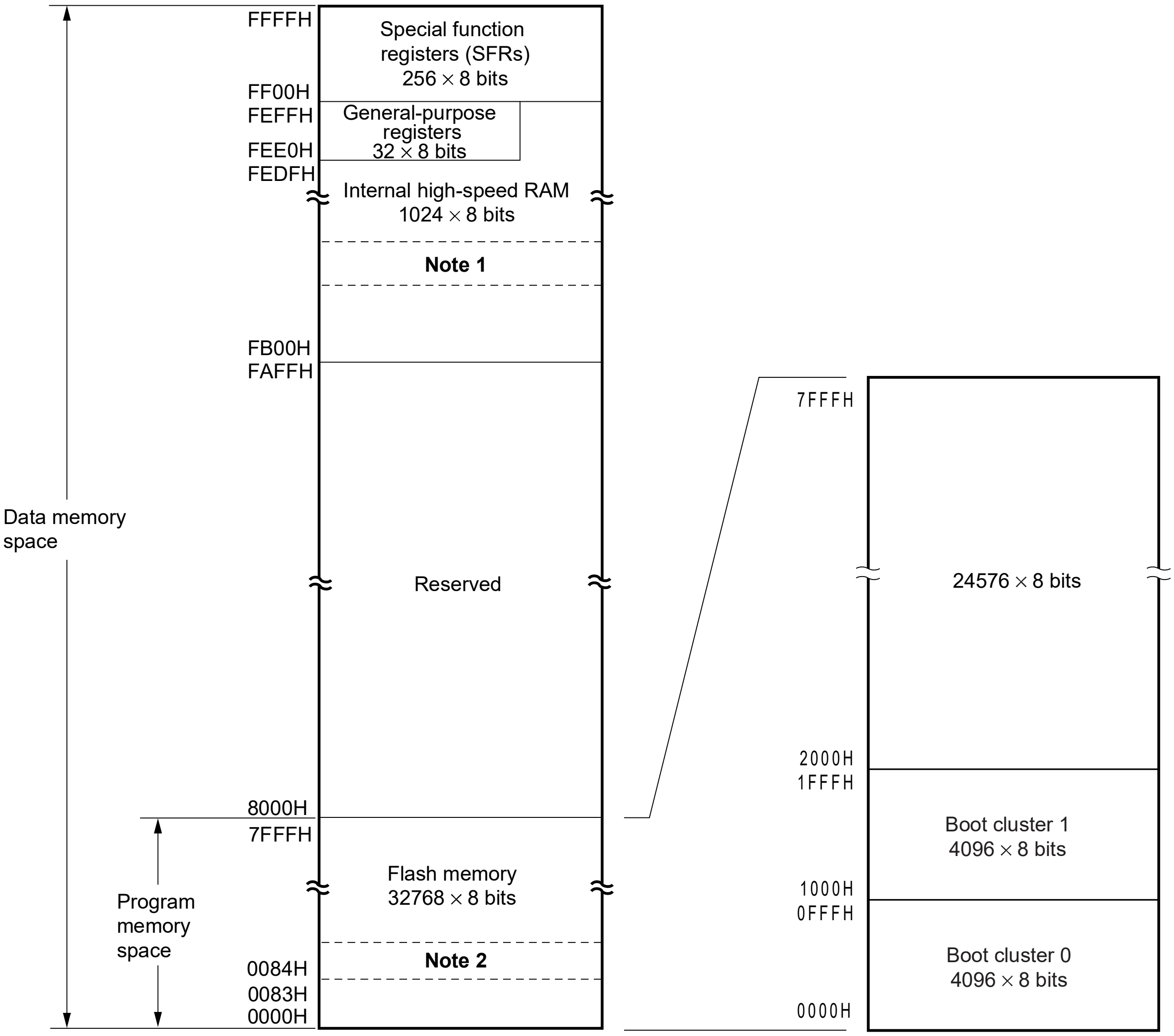


Figure 24-22. Memory Map and Boot Area (4/4)

(4) μPD78F0124HD


FFFFH
Special function registers (SFRs) 256 × 8 bits
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Internal high-speed RAM 1024 × 8 bits
Note 1
FB00H
FAFFH
Data memory space
Reserved
8000H
7FFFH
Flash memory 32768 × 8 bits
Program memory space
Note 2
0084H
0083H
0000H
7FFFH
24576 × 8 bits
2000H
1FFFH
Boot cluster 1 4096 × 8 bits
1000H
0FFFH
Boot cluster 0 4096 × 8 bits
0000H

Notes 1. During on-chip debugging, 9 bytes (planned) of this area are used as the user data backup area for communication.

2. During on-chip debugging, use of this area is disabled since it is used as the communication command area (256 bytes to 1 KB).

CHAPTER 25 ON-CHIP DEBUG FUNCTION (μPD78F0124HD ONLY)

The μPD78F0124HD uses the V_{DD}, FLMD0, $\overline{\text{RESET}}$, X1 (or P31), X2 (or P32), and V_{SS} pins to communicate with the host machine via an in-circuit emulator (IECUBE for 78K0/Kx1+ (provisional name)) for on-chip debugging. Whether X1 and P31, or X2 and P32 are used can be selected.

Cautions 1. Be sure to pull down P31 after reset to prevent malfunction.

2. When using P31 for the on-chip debug function, it is recommended not to use P31 for any purpose other than on-chip debugging, or to control P31 with an external circuit using P130 (that outputs a pseudo CPU reset signal).

Remark For details of the on-chip debug function, refer to a separate document (document name, release schedule: Pending).

CHAPTER 26 INSTRUCTION SET

This chapter lists each instruction set of the 78K0/KD1+ in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

26.1 Conventions Used in Operation List

26.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] 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 specification.

Table 26-1. Operand Identifiers and Specification Methods

Identifier	Specification 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[Note] Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, refer to **Table 3-5 Special Function Register List**.

26.1.2 Description of operation column

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
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
NMIS:	Non-maskable interrupt servicing flag
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\quad}$:	Inverted data
addr16:	16-bit immediate data or label
jdisp8:	Signed 8-bit data (displacement value)

26.1.3 Description of flag operation column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is restored

26.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9	A ← (addr16)			
		!addr16, A	3	8	9	(addr16) ← A			
		PSW, #byte	3	–	7	PSW ← byte	×	×	×
		A, PSW	2	–	5	A ← PSW			
		PSW, A	2	–	5	PSW ← A	×	×	×
		A, [DE]	1	4	5	A ← (DE)			
		[DE], A	1	4	5	(DE) ← A			
		A, [HL]	1	4	5	A ← (HL)			
		[HL], A	1	4	5	(HL) ← A			
		A, [HL + byte]	2	8	9	A ← (HL + byte)			
		[HL + byte], A	2	8	9	(HL + byte) ← A			
		A, [HL + B]	1	6	7	A ← (HL + B)			
		[HL + B], A	1	6	7	(HL + B) ← A			
		A, [HL + C]	1	6	7	A ← (HL + C)			
		[HL + C], A	1	6	7	(HL + C) ← A			
	XCH	A, r Note 3	1	2	–	A ↔ r			
		A, saddr	2	4	6	A ↔ (saddr)			
		A, sfr	2	–	6	A ↔ (sfr)			
		A, !addr16	3	8	10	A ↔ (addr16)			
		A, [DE]	1	4	6	A ↔ (DE)			
		A, [HL]	1	4	6	A ↔ (HL)			
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)			
		A, [HL + B]	2	8	10	A ↔ (HL + B)			
		A, [HL + C]	2	8	10	A ↔ (HL + C)			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp Note 3	1	4	–	AX ← rp			
		rp, AX Note 3	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp			
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r	×	×	×
		r, A	2	4	–	r, CY ← r + A	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	×	×	×
		r, A	2	4	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	4	–	A ← A ∨ r	×		
		r, A	2	4	–	r ← r ∨ A	×		
		A, saddr	2	4	5	A ← A ∨ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×		
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	4	–	A ← A ⊻ r	×		
		r, A	2	4	–	r ← r ⊻ A	×		
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×		
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×		
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×		
	CMP	A, #byte	2	4	–	A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A – r	×	×	×
		r, A	2	4	–	r – A	×	×	×
		A, saddr	2	4	5	A – (saddr)	×	×	×
		A, !addr16	3	8	9	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A – (HL + C)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
Call/return	**CALL**	!addr16	3	7	–	(SP – 1) ← $(PC + 3)_H$, (SP – 2) ← $(PC + 3)_L$, PC ← addr16, SP ← SP – 2	
	CALLF	!addr11	2	5	–	(SP – 1) ← $(PC + 2)_H$, (SP – 2) ← $(PC + 2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP – 2	
	CALLT	[addr5]	1	6	–	(SP – 1) ← $(PC + 1)_H$, (SP – 2) ← $(PC + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2	
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← $(PC + 1)_H$, (SP – 3) ← $(PC + 1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP – 3, IE ← 0	
	RET		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2	
	RETI		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
	RETB		1	6	–	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3	R R R
Stack manipulate	**PUSH**	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1	
		rp	1	4	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2	
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP + 1	R R R
		rp	1	4	–	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2	
	MOVW	SP, #word	4	–	10	SP ← word	
		SP, AX	2	–	8	SP ← AX	
		AX, SP	2	–	8	AX ← SP	
Unconditional branch	**BR**	!addr16	3	6	–	PC ← addr16	
		$addr16	2	6	–	PC ← PC + 2 + jdisp8	
		AX	2	8	–	PC_H ← A, PC_L ← X	
Conditional branch	**BC**	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)			
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n			
	NOP		1	2	–	No Operation			
	EI		2	–	6	IE ← 1 (Enable Interrupt)			
	DI		2	–	6	IE ← 0 (Disable Interrupt)			
	HALT		2	6	–	Set HALT Mode			
	STOP		2	6	–	Set STOP Mode			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

26.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL+byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 27 ELECTRICAL SPECIFICATIONS (TARGET)

Caution These specifications show target values, which may change after device evaluation. The operating voltage range may also change.

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	EV_{DD}			–0.3 to +6.5	V
	V_{SS}			–0.3 to +0.3	V
	EV_{SS}			–0.3 to +0.3	V
	AV_{REF}			–0.3 to V_{DD} + 0.3[Note]	V
	AV_{SS}			–0.3 to +0.3	V
Input voltage	V_{I1}	P00 to P03, P10 to P17, P20 to P27, P30 to P33, P60, P61, P70 to P77, P120, P140, X1, X2, XT1, XT2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note]	V
	V_{I2}	P62, P63	N-ch open drain	–0.3 to +13	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}			AV_{SS} – 0.3 to AV_{REF} + 0.3[Note] and –0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin		–10	mA
		Total of all pins –60 mA	P00 to P03, P10 to P14, P70 to P77	–30	mA
			P15 to P17, P30 to P33, P120, P130, P140	–30	mA
Output current, low	I_{OL}	Per pin	P00 to P03, P10 to P17, P30 to P33, P70 to P77, P120, P130, P140	20	mA
			P60 to P63	30	mA
		Total of all pins 70 mA	P00 to P03, P10 to P14, P70 to P77	35	mA
			P15 to P17, P30 to P33, P60 to P63, P120, P130, P140	35	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
		In flash memory programming mode		–10 to +85	
Storage temperature	T_{stg}			–40 to +125	°C

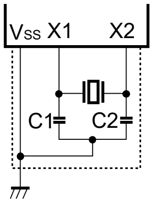
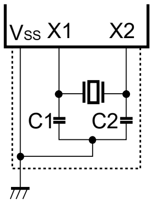
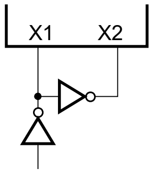
Note Must be 6.5 V or lower.

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 Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

High-Speed System Clock (Crystal/Ceramic) Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		16	MHz
			3.3 V ≤ V_{DD} < 4.0 V	2.0		8.38	
			2.7 V ≤ V_{DD} < 3.3 V	2.0		5.0	
Crystal resonator		Oscillation frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		16	MHz
			3.3 V ≤ V_{DD} < 4.0 V	2.0		8.38	
			2.7 V ≤ V_{DD} < 3.3 V	2.0		5.0	
External clock		X1 input frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	2.0		16	MHz
			3.3 V ≤ V_{DD} < 4.0 V	2.0		8.38	
			2.7 V ≤ V_{DD} < 3.3 V	2.0		5.0	
		X1 input high-/low-level width (t_{XPH}, t_{XPL})	4.0 V ≤ V_{DD} ≤ 5.5 V	30		500	ns
			3.3 V ≤ V_{DD} < 4.0 V	56		500	
			2.7 V ≤ V_{DD} < 3.3 V	96		500	

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the high-speed system clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. Since the CPU is started by the Ring-OSC after reset is released, check the oscillation stabilization time of the high-speed system clock using the oscillation stabilization time status register (OSTC). Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

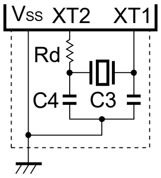
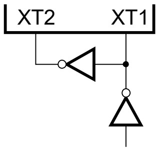
Ring-OSC Oscillator Characteristics

(T_A = −40 to +85°C, 2.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
On-chip Ring-OSC oscillator	Oscillation frequency (f_R)		120	240	480	kHz

Subsystem Clock Oscillator Characteristics

(T_A = −40 to +85°C, 2.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT})[Note]		32	32.768	35	kHz
External clock		XT1 input frequency (f_{XT})[Note]		32		38.5	kHz
		XT1 input high-/low-level width (t_{XTH}, t_{XTL})		12		15	μs

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the subsystem clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures 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 V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. The subsystem clock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the high-speed system clock oscillator. Particular care is therefore required with the wiring method when the subsystem clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/3)

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V[Note 1], 2.0 V ≤ AV_{REF} ≤ V_{DD}[Note 1], V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high	I_{OH}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
		Total of P00 to P03, P10 to P14, P70 to P77	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
		Total of P15 to P17, P30 to P33, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
		Total of all pins	2.0 V ≤ V_{DD} < 4.0 V			–10	mA
Output current, low	I_{OL}	Per pin for P00 to P03, P10 to P17, P30 to P33, P70 to P77, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
		Per pin for P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V			15	mA
		Total of P00 to P03, P10 to P14, P70 to P77	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
		Total of P15 to P17, P30 to P33, P60 to P63, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V			30	mA
		Total of all pins	2.0 V ≤ V_{DD} < 4.0 V			10	mA
Input voltage, high	V_{IH1}	P12, P13, P15	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P00 to P03, P10, P11, P14, P16, P17, P30 to P33, P70 to P77, P120, P140, $\overline{RESET}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0.85V_{DD}		V_{DD}	V
	V_{IH3}	P20 to P27[Note 2]	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7AV_{REF}		AV_{REF}	V
			2.0 V ≤ V_{DD} < 2.7 V	0.8AV_{REF}		AV_{REF}	V
	V_{IH4}	P60, P61	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		V_{DD}	V
	V_{IH5}	P62, P63	2.7 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		12	V
			2.0 V ≤ V_{DD} < 2.7 V	0.8V_{DD}		12	V
	V_{IH6}	X1, X2, XT1, XT2	2.7 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5		V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	V_{DD} – 0.2		V_{DD}	V
Input voltage, low	V_{IL1}	P12, P13, P15	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.2V_{DD}	V
	V_{IL2}	P00 to P03, P10, P11, P14, P16, P17, P30 to P33, P70 to P77, P120, P140, $\overline{RESET}$	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.15V_{DD}	V
	V_{IL3}	P20 to P27[Note 2]	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3AV_{REF}	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.2AV_{REF}	V
	V_{IL4}	P60, P61	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.2V_{DD}	V
	V_{IL5}	P62, P63	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.2V_{DD}	V
	V_{IL6}	X1, X2, XT1, XT2	2.7 V ≤ V_{DD} ≤ 5.5 V	0		0.4	V
			2.0 V ≤ V_{DD} < 2.7 V	0		0.2	V

Notes 1. When high-speed system clock is used: 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}

2. When used as digital input ports, set AV_{REF} = V_{DD}.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/3)

(T_A = −40 to +85°C, 2.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V[Note 1], 2.0 V ≤ AV_{REF} ≤ V_{DD}[Note 1], V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH}	Total of P00 to P03, P10 to P14, P70 to P77 I_{OH} = −25 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = −5 mA	V_{DD} − 1.0			V
		Total of P15 to P17, P30 to P33, P120, P130, P140 I_{OH} = −25 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = −5 mA	V_{DD} − 1.0			V
		I_{OH} = −100 μA	2.0 V ≤ V_{DD} < 4.0 V	V_{DD} − 0.5			V
Output voltage, low	V_{OL1}	Total of P00 to P03, P10 to P14, P70 to P77 I_{OL} = 30 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.3	V
		Total of P15 to P17, P30 to P33, P60 to P63, P120, P130, P140 I_{OL} = 30 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.3	V
		I_{OL} = 400 μA	2.7 V ≤ V_{DD} < 4.0 V			0.4	V
			2.0 V ≤ V_{DD} < 2.7 V			0.5	V
	V_{OL2}	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA			2.0	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD}	P00 to P03, P10 to P17, P30 to P33, $\overline{\text{P60}}$, $\overline{\text{P61}}$, P70 to P77, P120, P140, $\overline{\text{RESET}}$			3	μA
		V_I = AV_{REF}	P20 to P27			3	μA
	I_{LIH2}	V_I = V_{DD}	X1, X2[Note 2], XT1, XT2[Note 2]			20	μA
	I_{LIH3}	V_I = 12 V	P62, P63 (N-ch open drain)			3	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00 to P03, P10 to P17, P20 to P27, P30 to P33, $\overline{\text{P60}}$, $\overline{\text{P61}}$, P70 to P77, P120, P140, $\overline{\text{RESET}}$			−3	μA
	I_{LIL2}		X1, X2[Note 2], XT1, XT2[Note 2]			−20	μA
	I_{LIL3}		P62, P63 (N-ch open drain)			−3[Note 3]	μA
Output leakage current, high	I_{LOH}	V_O = V_{DD}				3	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				−3	μA
Pull-up resistance value	R_L	V_I = 0 V		10	30	100	kΩ
FLMD0 supply voltage	Flmd	In normal operation mode		0		0.2V_{DD}	V

Notes 1. When high-speed system clock is used: 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}

2. When the inverse level of X1 is input to X2 and the inverse level of XT1 is input to XT2.

3. If port 6 has been set to input mode when a read instruction is executed to read from port 6, a low-level input leakage current of up to −45 μA flows during only one cycle. At all other times, the maximum leakage current is −3 μA.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (3/3)

(T$_{A}$ = –40 to +85°C, 2.0 V ≤ V$_{DD}$ = EV$_{DD}$ ≤ 5.5 V[Note 1], 2.0 V ≤ AV$_{REF}$ ≤ V$_{DD}$[Note 1], V$_{SS}$ = EV$_{SS}$ = AV$_{SS}$ = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Crystal/ceramic oscillation operating mode[Note 3]	f_{XP} = 16 MHz V_{DD} = 5.0 V ±10%[Note 4]	When A/D converter is stopped		16.5	33.0	mA
				When A/D converter is operating[Note 7]		17.5	35.0	mA
			f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%[Note 4]	When A/D converter is stopped		10.5	21.0	mA
				When A/D converter is operating[Note 7]		11.5	23.0	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%[Note 4]	When A/D converter is stopped		3.2	6.7	mA
				When A/D converter is operating[Note 7]		3.8	7.9	mA
	I_{DD2}	Crystal/ceramic oscillation HALT mode	f_{XP} = 16 MHz V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		3.0	6.0	mA
				When peripheral functions are operating			15.0	mA
			f_{XP} = 10 MHz V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		2.1	4.8	mA
				When peripheral functions are operating			8.5	mA
			f_{XP} = 5 MHz V_{DD} = 3.0 V ±10%	When peripheral functions are stopped		0.8	1.8	mA
				When peripheral functions are operating			3.0	mA
	I_{DD3}	Ring-OSC operating mode[Note 5]	V_{DD} = 5.0 V ±10%			1.8	7.2	mA
			V_{DD} = 3.0 V ±10%			0.8	3.2	mA
	I_{DD4}	Ring-OSC HALT mode[Note 5]	V_{DD} = 5.0 V ±10%			T.B.D.	T.B.D.	μA
			V_{DD} = 3.0 V ±10%			T.B.D.	T.B.D.	μA
	I_{DD5}	32.768 kHz crystal oscillation operating mode[Notes 5, 6]	V_{DD} = 5.0 V ±10%			43.0	86.0	μA
			V_{DD} = 3.0 V ±10%			30.0	60.0	μA
	I_{DD6}	32.768 kHz crystal oscillation HALT mode[Notes 5, 6]	V_{DD} = 5.0 V ±10%			20.0	40.0	μA
			V_{DD} = 3.0 V ±10%			6.0	12.0	μA
	I_{DD7}	STOP mode	V_{DD} = 5.0 V ±10%	RING: OFF		3.5	35.5	μA
				RING: ON		17.5	63.5	μA
			V_{DD} = 3.0 V ±10%	RING: OFF		3.5	15.5	μA
				RING: ON		11.0	30.5	μA

Notes 1. When high-speed system clock is used: 2.7 V ≤ V$_{DD}$ ≤ 5.5 V, 2.7 V ≤ AV$_{REF}$ ≤ V$_{DD}$

2. Total current flowing through the internal power supply (V$_{DD}$). Peripheral operation current is included (however, the current that flows through the pull-up resistors of ports is not included).

3. I$_{DD1}$ includes peripheral operation current.

4. When PCC = 00H.

5. When high-speed system clock (crystal/ceramic) oscillator is stopped.

6. When Ring-OSC oscillator is stopped.

7. Including the current that flows through the AV$_{REF}$ pin.

AC Characteristics

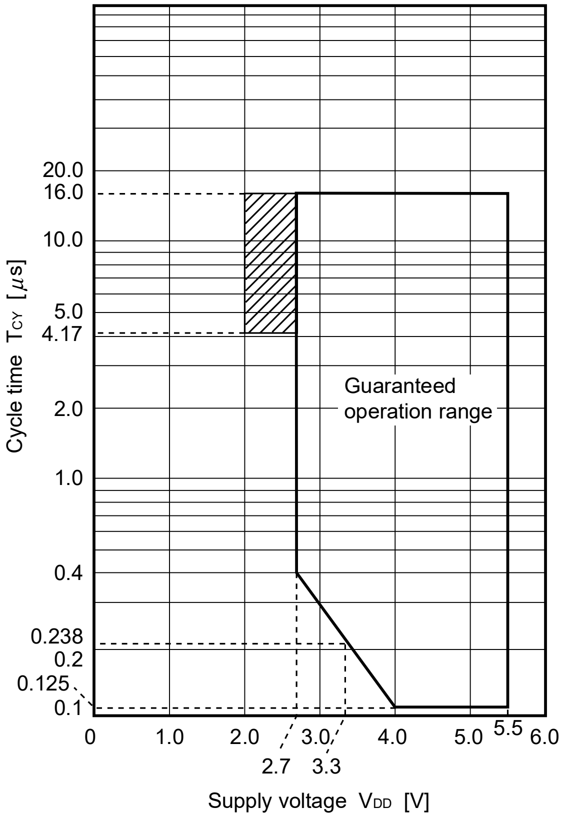
(1) Basic operation

(T_A = –40 to +85°C, 2.0 V $\le V_{DD} = EV_{DD} \le$ 5.5 V, 2.0 V $\le AV_{REF} \le V_{DD}$, $V_{SS} = EV_{SS} = AV_{SS}$ = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock operation	High-speed system clock (crystal/ceramic oscillation clock)	4.0 V $\le V_{DD} \le$ 5.5 V	0.125		16	µs
				3.3 V $\le V_{DD} <$ 4.0 V	0.238		16	µs
				2.7 V $\le V_{DD} <$ 3.3 V	0.4		16	µs
			Ring-OSC clock		4.17	8.33	33.3[Note 1]	µs
		Subsystem clock operation			114	122	125	µs
TI000, TI010 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V $\le V_{DD} \le$ 5.5 V			$2/f_{sam}$ + 0.1[Note 2]			µs
		2.7 V $\le V_{DD} <$ 4.0 V			$2/f_{sam}$ + 0.2[Note 2]			µs
TI50, TI51 input frequency	f_{TI5}	4.0 V $\le V_{DD} \le$ 5.5 V					10	MHz
		2.7 V $\le V_{DD} <$ 4.0 V					5	MHz
TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V $\le V_{DD} \le$ 5.5 V			50			ns
		2.7 V $\le V_{DD} <$ 4.0 V			100			ns
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	2.7 V $\le V_{DD} \le$ 5.5 V			1			µs
		2.0 V $\le V_{DD} <$ 2.7 V			2			µs
Key return input low-level width	t_{KR}	4.0 V $\le V_{DD} \le$ 5.5 V			50			ns
		2.7 V $\le V_{DD} <$ 4.0 V			100			ns
		2.0 V $\le V_{DD} <$ 2.7 V			200			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}	2.7 V $\le V_{DD} \le$ 5.5 V			10			µs
		2.0 V $\le V_{DD} <$ 2.7 V			20			µs

Notes 1. This value may change after evaluation.

2. Selection of $f_{sam} = f_{XP}$, $f_{XP}/4$ or $f_{XP}/256$ is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, $f_{sam} = f_{XP}$.



Remark The values indicated by the shaded section are only when the Ring-OSC clock is selected.

(2) Serial interface

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART mode (UART6, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbps

(b) UART mode (UART0, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbps

(c) 3-wire serial I/O mode (master mode, $\overline{\text{SCK10}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
		3.3 V ≤ V_{DD} < 4.0 V	240			ns
		2.7 V ≤ V_{DD} < 3.3 V	400			ns
$\overline{\text{SCK10}}$ high-/low-level width	t_{KH1}, t_{KL1}		$t_{KCY1}/2 - 10$			ns
SI10 setup time (to $\overline{\text{SCK10}}$↑)	t_{SIK1}		30			ns
SI10 hold time (from $\overline{\text{SCK10}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK10}}$↓ to SO10 output	t_{KSO1}	C = 100 pF[Note]			30	ns

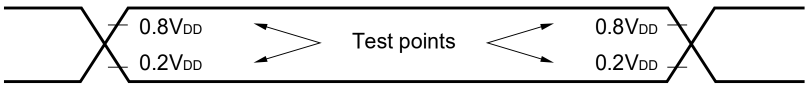
Note C is the load capacitance of the $\overline{\text{SCK10}}$ and SO10 output lines.

(d) 3-wire serial I/O mode (slave mode, $\overline{\text{SCK10}}$... external clock input)

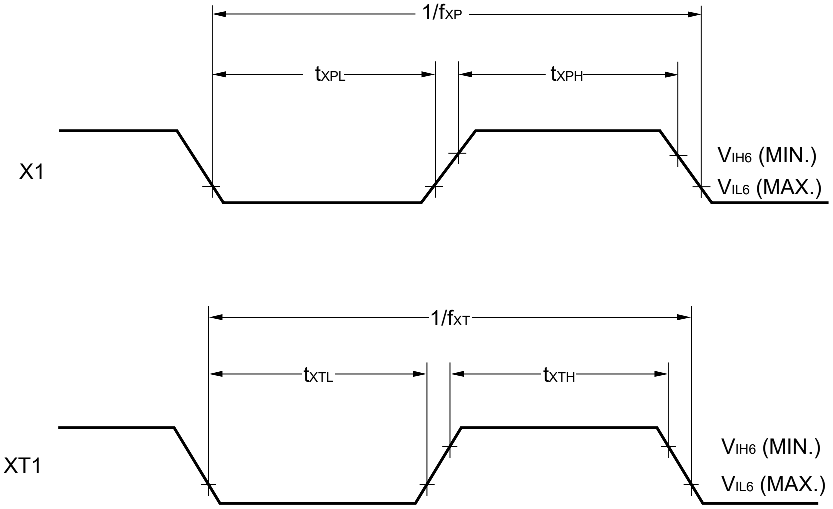
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	t_{KCY2}		400			ns
$\overline{\text{SCK10}}$ high-/low-level width	t_{KH2}, t_{KL2}		$t_{KCY2}/2$			ns
SI10 setup time (to $\overline{\text{SCK10}}$↑)	t_{SIK2}		80			ns
SI10 hold time (from $\overline{\text{SCK10}}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{\text{SCK10}}$↓ to SO10 output	t_{KSO2}	C = 100 pF[Note]			120	ns

Note C is the load capacitance of the SO10 output line.

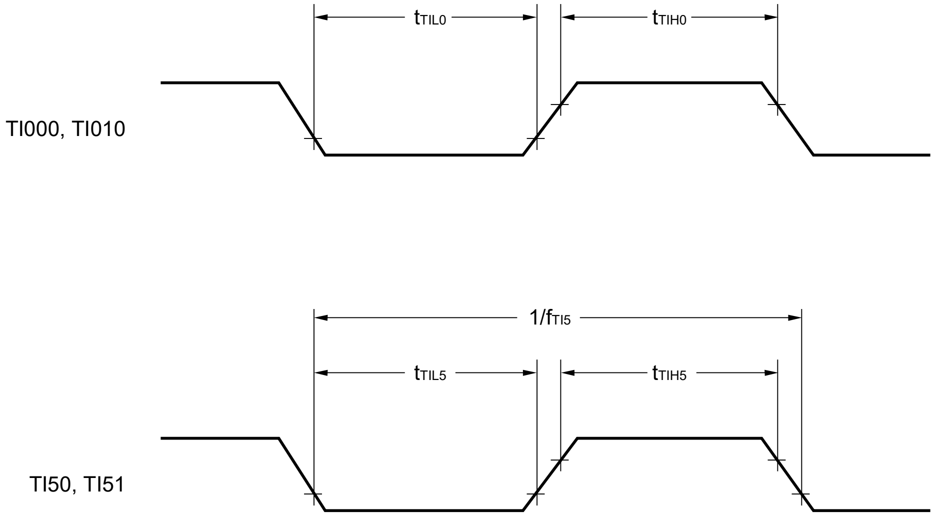
AC Timing Test Points (Excluding X1, XT1)



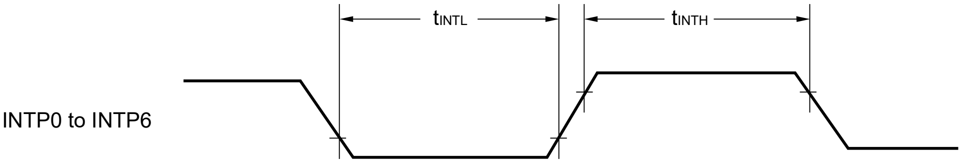
Clock Timing



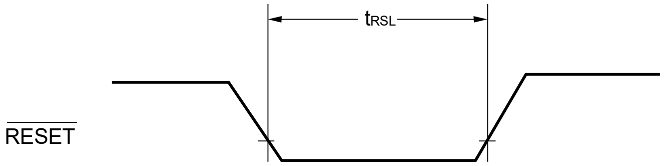
TI Timing



Interrupt Request Input Timing

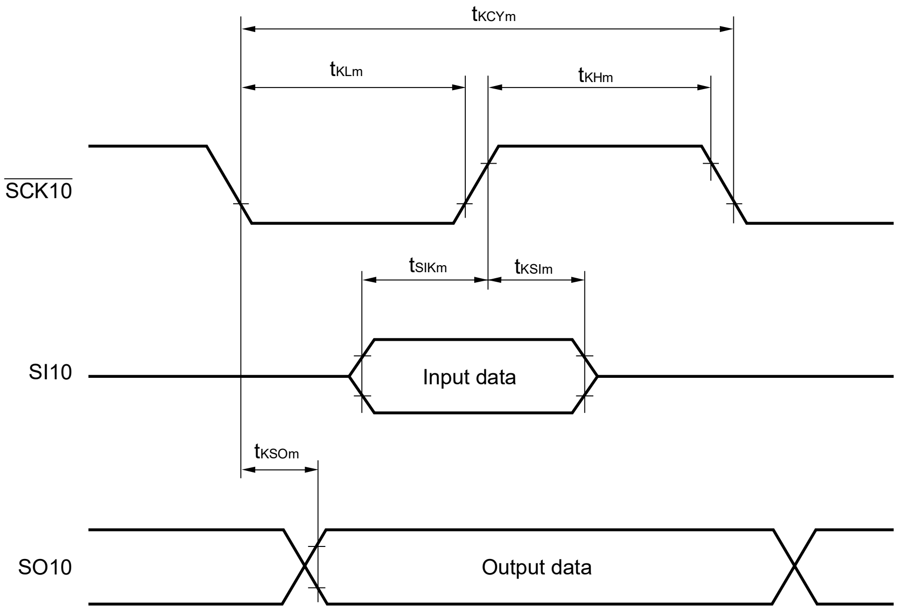


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

A/D Converter Characteristics
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V		±0.3	±0.6	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	14		100	μs
		2.7 V ≤ AV_{REF} < 4.0 V	17		100	μs
Zero-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Full-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
Integral non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
Differential non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
Analog input voltage	V_{AIN}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

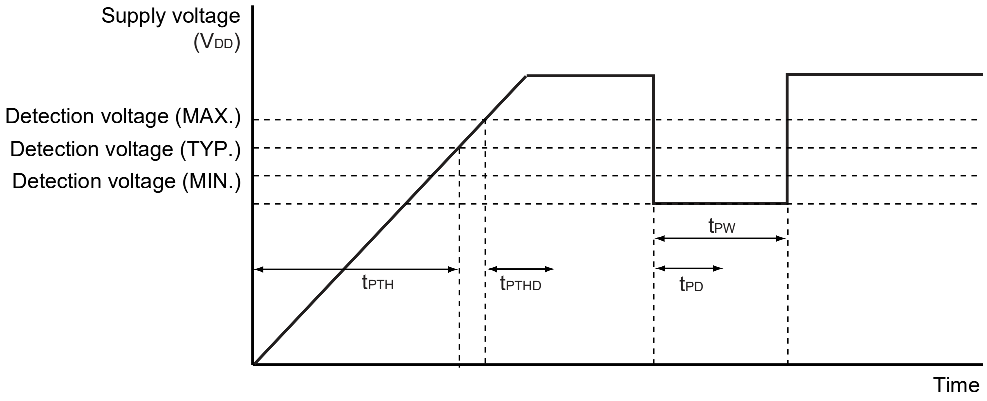
2. This value is indicated as a ratio (%FSR) to the full-scale value.

POC Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		2.0	2.1	2.2	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 2.1 V	0.0015			ms
Response delay time 1[Note]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note]	t_{PD}	When V_{DD} falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Note Time required from voltage detection to reset release.

POC Circuit Timing



LVI Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		4.1	4.3	4.5	V
	V_{LVI1}		3.9	4.1	4.3	V
	V_{LVI2}		3.7	3.9	4.1	V
	V_{LVI3}		3.5	3.7	3.9	V
	V_{LVI4}		3.3	3.5	3.7	V
	V_{LVI5}		3.15	3.3	3.45	V
	V_{LVI6}		2.95	3.1	3.25	V
	V_{LVI7}		2.7	2.85	3.0	V
	V_{LVI8}		2.5	2.6	2.7	V
	V_{LVI9}		2.25	2.35	2.45	V
Response time[Note 1]	t_{LD}			0.2	2.0	ms
Minimum pulse width	t_{LW}		0.2			ms
Operation stabilization wait time[Note 2]	t_{LWAIT}			0.1	0.2	ms

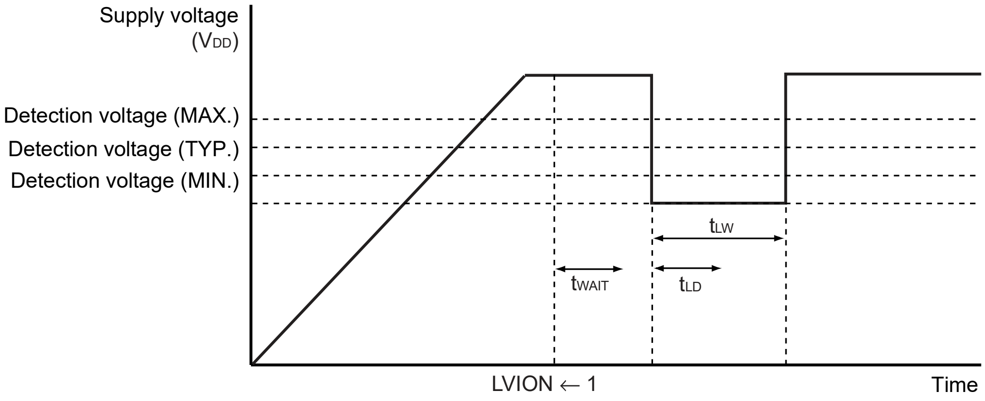
Notes 1. Time required from voltage detection to interrupt output or internal reset output.

2. Time required from setting LVION to 1 to operation stabilization.

Remarks 1. $V_{LVI0} > V_{LVI1} > V_{LVI2} > V_{LVI3} > V_{LVI4} > V_{LVI5} > V_{LVI6} > V_{LVI7} > V_{LVI8} > V_{LVI9}$

2. $V_{POC} < V_{LVIm}$ (m = 0 to 9)

LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		2.0		5.5	V
Release signal set time	t_{SREL}		0			μs

Flash Memory Programming Characteristics
(T_A = +10 to +65°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

(1) Basic characteristics

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply voltage		I_{DD}	f_{XP} = 10 MHz, V_{DD} = 5.5 V			T.B.D.	mA
Step erase time	Chip unit	T_{erac}		T.B.D.	T.B.D.	T.B.D.	ms
	Sector unit	T_{eras}		T.B.D.	T.B.D.	T.B.D.	ms
Erase time[Note 1]	Chip unit	T_{eraca}				25.5[Note 2]	ms
	Sector unit	T_{erasa}				25.5[Note 2]	ms
Step write time		T_{wrw}		T.B.D.	T.B.D.	T.B.D.	μs
Write time		T_{wrwa}				500[Note 2]	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 3]	100[Note 2]			Times

Notes 1. The prewrite time before erasure and the erase verify time (writeback time) are not included.

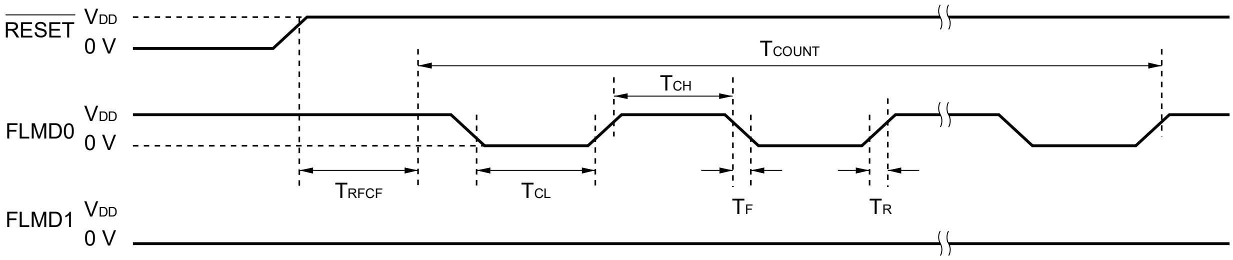
2. These values may change after evaluation.

3. When a product is first written after shipment, "erase → write" and "write only" are both taken as one rewrite.

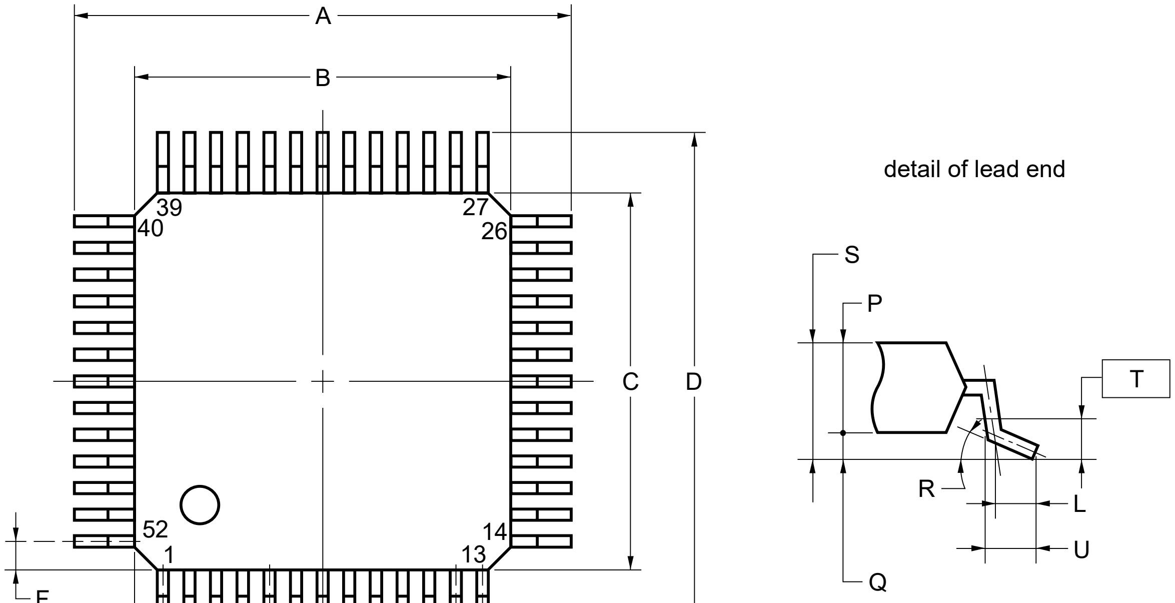
(2) Serial write operation characteristics

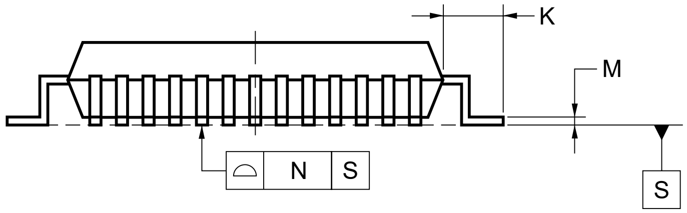
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time from $\overline{\text{RESET}}$↑ to FLMD0 count start	T_{RFCF}		$2^{19}/f_X + \alpha$			μs
Count execution time	T_{COUNT}				10	ms
FLMD0 counter high-/low-level width	T_{CH}/T_{CL}		T.B.D			μs
FLMD0 counter rise/fall time	T_R/T_F		T.B.D			μs

Serial Write Operation



52-PIN PLASTIC LQFP (10x10)





ITEM	MILLIMETERS
A	12.0±0.2
B	10.0±0.2
C	10.0±0.2
D	12.0±0.2
F	1.1
G	1.1
H	0.32±0.06
I	0.13
J	0.65 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.05}$
N	0.10
P	1.4
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.5±0.1
T	0.25
U	0.6±0.15

S52GB-65-8ET-2

CHAPTER 29 CAUTIONS FOR WAIT

29.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, refer to **Table 29-1**). This must be noted when real-time processing is performed.

29.2 Peripheral Hardware That Generates Wait

Table 29-1 lists the registers that issue a wait request when accessed by the CPU, and the number of CPU wait clocks.

Table 29-1. Registers That Generate Wait and Number of CPU Wait Clocks

Peripheral Hardware	Register	Access	Number of Wait Clocks
Watchdog timer	WDTM	Write	3 clocks (fixed)
Serial interface UART0	ASIS0	Read	1 clock (fixed)
Serial interface UART6	ASIS6	Read	1 clock (fixed)
A/D converter	ADM	Write	2 to 5 clocks[Note] (when ADM.5 flag = “1”) 2 to 9 clocks[Note] (when ADM.5 flag = “0”)
	ADS	Write	
	PFM	Write	
	PFT	Write	
	ADCR	Read	1 to 5 clocks (when ADM.5 flag = “1”) 1 to 9 clocks (when ADM.5 flag = “0”)
	<Calculating maximum number of wait clocks> $\{(1/f_{MACRO}) \times 2/(1/f_{CPU})\} + 1$ * The result after the decimal point is truncated if it is less than t_{CPUL} after it has been multiplied by $(1/f_{CPU})$, and is rounded up if it exceeds t_{CPUL}. f_{MACRO}: Macro operating frequency (When bit 5 (FR2) of ADM = “1”: $f_X/2$, when bit 5 (FR2) of ADM = “0”: $f_X/2^2$) f_{CPU}: CPU clock frequency t_{CPUL}: Low-level width of CPU clock		

Note No wait cycle is generated for the CPU if the number of wait clocks calculated by the above expression is 1.

Caution When the CPU is operating on the subsystem clock and the high-speed system clock is stopped (MCC = 1), do not access the registers listed above using an access method in which a wait request is issued.

Remark The clock is the CPU clock (f_{CPU}).

29.3 Example of Wait Occurrence

<1> Watchdog timer

<On execution of MOV WDTM, A>

Number of execution clocks: 8

(5 clocks when data is written to a register that does not issue a wait (MOV sfr, A).)

<On execution of MOV WDTM, #byte>

Number of execution clocks: 10

(7 clocks when data is written to a register that does not issue a wait (MOV sfr, #byte).)

<2> Serial interface UART6

<On execution of MOV A, ASIS6>

Number of execution clocks: 6

(5 clocks when data is read from a register that does not issue a wait (MOV A, sfr).)

<3> A/D converter

Table 29-2. Number of Wait Clocks and Number of Execution Clocks on Occurrence of Wait (A/D Converter)

<On execution of MOV ADM, A; MOV ADS, A; or MOV A, ADCR>

- When f_X = 10 MHz, t_{CPUL} = 50 ns

Value of Bit 5 (FR2) of ADM Register	f_{CPU}	Number of Wait Clocks	Number of Execution Clocks
0	f_X	9 clocks	14 clocks
	$f_X/2$	5 clocks	10 clocks
	$f_X/2^2$	3 clocks	8 clocks
	$f_X/2^3$	2 clocks	7 clocks
	$f_X/2^4$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)
1	f_X	5 clocks	10 clocks
	$f_X/2$	3 clocks	8 clocks
	$f_X/2^2$	2 clocks	7 clocks
	$f_X/2^3$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)
	$f_X/2^4$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)

Note On execution of MOV A, ADCR

Remark The clock is the CPU clock (f_{CPU}).

f_X: High-speed system clock oscillation frequency

t_{CPUL}: Low-level width of CPU clock

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0/KD1+.
Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows 95
  - Windows 98
  - Windows NT™ Ver. 4.0
  - Windows 2000
  - Windows XP

Caution For the development tools of the 78K0/KD1+, contact an NEC Electronics sales representative.